

SISTEMINĖS SUDĖTINGŲ ORGANIZACIJŲ MODELIAVIMO, PROJEKTAVIMO IR TOBULINIMO METODOLOGIJOS PAGRINDINIAI BRUOŽAI

(Metodinė priemonė)

1. ĮVADAS

Projektuodami valstybinę statistinę informacinę sistemą (SIS), mes naudosisimės **sisteminio sudėtingų organizacijų projektavimo metodu** ir tokia **sisteminio projektavimo automatizavimo priemone** kaip „Computer – Aided Systems Engineering“ (CASE), reiškiančia sistemų kūrimą kompiuterių pagalba.

Sisteminio projektavimo metodo esmę sudaro tai, kad sudėtingos organizacijos projektavimą pradedame nuo jos struktūros projektavimo, taikydami **funkcinio ir objektinio struktūros ir sudėties projektavimo metodų sintezę**, o po to, remdamiesi suprojektuota struktūra, projektuojame sistemos organizacijos funkcionavimą – **jos veiklos modelį, informacinius srautus** ir organizacijos **informacinį banką** bei reikalingą **programinę įrangą**.

Taikydami **funkcinį organizacijos struktūros projektavimo metodą**, pradedame nuo **organizacijos tikslų nustatymo**, jų formulavimo ir pagrindinių **tikslų dekompozicijos** į pagalbinius tikslus, šių – į dar smulkesnius pagalbinius tikslus, kol gauname **tikslų medžius**, kurių žemutinį lygį sudaro **elementarios funkcijos (operacijos)**.

Tokiu būdu gaunama organizacijos **funkcinė struktūra**, vaizduojanti **funkcijų hierarchiją ir paskirstymą organizacijoje**, – **funkcinis organizacijos struktūros modelis**.

Taikydami **objektinį organizacijos sudėties, jos struktūros ir jos veiklos projektavimo metodą**, pradžioje **išskiriame pagrindines sudėties sistemos dalis (posistemius ir jų elementus)**, įvesdami tokių objektų kaip **esybė** sąvoką; nustatome **esybių atributus ir ryšius**; valdymo sistemose, išskirdami **valdymo objektus ir subjektus bei ryšius tarp jų**, sudarome **valdymo kontūrus ir valdymo kontūrų piramides**; nustatome organizacijos **funkcionavimui ir jos valdymui reikalingą informaciją**. Tokiu būdu gaunamas **objektinis organizacijos sudėties ir struktūros modelis** – **sistemos elementų, jos veiklos subjektų ir objektų ryšių modelis**, vadinamas **sudėties schema (esybių ryšių modeliu)**.

Funkcinis ir objektinis organizacijos struktūros ir sudėties modeliai **derinami** vienas su kitu ir **jungiami (sintezuojami)** į vieną bendrą **sisteminį organizacijos struktūros ir sudėties modelį**. To jungimo metu **nustatomi ryšiai tarp** organizacijoje vykstančios veiklos subjektų ir objektų, pavadintų bendru **esybių** vardu, ir tos veiklos metu realizuojamų **funkcijų**; **pradedama projektuoti funkcijų vykdymo technologija**, kuri fiksuojama specialioje **technologinėje normatyvinėje organizacinėje dokumentacijoje (TNOD)**; funkcijų paskirstymas tarp veiklos subjektų fiksuojamas specialioje **struktūrinėje normatyvinėje organizacinėje dokumentacijoje (SNOD)**, kurią sudaro visos organizacijos **įstatai**, jos padalinių **nuostatai** bei pareigūnų **pareiginės instrukcijos (pareigybių pasai)**; sudarant SNOD ir TNOD, nustatomi funkcijų ir pareigūnų **informaciniai poreikiai ir informaciniai ryšiai**, o tai leidžia suprojektuoti **informacinius srautus ir lokalius informacinius bankus** bei **bendrą visos organizacijos informacinį banką**. Tokiu būdu gaunama **vykdomoji organizacijos struktūra**, vaizduojanti **pavaldumo ir informacinius ryšius** tarp organizacijos **subjektų, jų vykdomų funkcijų ir tų funkcijų objektų**, o taip pat pradedamas projektuoti **organizacijos funkcionavimo (veiklos) modelis** – jos **TNOD (OK, OI ir MTK) ir PNOD (darbo planai ir grafikai – tvarkaraščiai)**.

Vėlesniuose projektavimo etapuose sprendžiama, kurias iš organizacijoje realizuojamų funkcijų tikslinga palikti „**rankiniam**“ organizacijos darbuotojų vykdymui, kurias tikslinga

automatizuoti. Ryšium su tuo kuriamos atitinkamos **taikomosios sistemos**, jų **programinė įranga**. Tuo pačiu metu sprendžiami vykdomų funkcijų ir jų vykdytojų **informacinio aprūpinimo** klausimai, nustatoma, **kam kada kokia informacija ir kokiame režime (reglamentuotame, signaliniame ar užklausimu režime) bus teikiama.**

Kadangi **pagrindinę – aktyviąją – kuriamos bei tobulinamos organizacijos dalį sudaro žmonės, jos darbuotojai**, o normatyvinė organizacinė dokumentacija (SNOD, TNOD ir PNOD), kompiuteriai su savo programine įranga ir informacija su savo šaltais ir bankais yra tik priemonės, padedančios didinti žmonių veiklos efektyvumą, tai **vienas iš svarbiausių naujos sistemos kūrimo ir jos tobulinimo darbų yra organizacijos kolektyvo paruošimas darbui naujomis projekte numatytais sąlygomis.** Kolektyvas nepradės naujoviškai dirbti, tik pateikus jam kad ir geriausiai sudarytų projektų toms, kurių jis net perskaityti ir suprasti nepajėgs, ir todėl, **norėdami sukurti „žmogiškąją“ naujos – tobulesnės – sistemos dalį, mes privalome jau pačioje projektavimo pradžioje įtraukti organizacijos kolektyvą į naujos sistemos kūrimo darbus ir stengtis, kad visi organizacijos darbuotojai, kurie turės dirbti naujomis sąlygomis, maksimaliai tuose projektavimo darbuose dalyvautų.** Taip darydami, mes ne tik pagreitiname ir palengvinsime naujos sistemos įdiegimą, bet ir pagerinsime projekto kokybę, nes **niekas geriau nežino organizacijos darbo specifikos už tos organizacijos darbuotojus.** Reikia tik išmokyti juos projektuoti ir tobulinti savo veiklą, derinti interesus bei tikslus taip, kad kiekvienam darbuotojui taptų asmeniškai naudinga dirbti gerai ir asmeniškai nuostolinga dirbti blogai.

Gali kilti klausimas, **ar būtinai reikalingas, kuriant SIS, visos valstybinės statistikos sistemos struktūros ir veiklos projektavimas, ar nepakanka automatizuoti dabar atliekamus toje sistemoje darbus, įdiegiant atitinkamą techninę ir programinę įrangą, sukuriant informacinius bankus.**

Atsakymas į šį klausimą: **„Nepakanka. Nepakanka dėl dviejų priežasčių.“**

Pirmosios priežasties esmę sudaro tai, kad **sudėtingose informacinėse sistemose, kuriose informacija buvo apdorojama „rankiniu“ būdu, informacijos apdorojimo procesai buvo dekomponuoti į daug smulkių procesų**, nes žmogus per tam tikrą laiko vienetą pajėgus aprėpti ir apdoroti, palyginti, nedaug informacijos. Toks informacijos apdorojimo proceso skaidymas į smulkius paprocesius buvo neišvengiama būtinybė, išplaukianti iš „rankinio“ darbo specifikos ir gimdanti daug papildomų tarpinių darbų, surištų su informacijos perdavimais, perrašymais, skirtingais vienos ir tos pačios informacijos interpretavimais ir t.t., ko būtų galima išvengti (ar bent jau minimizuoti), jei informacijos apdorojimo procesus vykdytų vienas vykdytojas (ar bent jau mažesnis jų skaičius).

Pereinant prie informacijos apdorojimo kompiuterinės technologijos ir atsižvelgiant į tai, kad kompiuteris gali operuoti su žymiai didesnėmis informacijos apimtimis ir apdorojimo procesus atlikti žymiai greičiau, atsiranda galimybė smulkias informacijos apdorojimo operacijas jungti į stambius procesus ir tokiu būdu išvengti daugybės bereikalingų tarpinių darbų, tame skaičiuje ir daugybės tarpinių informacijos išvedimų iš kompiuterių bei įvedimų į juos, daugybės tarpinių dokumentų pildymo bei skaitymo.

Tai reiškia, jog, **automatizuojant informacijos apdorojimo procesus, netikslinga palikti nepakeistą „rankinę“ informacijos apdorojimo technologiją**, kompiuterizuojant ją „vienas prie vieno“. **Informacijos apdorojimo procesus reikia perprojektuoti, jungiant smulkius to apdorojimo procesus į stambesnius, o ryšium su tuo keisti funkcijų paskirstymą tarp darbuotojų ir kompiuterių, keisti organizacijos veiklą ir struktūrą.**

Antrosios priežasties galėtų nebūti, jei valstybinė statistikos sistema jau būtų pakankamai ilgą laiką dirbusi stabiliose sąlygose ir evoliucijos keliu būtų pasiekusi tobulą struktūrą ir veiklos technologiją, kurią dabar tereikėtų tik automatizuoti. Tačiau taip nėra. **Lietuvoje statistikos sistema dabar pergyvena pereinamąjį laikotarpį, ir joje funkcijų pasiskirstymas bei darbų vykdymo technologija dar nepasiekė nei reikiamo tobulumo, nei reikiamo stabilumo.** Ji dabar yra tam tikras konglomeratas, kuriame vienu metu koegzistuoja ir naujos sistemos elementai ir senos sistemos rudimentai. Automatizuoti dabar

vykstančius šioje sistemoje informacinius procesus „vienas prie vieno“ būtų ne tik netikslinga, bet ir žalinga, nes tai užkonservuotų neracionalią struktūrą ir stabdytų sistemos evoliuciją. Todėl, prieš pradedant informacinių procesų automatizavimą (ir kartu su juo), būtina peržiūrėti dabar veikiančios sistemos struktūrą ir joje vykdomu darbu technologijas, suprojektuoti tobulesnes, labiau atitinkančias šiuolaikinius reikalavimus bei ryšium su mokslo ir technikos progresu atsirandančias naujas galimybes.

Todėl ir siūlome SIS kūrimo procese siekti ne tik vykdomų darbų automatizavimo, bet ir valstybinės statistinės sistemos, jos struktūros ir veiklos **tobulinimo**. Įvykdyti šį didelį ir sudėtingą darbą, kaip jau buvo minėta, galima tik visos valstybinės statistinės sistemos kolektyvo, pirmoje eilėje jos ir jos padalinių vadovų, pastangomis.

Norint įtraukti organizacijos kolektyvą ir jo vadovus į organizacijos struktūros ir veiklos tobulinimo ir projektavimo darbus, reikia supažindinti kolektyvą ir jo vadovus su tų darbų esme ir jų vykdymo metodais. Tokio **supažindinimo tikslams ir skiriama ši apžvalginė metodinė priemonė.**

Joje bus išdėstyti pagrindiniai sisteminio organizacijų projektavimo metodologijos bruožai, kurių žinojimas ir panaudojimas galės būti naudingas visiems tiems, kas dalyvaus SIS projektavime ir projekto realizavime.

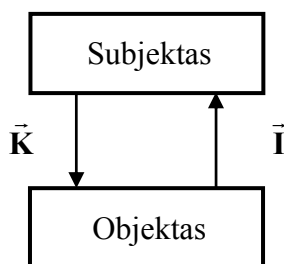
Kartu bus įvesta ir apibrėžta (ar bent jau suprantamai aprašyta) visa eilė sąvokų, kurių žinojimas yra būtinas, norint įsisavinti sisteminio projektavimo metodologiją ir tokio projektavimo automatizavimo priemonę – CASE.

**2. VEIKLOS SUBJEKTAI IR OBJEKTAI, ESYBĖS.
ESYBIŲ ATRIBUTAI IR RYŠIAI. VEIKLOS FUNKCIJOS.
ŠTRUKTŪRINĖ IR TECHNOLOGINĖ NOD.**

Veiklos subjektais vadinsime aktyviuosius veiklos elementus, veiklos iniciatorius ir vykdytojus.

Veiklos objektais vadinsime pasyviuosius veiklos elementus, tuos objektus, į kuriuos nukreipta veiklos subjektų veikla, kuriuose pasireiškia veiklos subjektų poveikių rezultatai ir kiti veiklos objektų sudėties (turinio), formos ar judėjimo trajektorijos bei greičio pasikeitimai.

Veiklos subjektų poveikiai į veiklos objektus gali būti **medžiaginio, energetinio ir informacinio** pobūdžio. Jie yra **išėjimai** veiklos subjektų atžvilgiu ir **įėjimai** veiklos objektų atžvilgiu. Jei veikla yra **valdymas**, tai veiklos subjektas vadinamas **valdymo subjektu**, veiklos objektas – **valdymo objektu** ir kartu sudaro **valdymo kontūrą** (žr. 1 pav.), kur $\vec{K}$ subjekto poveikių į objektą – **valdymo poveikių** – vektorius, $\vec{I}$ – **informacijos apie objektą, kurią gauna subjektas, vektorius**.



1 pav.

Valdymo kontūre realizuojamas tiesioginis ir grįžtamasis informaciniai ryšiai tarp subjekto ir objekto (plačiau apie informacinius poveikius ir valdymą žiūr. šios metodinės priemonės 54÷58 psl.).

Veiklos subjekto ir veiklos objekto sąvokos yra **reliatyvios**, nes tai, kas vienoje sistemoje bei situacijoje yra veiklos subjektas, kitoje situacijoje gali būti veiklos objektu ir atvirkščiai.

Veiklos objekto ir veiklos subjekto sąvoką galima apibendrinti, įvedus esybės sąvoką.

Esybe yra vadinama bet kas, kas egzistuoja ar gali egzistuoti realioje ar virtualioje erdvėje ir laike, kas pasireiškia ar gali pasireikšti kokiomis nors **savybėmis (atributais)**, kas yra ar gali būti kokiuose nors **santykiuose (ryšiuose)** su veiklos subjektais bei objektais. Patys veiklos subjektai ir objektai – taip pat yra esybės. Galima sakyti, jog **esybė – tai bet koks kokio nors atributų rinkinio nešėjas**. Ryšiai tarp esybių vaizduojami **esybių ryšių diagramomis (ERD)**. Valdymo kontūras, pavaizduotas 1 pav., yra ERD fragmento pavyzdys. **Valdymo kontūru piramidė**, vaizduojanti **pavaldumo ir atsiskaitomybės ryšius** kokios nors organizacijos **valdymo sistemoje**, yra ERD pavyzdys. ERD gali vaizduoti ne tik medžiaginius bei energetinius, bet ir informacinius ryšius, pavyzdžiui, tarp Statistikos departamento tarnybų, skyrių ir darbuotojų, o taip pat, tarp statistinės informacijos vartotojų.

Veikla, kurią organizacijoje, būdami jos nariais, privalo vykdyti veiklos subjektai, vadinama tų subjektų **veiklos funkcijomis** arba trumpiau – tiesiog **funkcijomis**. Veiklos funkcijos gali turėti įvairų **sudėtingumą** ir **darbo imlumą**, o jų subjektais gali būti ne tik pavieniai organizacijos darbuotojai, bet ir įvairūs organizacijos padaliniai bei jų grupės, o kartais net ir visa organizacija.

Kiekvieną sudėtingą veiklos funkciją galima dekomponuoti į elementarias funkcijas (operacijas).

Elementaria veiklos funkcija (operacija) vadinama tokia veiklos funkcija, kurią vykdo vienas elementarus veiklos subjektas (vienas asmuo ar viena pastovi kartu veikianti kelių asmenų grupė), veikdamas vieną elementarų veiklos objektą vienoje darbo vietoje su viena ir ta pačia įranga, vieną (paprastai nepertraukiamą) laiko intervalą, kurio metu vykdoma kelių, būdingų tai operacijai, subjekto veiksmų seka (kartais – vienas veiksmas). Operacijų pavyzdžiais gali būti gręžimo bei tekinimo operacijos gamybinėje veikloje, algos išmokėjimo darbuotojui operacija kokios nors įstaigos kasoje, atvykusio į polikliniką ligonio registracijos operacija ir pan.

Operacijos **vykdymo eiga**, jos **algoritmas**, aprašomi specialiame technologiniame normatyviniame dokumente, vadinamame **operacine korta (OK)**, jei tą aprašymą galima išsprauti į standartizuotą formą, turinčią specializuotos paskirties eilutes ir stulpelius, arba turinčiame laisvą formą dokumente – **operacijos vykdymo instrukcijoje (OI)**, jei operacijos vykdymo algoritmo aprašymo į standartizuotą formą išsprauti nepavyksta ir tenka naudotis laisva verbaline forma. OK ir OI atsako į klausimą, **kaip** vykdyti atitinkamą operaciją ir **per kiek laiko** tą operaciją reikėtų atlikti. OK pavyzdžių galima rasti gamybinių procesų aprašyme. OI pavyzdžiais gali būti įvairios instrukcijos darbui su kompiuteriais. Automatizuojant kokią nors „rankinę“ operaciją, jos OK (ar OI) tampa tos operacijos vykdymo algoritmas (programos modulis, realizuojantis tą algoritmą).

Aprašant sudėtingas funkcijas, susidedančias iš kelių operacijų, kurias vykdo skirtingi veiklos subjektai skirtingose darbo vietose ir skirtingu laiku, tokių dokumentų kaip OK ar OI jau nebepakanka ir tenka papildomai sudaryti taip vadinamas **maršrutines technologines kortas (MTK)**. MTK nurodo, **iš kokių elementarių operacijų sudėtinga veiklos funkcija susideda, kokia eilės tvarka tos operacijos vykdomos, kas ir kur tas operacijas vykdo, kokiomis OK ar OI vadovaujasi**. MTK kartu su savo OK ir OI pilnai aprašo sudėtingos funkcijos vykdymo procesą erdvėje, nustato tos funkcijos **vykdymo maršrutą**.

Kad funkcijų vykdymo procesai būtų **suderinti ne tik erdvėje, bet ir laike**, sudaroma dar viena normatyvinių technologinių organizacinių dokumentų grupė – **planinė normatyvinė organizacinė dokumentacija (PNOD)**, t.y. visos organizacijos, jos padalinių ir darbuotojų **darbo planai ir grafikai – tvarkaraščiai (DP ir GT)**. Juose nustatoma, **kas kada kokias funkcijas (operacijas) privalės vykdyti**. Sudarant DP ir GP, kurių sudarymas **pradedamas „nuo viršaus“** – nuo visos organizacijos ir jos vadovų DP ir GT sudarymo,

tenka įvykdyti sudėtingą įvairių operacijų laike ir erdvėje vykdymo derinimą. Tokiam derinimui gali daug padėti speciali priemonė – **tinkliniai grafikai (TG)** – ir specialus metodas – **tinklinis planavimas**. TG gali būti panaudoti ne tik kaip planavimo, bet ir kaip **operatyvaus valdymo ir kontrolės instrumentai**.

OK, OI, MTK, DP ir GT, sudaro taip vadinamą **technologinę ir planinę normatyvinę organizacinę dokumentaciją (TNOD ir PNOD)** – **normatyvinį veiklos funkcijų vykdymo technologijos modelį**, vaizduojantį, kaip, kokia eilės tvarka, kada ir naudojantis kokiomis priemonėmis (tame skaičiuje, ir naudojantis kokia, iš kur gaunama ir kam perduodama informacija) turėtų būti vykdomos veiklos funkcijos organizacijoje. **Turint gerai suprojektuotą TNOD ir PNOD jau gana nesunku automatizuoti organizacijoje vykstančią veiklą (tiksliau tos veiklos rutininę dalį)**, perduodant eilės funkcijų, turinčių didelį darbo imlumą ir aiškiai apibrėžtus algoritmus, atlikimą kompiuteriams **reglamentuotame programiniame ar žmogaus dialogo su kompiuteriu režime**.

Tačiau sudaryti TNOD ir PNOD galima tik tada, kada jau žinoma, **kokias veiklos funkcijas reikia atlikti organizacijoje, kas ir kur tas funkcijas privalo vykdyti, kas kam turi būti pavaldus, kas už ką ir kam turi būti atskaitingas bei atsakingas**. Atsakymus į šiuos klausimus gauname, projektuodami taip vadinamąją **struktūrinę normatyvinę organizacinę dokumentaciją (SNOD)**, t.y. **projektuodami organizacijos struktūrą**, apie ką jau buvo kalbėta šios metodinės priemonės įvade.

Struktūrinę NOD sudaro: 1) visos **organizacijos įstatai**, nusakantys organizacijos **misiją** (paskirtį), pagrindines jos **veiklos kryptis ir tikslus**, pagrindines **organizacijos funkcijas, organizacijos sudėtį ir struktūrą**; 2) organizacijos **padalinių nuostatai**, vaidinantys padalinio atžvilgiu tą patį vaidmenį kaip ir organizacijos įstatai visos organizacijos atžvilgiu ir šalia to apibrėžiantys padalinio vietą ir vaidmenį organizacijoje; 3) pareigūnų **pareiginės instrukcijos (pareigybių pasai)**, apibrėžiantys pareigybės vietą ir vaidmenį organizacijoje, pareigybės tikslus ir veiklos kryptis, pareigybės funkcijas, teises ir atsakomybę, pareigūno gaunamą, apdorojamą ir perduodamą informaciją.

Apie tai, kaip projektuojama organizacijos struktūra ir tą struktūrą aprašanti SNOD, o po to – ir TNOD bei PNOD, bus rašoma tolimesniuose šios metodinės priemonės skyriuose, kai jau bus įvestos tokio projektavimo aprašymui reikalingos specialios sąvokos.

3. SISTEMOS IR JŲ TIPAI.

SISTEMOS SUDĖTIS IR STRUKTŪRA. SISTEMOS ELEMENTAI IR POSISTEMIAI. SPECIFINĖS TECHNINIŲ SISTEMŲ IR ORGANIZACIJŲ SAVYBĖS. SISTEMOTYRA IR JOS SUDĖTINĖS DALYS: TEORINĖ, METODOLOGINĖ IR TAIKOMOJI.

Žodis „sistema“ vartojamas gana plačiai, tačiau ne visi ir ne visada pilnai suvokia tiksliai to žodžio reikšmę ir ne visuomet tinkamai jį vartoja.

Galima išskirti **du sistemų tipus** ir pakankamai pilnai bei aiškiai apibūdinti kiekvieną iš jų.

Pirmąjį tipą sudaro sistemos – **konstravimo** (sąmoningo ir nesąmoningo) rezultatai. Tai sistemos, sukonstruotos žmogaus bei atsiradusios gamtoje (ir bendrai pasaulyje) evoliucijos (ir kitų procesų) keliu.

Šio tipo sistema – tai kelių (keliolikos ir daugiau) sąveikaujančių **esybių visuma**, kurioje to sąveikavimo, turinčio tai visumai būdingus **esybių ryšius–sistemos struktūrą**, rezultate **atsiranda naujos, priklausančios visai visumai savybės** (taip vadinami **sisteminiai atributai**), kurių neturi tą visumą sudarančios esybės, paimitos atskirai.

Pavyzdžiui, lėktuvas yra pirmojo tipo sistema, nes sujungę to lėktuvo sudėtines dalis į tam tikrą struktūrą, gauname visumą, kuri gali skristi, ir ta savybė skristi priklauso visai visumai, nes atskirai paimitos lėktuvo detalės skristi negali.

Šeima taip pat yra pirmojo tipo sistema, nes turi tokių savybių, kurių neturi nė vienas atskirai paimtas šeimos narys.

Pirmojo tipo sistema yra ir vandens molekulė, turinti cheminę formulę H_2O , nes, susijungus vandeniliui ir deguoniui, atsiranda tokios vandens savybės, kurių neturi nei deguonis, nei vandenilis, paimti atskirai.

Pirmojo tipo sistemų esmę sudaro tai, kad iš santykių tarp esybių, susijungusių į visumą, išplaukia naujos savybės, priklausančios visai visumai ir išskiriančios ją kaip **naują esybę**, turinčią savus atributus. **Naujų savybių gavimas ir yra pagrindinis pirmojo tipo sistemų kūrimo tikslas.**

Antrąjį sistemų tipą sudaro sistemos – **sistematizavimo (klasifikavimo) rezultatai**. Šias sistemas sudarančios esybės būdingos tuo, kad jų atributai yra tokie, jog pagal tuos atributus esybes galima išdėstyti (klasifikuoti) tokia tvarka, kurioje **kiekviena esybė gauna tik jai vienai būdingą vietą**, o esybių atributai tenkina visai **esybių visumai būdingus dėsningumus** – taip vadinamus **sisteminis santykius**.

Antrojo tipo sistemų pavyzdžiai: periodinė cheminių elementų sistema, augalų bei gyvūnų klasifikavimo sistemos, duomenų bankai kompiuteriuose, statistinių rodiklių klasifikatoriai ir pan.

Jei pirmojo tipo sistemose iš santykių tarp jas sudarančių esybių išplaukia naujos tas sistemas charakterizuojančios ir išskiriančios jas iš kitų esybių sisteminės savybės, tai antrojo tipo sistemose jas sudarančių esybių savybės tenkina būdingus toms sistemoms ir išskiriančius iš kitų sistemų sisteminis santykius. Tokių **sisteminis santykių suradimas ir jų panaudojimas, nustatant kiekvienos esybės, įeinančios į sistemos sudėtį, vietą sistemoje kaip tik ir yra antrojo tipo sistemų sudarymo pagrindinis tikslas.**

Esybės, kurios sudaro pirmojo bei antrojo tipo sistemas, nusako tų sistemų **sudėtį**, ryšiai (santykiai) tarp tų esybių bei jų savybių ir tų esybių vietos bei vaidmenys sistemoje – vidinę sistemų **struktūrą**.

Nagrinėdami sistemų sudėtį ir struktūrą, dekomponuodami sistemas į jų sudėtinės dalis, visuomet galima pasiekti tokį dekompozicijos lygį, kada tolimesnė dekompozicija į dar smulkesnes ir paprastesnes esybes tampa arba **nebegalima arba nebetikslinga**. Tokias maksimaliai paprastas ir toliau nebedalomas esybes, įeinančias į sistemos sudėtį, vadina **sistemos elementais**, o ryšius tarp tokių esybių – sistemos vidinę **struktūrą elementu lygyje**.

Dekomponuojant sistemą į sudėtinės dalis, galima išskirti ir tokias jos dalis, kurios pačios susideda iš sistemos elementų ir gali būti dekomponuotos į smulkesnes dalis. Tokios sudėtinės dalys vadinamos sistemos **posistemiais**, o ryšiai tarp jų – sistemos vidinę **struktūrą posistemų lygyje**.

Sistemos, posistemo ir elemento sąvokos yra reliatyvios, nes viena ir ta pati esybė neretai gali būti interpretuojama ir kaip sistema, ir kaip didesnės sistemos posistemis, ir net kaip kokios nors sistemos elementas. Pavyzdžiui, mūsų planeta Žemė yra labai sudėtinga sistema ir tuo pačiu metu – didesnės sistemos – Saulės sistemos – elementas.

Kalbant apie sistemų ir jų elementų savybes (atributus), reikia skirti **tris atributų tipus**: 1) **individualios** sistemos elementų savybės, kurios priklauso elementams nepriklausomai nuo to, priklauso tie elementai kokiai nors sistemai ar nepriklauso; 2) **struktūrinės** elementų savybės, kurios atsiranda elementuose, patekus jiems į sistemos sudėtį ir užėmus jos struktūroje atitinkamą vietą; 3) **sisteminės** visos sistemos savybės, kurios atsiranda tik susijungus esybėms į sistemą, ir priklauso visai sistemai.

Reikia skirti ir tokias sąvokas, kaip **sistemos sudėties** ir **sistemos struktūros** elementas.

Sistemos sudėties elementas – tai į sistemos sudėtį įeinanti esybė su savo individualiais atributais.

Sistemos struktūros elementas – tai ta vieta sistemos struktūroje, kurią turi užimti į sistemos sudėtį įeinanti elementari esybė, tos esybės vaidmuo (rolė) sistemoje.

Pavyzdžiui, kokia nors pareigybė organizacijoje yra organizacijos struktūros

elementas, o konkretus asmuo, užimantis šiuo metu minėtas pareigas – organizacijos sudėties elementas.

Galima sakyti, jog sistemos elemento individualios savybės priklauso sistemos sudėties elementui, o struktūrinės – sistemos struktūros elementui, nes jos išplaukia iš elemento vietos ir vaidmens sistemos struktūroje.

Priklausomai nuo individualių savybių sistemų elementai gali būti **pasyvūs** ir **aktyvūs**.

Pasyvūs sistemų elementai neturi savo tikslų ir jų funkcionavimas, jei tie elementai yra veikūs (nesugedę), pilnai apibrėžiamas visos sistemos funkcionavimu ir elemento užimama vieta sistemos struktūroje.

Aktyvūs elementai gali turėti nuosavus tikslus, kurie ne būtinai sutampa su visos sistemos tikslais, o taip pat su kitų sistemos elementų bei posistemų tikslais. Todėl aktyviųjų elementų veikla ne visuomet yra pilnai apibrėžta ir ne visuomet išplaukia iš elementų užimamos vietos sistemos struktūroje. Aktyvus elementas gali inicijuoti ir sistemoje nenumatytą veiklą.

Pasyviųjų elementų pavyzdžiu gali būti kokios nors mašinos detalės, o aktyviųjų – žmonės, įeinantys į kokios nors organizacijos sudėtį.

Jei kokios nors sistemos pagrindiniai elementai yra pasyvūs, o aktyviųjų elementų sistemoje arba visai nėra arba aktyvūs elementai tą sistemą tik aptarnauja ar naudojami ja savo tikslams, tai tokia sistema vadinasi **technine sistema**.

Jei kokios nors sistemos pagrindiniai elementai yra aktyvūs, tai tokia sistema vadinasi **organizacija**. Į organizacijos sudėtį kartu su aktyviais elementais gali įeiti ir pasyvūs elementai.

Techninės sistemos pavyzdys – kompiuteriai bei kompiuterių tinklas. Organizacijų pavyzdžiai: Statistikos departamentas, įmonė, visuomeninė ar politinė organizacija.

Svarbu įsisąmoninti, jog SIS, kurią turi sukurti Statistikos departamento darbuotojai, **nėra tik techninė sistema**, o yra **organizacinė sistema**, ir todėl jos sukūrimas nėra tik techninių elementų sukūrimas (ar nupirkimas) ir jų sujungimas į techninę sistemą, o yra ir aktyviųjų elementų paruošimas bei jų subūrimas (suorganizavimas) į atitinkamą struktūrą atitinkamos veiklos vykdymui. Tinkamai veikiančios organizacijos sukūrimas yra žymiai sudėtingesnis ir sunkesnis darbas, negu tą organizaciją aptarnaujančių techninių sistemų sukonstravimas.

Techninės sistemos ir organizacijos, būdamos pirmojo tipo sistemomis, turi eilę bendrų savybių, ir todėl, konstruojant ir kuriant organizacijas, **galima ir tikslinga naudotis gana turtinga sudėtingų techninių sistemų projektavimo patirtimi, inžinieriniais projektavimo metodais**.

Tačiau vien tik tokių metodų, kuriant organizacijas, **nepakanka, nes organizacijos turi eilę specifinių savybių**, kuriomis skiriasi nuo techninių sistemų.

Kadangi techninių sistemų elementai yra pasyvūs, tai jų veiklos „organizavimui“ pakanka suprojektuoti techninės sistemos struktūrą ir funkcionavimo technologiją, o po to realizuoti tuos projektus, mechanškai sujungiant elementus į sistemą taip, kaip reikalauja suprojektuota struktūra. Tam pilnai pakanka taip vadinamų **mechanistinės ir terminalinės metodologijų**, besiremiančių gamtos dėsnių ir sistemos elementų savybių žinojimu. Kadangi ir pasyviųjų elementų funkcionavime gali įvykti įvairių atsitiktinumų (pvz. atsitiktinių gedimų), tai naudojantis **tikimybių teorija ir statistiniais metodais**, nustatomos tų atsitiktinumų tikimybės, apskaičiuojami ir sudaromi atitinkami rezervai, užtikrinantys normalų techninės sistemos funkcionavimą. Techninių sistemų ir jų elementų valdymas – tai techninėje sistemoje vykstančių **veiksmų valdymas**.

Bandyti **tiesiogiai valdyti aktyviųjų elementų veiksmus organizacijoje yra neefektyvu ir todėl netikslinga**, nes tuomet reikėtų nuolat kontroliuoti kiekvieno aktyvaus elemento veiklą, o tai ir brangu, ir neefektyvu, kadangi iškyla pačių kontrolierių veiklos kontroliavimo problema, antros eilės kontrolierių kontroliavimo problema ir t.t. be galo.

Kadangi aktyvūs elementai turi nuosavas individualias valdymo sistemas (savo

galvas), nuosavus interesus ir siekia nuosavų tikslų, kuriuos galima išsiaiškinti ir žinoti, tai žymiai efektyviau aktyviųjų elementų veiksmus valdyti ne tiesiogiai, o per tų elementų interesus ir tikslus. Jei norime, kad aktyvusis elementas gerai vykdytų savo funkcijas organizacijoje, reikia sudaryti tokias sąlygas (suprojektuoti tokią organizaciją), kad savo funkcijų kokybiškas vykdymas aktyviajam elementui būtų asmeniškai naudingas, o jų nevykdymas ar nekokybiškas vykdymas – asmeniškai nuostolingas. Kitaip tariant, reikia suderinti organizacijos ir jos aktyviųjų elementų tikslus, aktyviųjų elementų veiksmus reikia valdyti per jų tikslus.

Todėl, projektuojant organizacijas ir jų veiklą, **terminalinės ir mechanistinės metodologijų** jau nebepakanka. Jas reikia papildyti **teleologine ir organizmine metodologijomis**, kur žodis „teleo“ reiškia **tikslą**, o organizminė metodologija supranta organizacijos kūrimą ne tik kaip mechaninių sistemos dalių sujungimą į sistemą, bet ir kaip organizacijos, atstovaujančios tam tikrą **organizmų tipą, ugdymą**, būdingą visiems organizmams.

Teleologinė metodologija pradeda nuo organizacijos ir jos aktyviųjų elementų tikslų išsiaiškinimo ir tų tikslų derinimo (koordinavimo), o organizminė metodologija žiūri į organizacijos kūrimo procesą kaip į organizacijos organizmo ir jo sudėtinių dalių nuolatinio ugdymo, vystymo ir tobulinimo procesą.

Sisteminis sudėtingų organizacijų, į kurių sudėtį įeina ir aktyvūs ir pasyvūs elementai, kūrimo **metodas realizuoja įvairių metodologijų – teleologinės ir terminalinės, organizminės ir mechanistinės – sintezę**, o taip pat, kaip jau buvo minėta šios metodinės priemonės įvade – **funkcinės ir objektinės metodologijų sintezę**.

Įvairių metodų ir mokslų sintezė, reikalaujanti įvairių sričių specialistų bendradarbiavimo, tiriant ir kuriant sudėtingas sistemas, – būdingas naujo mokslo – **sistemotyros** – bruožas.

Pirmąsias **bendrosios sistemų teorijos** idėjas dar trečiajame XX amžiaus dešimtmetyje yra išsakęs austrų biologas L. fon Bertalanfi (L. von Bertalanffy), tačiau pradžioje tos idėjos platesnio atgarsio nesulaukė. Ir tik po Antrojo pasaulinio karo, prasidėjus „kibernetiniam bumui“, tos idėjos buvo įvertintos ir pradėtos audringai vystyti.

Istoriškai susiklostė trys persidengiančios ir viena kita papildančios mokslo apie sistemas vystymosi kryptys: **teorinė, metodologinė ir taikomoji**.

Teorinė sistemotyros vystymosi kryptis, nagrinėjanti sistemų sandaros, jų funkcionavimo bei kūrimo (kūrimosi) dėsningumus, įvedanti pagrindines sistemų mokslo sąvokas ir kurianti sistemotyros mokslinę kalbą bei matematinį aparatą, buvo pavadinta **sistemų teorija (bendrają, matematine ir pan.)**. Tos krypties atstovai: jau minėtas L. fon Bertalanfi, K.Bouldingas (K.Boulding), M.Mesarovičius (M.D.Mesarovic), A.Rapoportas (A. Rapoport), U.Ešbi (W. Ross Ashby) ir kiti.

Metodologinę sistemotyros vystymosi kryptį, nagrinėjančią **sisteminės analizės metodus**, atstovauja S.Optneris (žr. S.L.Optner, „Systems Analysis for Business and Industrial Problem Solving“), o taip pat L. fon Bertalanfi, M. Mesarovičius, A.Rapoportas, U.Ešbi, R.Akofas (R.L.Ackoff), S.Jangas (S.Joung) ir kiti.

Taikomojoje sistemotyros dalyje galima išskirti dvi pagrindines kryptis:

1) **sistemotechniką**, užsiimančią sudėtingų techninių sistemų projektavimo ir realizavimo metodų bei sisteminės technologijos taikymais (žr., pav., knygą rusų kalba: Gud G.Ch. Ir Makol R.E. „Sistemotechnika“, esančią vertimu iš anglų kalbos) ir 2) **orgprojektavimą, t.y. sudėtingų organizacijų konstravimo metodų taikymą**, projektuojant bei tobulinant sudėtingas organizacijas (žr., pav., knygą S.Juong, „Management a Systems Analyses“, turinčią vertimą į rusų kalbą: Jang S. „Sistemnoe upravlenie organizacije“ su gana informatyviu ir vertingu S.P. Nikanorovo įvadinio straipsniu).

Gana artima taikomajai sistemotyros daliai yra ir **operacijų tyrimo** kryptis, kuriai atstovauja jau minėtas R.L.Akofas ir eilė kitų mokslininkų ir kuri dabar išsivystė į atskirą **sudėtingų funkcijų vykdymo technologijos optimizavimo mokslą ir praktiką**.

Šioje metodinėje priemonėje mes nenagrinėsime visų sistemotyros sudėtinių dalių, nes tai pareikalautų daug laiko ir vietos. Tolimesniuose metodinės priemonės skyriuose mes apsistosime tik prie **sistemų modeliavimo, organizacijų projektavimo ir jų veiklos optimizavimo klausimų**, kurie yra labai aktualūs SIS projektuotojams. Žymiai daugiau žinių apie sistemotyra skaitytojas gali rasti šių eilučių autoriaus parengtoje mokymo priemonėje „Matematinės sistemotyros pradmenys“.

4. SISTEMŲ MODELIAI

Žmogus, galvodamas apie kokią nors esybę, operuoja tos esybės sąvoka, jos **konceptiniu modeliu**, išreiškiančiu tos esybės įvaizdį ir sampratą žmogaus sąmonėje.

Sudėtingų sistemų atveju žmogus jau nebepajėgus iškart aprėpti savo intelektu visą sudėtingą sistemą, susidaryti pakankamai pilną jos vaizdą ir tos sistemos funkcionavimo „mechanizmų“ supratimą.

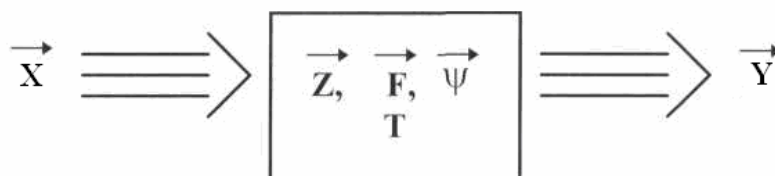
Tuomet žmogus pasitelkia sau į pagalbą įvairius **vaizdinius bei simbolinius modelius**, kuriuos jis susikuria, nagrinėdamas sudėtingą sistemą, ir užfiksuoja (popieriuje, kompiuterio atmintyje ir pan.), o po to tą **neaprepiamos sistemos nagrinėjimą pakeičia lengviau aprėpiamo jos modelio nagrinėjimu**.

Sistemų modeliai gali būti sisteminiai ir optimizaciniai.

Sisteminiai modeliai vaizduoja nagrinėjamos sistemos sąveikavimą su jos aplinka ir vidinį sistemos funkcionavimą, o **optimizaciniai modeliai** padeda sudaryti optimalią sistemos struktūrą, optimalią sistemos funkcionavimo („judėjimo“ daugiamatėje funkcinėje erdvėje ir laike) trajektoriją, nustatyti optimalias jei ne visų tai bent kai kurių nagrinėjamos sistemos parametrų bei valdomųjų kintamųjų reikšmes.

Tie sisteminiai modeliai, kurie vaizduoja tik sistemos sąveikavimą su išorine jos aplinka, vadinami **išoriniais**, o tie sisteminiai modeliai, kurie vaizduoja sistemos sudėtį, jos vidinę struktūrą, o taip pat vidinį sistemos funkcionavimą (procesus, vykstančius sistemos viduje) – **vidiniais**.

Išorinį matematinį modelį (žr. 2 pav.), sudaro:



2 pav.

1) sistemos **įėjimų vektoriaus** $\vec{X} = (x_1, x_2, \dots, x_i, \dots, x_n)$ **aprašymas**, kur vektoriaus $\vec{X}$ koordinatės x_i yra aplinkos poveikių į nagrinėjamą sistemą įverčiai, galintys turėti medžiaginę, energetinę ar informacinę prigimtį ir galintys būti kiekybiniai ar kokybiniai;

2) sistemos **išėjimo vektoriaus** $\vec{Y} = (y_1, y_2, \dots, y_i, \dots, y_m)$ **aprašymas**, kur vektoriaus $\vec{Y}$ koordinatės y_i yra sistemos poveikių aplinkai įverčiai; jie taip pat gali turėti medžiaginę, energetinę ar informacinę prigimtį ir gali būti kiekybiniai ar kokybiniai;

3) sistemos **vidinės būsenos vektoriaus** $\vec{Z} = (z_1, z_2, \dots, z_p, \dots, z_r)$ **aprašymas**, kur vektoriaus $\vec{Z}$ koordinatės z_p yra sistemos vidinių parametrų įverčiai; šie įverčiai taip pat gali turėti medžiaginę, energetinę ar informacinę prigimtį ir gali būti kiekybiniai ar kokybiniai;

4) kadangi sistemos įėjimų, išėjimų ir būsenų koordinačių reikšmės gali būti stebimos, matuojamos bei fiksuojamos dokumentuose ne būtinai kiekvienu laiko momentu, o tik tam tikrais laiko momentais ar laiko tarpais, tai į sistemos išorinį modelį įeina ir **sistemos**

kalendoriaus T aprašymas; kalendorius T gali būti **tolydus** (tolydi laiko atkarpa), **diskretus** (laiko momentų seka) ir **mišrus** (susidedantis iš laiko atkarpų ir atskirų laiko momentų);

5) sistemos išėjimų priklausomybės nuo sistemos įėjimų ir būsenos, taip vadinamos **išėjimų funkcijos** $\vec{Y} = \vec{F}(\vec{X}, \vec{Z})$, susidedančios iš m skaliarinių išėjimo funkcijų $y_j = f_j(\vec{X}, \vec{Z})$, **aprašymas**;

6) naujos būsenos $\vec{Z}(t+\Delta t)$ ar jos kitimo greičio $\frac{d\vec{Z}(t)}{dt}$ priklausomybės nuo sistemos įėjimo $\vec{X}(t)$ ir senos būsenos $\vec{Z}(t)$, taip vadinamos **perėjimų funkcijos** $\vec{Z}(t+\Delta t) = \vec{\psi}(\vec{X}(t), \vec{Z}(t))$ ar $\frac{d\vec{Z}(t)}{dt} = \vec{f}(\vec{Z}(t), \vec{X}(t))$, susidedančios iš r skaliarinių funkcijų $z_p(t+\Delta t) = \psi_p(\vec{X}(t), \vec{Z}(t))$ ar $\frac{dz_p(t)}{dt} = f_p(\vec{X}(t), \vec{Z}(t))$, **aprašymas**. Abi šias perėjimų funkcijos aprašymo formas, naudodamiesi Lebegeo (ar dar bendresniu) integralu, galime suvesti į vieną: $\vec{Z}(t+\Delta t) = \vec{Z}(t) + \int_t^{t+\Delta t} \vec{f}(\vec{Z}(t), \vec{X}(t)) dt = \vec{\psi}(\vec{Z}(t), \vec{X}(t))$, tinkančią ir tolydžiam, ir diskrečiam, ir mišriems kalendoriui.

Turėdami išorinį matematinį sistemos modelį ir **žinodami sistemos pradinę būseną** $\vec{Z}(t_0)$ (kur $t=t_0$ yra koks nors pradinis laiko momentas, kada žinome sistemos būseną) bei sistemos įėjimų priklausomybę nuo laiko (**sistemos įėjimų trajektoriją** $\vec{X}(t)$), mes galime apskaičiuoti sistemos išėjimą bet kuriuo laiko momentu $t>t_0$ (**sistemos išėjimų trajektoriją** $\vec{Y}(t)$) ir sistemos būseną, bet kuriuo laiko momentu $t>t_0$ (**sistemos fazinę trajektoriją** $\vec{Z}(t)$).

Trajektorijos $\vec{X}(t)$, $\vec{Y}(t)$ ir $\vec{Z}(t)$ matematiškai aprašo sistemos **išorinį funkcionavimą, matematiškai modeliuoja (imituoja) jį**.

Vidinį matematinį sistemos modelį sudaro sistemos **sudėties ir struktūros matematinis aprašymas**.

Sistemos **sudėties matematinį aprašymą** sudaro:

1) sistemos elementų **išskyrimas (nustatymas)** ir jų pavadinimas bei **pažymėjimas (identifikavimas)**,

2) visų sistemos **elementų išorinių matematinių modelių sudarymas**.

Sistemos **struktūros matematinį aprašymą** sudaro:

1) visų sistemos elementų **išėjimo ir įėjimo kontaktų nustatymas ir pažymėjimas**,

2) visų **tarpelementinių ryšių**, o taip pat **elementų ryšių su sistemos aplinka nustatymas ir aprašymas, tų ryšių matematinis atvaizdavimas matricomis bei jungimo operatoriais ar grafais (ERD)**.

Turėdami vidinį matematinį sistemos modelį ir žinodami sistemos elementų pradines būsenas bei sistemos įėjimų trajektoriją $\vec{X}(t)$, mes galime apskaičiuoti kiekvieno sistemos elemento įėjimo, išėjimo ir fazinės trajektorijas, t.y. galime **matematiškai modeliuoti (imituoti) vidinį sistemos funkcionavimą**.

Vidinį matematinį sistemos modelį galima panaudoti ir **sistemos išorinio modelio vaidmenyje**. Tai būna naudinga, kai sistema yra tiek sudėtinga, kad mes negalime iškart ir tiesiogiai sudaryti jos išorinio matematinio modelio, nes negalime nustatyti ir analitiškai išreikšti sistemos išėjimų ir perėjimų funkcijų $\vec{Y} = \vec{F}(\vec{X}, \vec{Z})$, $\vec{Z}(t+\Delta t) = \vec{\psi}(\vec{X}(t), \vec{Z}(t))$ ar $\frac{d\vec{Z}(t)}{dt} = \vec{f}(\vec{X}(t), \vec{Z}(t))$.

Tokiais atvejais, dekomponavę sudėtingą sistemą į paprastus elementus, sudarę jų išorinius matematinius modelius ir sujungę juos taip, kaip sujungti realūs elementai sistemoje, mes gauname vidinį sistemos modelį, kurį turėdami, galime apskaičiuoti ne tik vidines, bet ir

išorines išėjimų bei fazines sistemos trajektorijas, atitinkančias duotąją (arba įsivaizduojamą) sistemos įėjimų trajektoriją $\vec{X}(t)$.

Vidinis matematinis sistemos modelis leidžia gauti jei ne **analitines** (kartais tai pavyksta), tai bent jau **algoritmines sistemos išėjimų ir perėjimų funkcijų išraiškas**, t.y. sudaryti sistemos **išėjimų ir fazinių trajektorijų apskaičiavimo algoritmus**. Kartais tokie algoritmai ne būtinai būna **determinuoti** (vienareikšmiškai apibrėžti), bet ir **stochastiniai**, t.y. modeliuojantys atsitiktinius procesus. Toks modeliavimas vadinamas **statistiniu modeliavimu**.

Čia ką tik aprašyti išoriniai ir vidiniai matematiniai sistemų modeliai priklauso **analiziniu matematinių modelių** klasei, nes jų pagrindą sudaro **analitinės elementų išėjimų ir perėjimų funkcijų išraiškos**. Turėdami tas funkcijas ir žinodami sistemos struktūrą, mes galime sudaryti **visos sistemos išėjimų ir perėjimų funkcijų analitines ar algoritmines išraiškas**. Tačiau ne visada pavyksta sudaryti analizinius sistemų ir jų elementų modelius. Tuomet vietoj analizinių sudaromi **imitaciniai** modeliai, kurie taip pat gali būti išoriniai ir vidiniai.

Imitaciniai modeliai skiriasi nuo analizinių visų pirma tuo, jog analiziniuose modeliuose mes pradžioje surandame išėjimų ar perėjimų funkcijas ir, naudodamiesi jomis, apskaičiuojame vidinio ir išorinio funkcionavimo trajektorijas, o **imitaciniuose modeliuose mes, net nežinodami išėjimų ar perėjimų funkcijų analitinių išraiškų, iškart bandome imituoti sistemų ir jų elementų funkcionavimą**, imituoti sistemos ir jos elementų įėjimų, būsenų ir išėjimų keitimosi procesus.

Imitacinis modeliavimas – tai realioje erdvėje ir realiame laike vykstančių procesų kartojimas (imitavimas, mėgdžiojimas) atitinkamoje matematinėje (informacinėje) erdvėje ir sąlyginiame (pav., „mašiniame“ modeliavimo proceso) laike, kuris nebūtinai turi sutapti su realiuoju laiku.

Statistinis modeliavimas, jei jo metu imituojami atsitiktiniai procesai, taip pat yra ir imitacinis modeliavimas, nors ne visada imitacinis modeliavimas yra statistinis modeliavimas, nes imituoti galima ne tik atsitiktinius, bet ir determinuotus procesus.

Praktikoje taikomose imitacinio modeliavimo sistemose **naudojamas kaip statistinis taip ir determinuotas imitacinis modeliavimas**.

Kaip jau buvo minėta, **imitacinis modeliavimas gali būti taikomas ir ten, kur nepavyksta sudaryti analizinių modelių**, nes imituoti (kartoti) kokius nors veiksmus galima ir tada, kada mes nepajėgūs pilnai suprasti, kodėl ir kaip tie veiksmai atsiranda, nuo ko ir kaip jie priklauso.

Pavyzdžiui, mes šiuo metu dar nepajėgūs sudaryti pakankamai detalų didelės įmonės analizinį sisteminį (vidinį ar išorinį) modelį, tačiau nėra labai sunku imituoti įmonėje vykstančius determinuotus ir atsitiktinius procesus, t.y. sudaryti didelės įmonės imitacinį modelį ir eksperimentuoti su juo. Imitaciniai modeliai leidžia įvertinti sudėtingos sistemos išėjimus, būsenas ir vidinių parametrų reikšmes, jų tikimybes ir t.t., **pakeisdami natūrinius eksperimentus su realia sistema matematiniais eksperimentais su tos sistemos imitaciniu modeliu**. Tai labai naudinga imitacinių modelių ir matematinių eksperimentų su jais savybė, nes natūriniai eksperimentai brangiai kainuoja, dažnai tęsiasi gana ilgai ir kartais būna gana rizikingi ir net pavojingi, kai tuo tarpu matematinių eksperimentų kaina, vykdant tuos eksperimentus, naudojantis kompiuteriais, yra, palyginti, neaukšta, imitavimą kompiuteryje galima atlikti žymiai greičiau, negu realūs procesai vyksta gyvenime, o ir jokios rizikos bei pavojų nėra. Imitacinis modeliavimas net padeda jų išvengti realiame gyvenime, nes leidžia numatyti tos ar kitos veiklos pasekmes. Dėl šios priežasties **eksperimentai su imitaciniais modeliais pradedami vis plačiau taikyti valdyme, norint numatyti galimų valdymo sprendimų pasekmes ir išsirinkti pakankamai gerą sprendimo variantą**.

Sisteminiai modeliai, tiek vidiniai, tiek išoriniai, tiek analiziniai, tiek ir imitaciniai gali būti **deskriptiniai ir normatyviniai**.

Deskriptinis sistemos modelis vaizduoja sistemą tokia, kokia ji, modeliuotojo

nuomone, yra dabar. Deskriptiniai sistemų modeliai sudaromi, tiriant jau funkcionuojančias sistemas, kaupiant informaciją apie jų veiklą, bandant nustatyti jų trūkumus bei galimybes.

Normatyvinis (preskriptyvinis) modelis vaizduoja tai, kas privalėtų būti, tai, ko norėtų to modelio sudarytojai. Normatyviniai sistemų modeliai sudaromi tobulinant jau veikiančias sistemas, o taip pat projektuojant naujas. Todėl normatyviniai modeliai vadinami dar ir **projektais**.

Imitacinis modelis, imituojantis kokios nors dabar veikiančios sistemos funkcionavimą yra deskriptinis modelis, o, pavyzdžiui, pareigybinė instrukcija (pareigybės pasas), aprašantis, ką ir kaip privalėtų daryti pareigūnas, atlikdamas savo pareigas, yra normatyvinis pareigybės modelis. SNOD ir TNOD bei PNOD taip pat priklauso normatyvinių modelių klasei.

Nagrinėdami kokio nors pareigūno veiklą ir jo kokioje nors situacijoje priimamus sprendimus, mes galime sudaryti tris modelius:

- 1) **deskriptinį situacijos modelį – situacinį modelį;**
- 2) **normatyvinį** pareigūno užimamos **pareigybės modelį**, vaizduojantį pareigūno pareigas, teises ir atsakomybę, t.y. pareigybės pasą (PP);
- 3) **motyvacinį** pareigūno kaip realaus individo **modelį**, išreiškiantį pareigūno asmeninius interesus, asmeninius jo tikslus, polinkius bei norus, asmeninius pareigūno poelgių motyvus.

Atsižvelgdamas į šiuos tris modelius, pareigūnas ir priima sprendimą, jei sprendimas priimamas apgalvotai, o ne spontaniškai, stichiškai. Neretai normatyvinis pareigybės modelis ir motyvacinis pareigūno modelis gali daugiau ar mažiau prieštarauti vienas kitam. Tuomet, prieš priimdamas sprendimą, pareigūnas privalo apsispręsti, kam suteikti prioritetą: pareigai ar asmeniniam interesui. Tie pareigūnai, kurie prioritetą atiduoda pareigos jausmui, stengiasi dorai ir atsakingai atlikti savo pareigas, o tie pareigūnai, kurie savo asmeninius interesus laiko svarbesniais negu jų pareigos, dažnai ima piktnaudžiauti užimamomis pareigomis, jų suteikiamomis teisėmis.

Todėl, parenkant pareigūnus, labai svarbu turėti pareigūno deskriptinį modelį, kurį dokumentaliai turėtų fiksuoti pareigūno asmens byla ir jo charakteristikos, ir į kurio sudėtį turėtų įeiti ir pareigūno motyvacinis modelis. Lygindami pareigūno deskriptinį modelį su pareigybės normatyviniu modeliu, mes gana kvalifikuotai galėtume spręsti apie pareigūno atitikimą ar neatitikimą užimamoms pareigoms.

Pareigūnų deskriptiniai modeliai ir pareigybių normatyviniai modeliai gali būti panaudoti ir keliant organizacijos darbuotojų kvalifikaciją, nes leidžia sutelkti dėmesį į deskriptinio ir normatyvinio modelių neatitikimus bei tų neatitikimų sumažinimą.

Valdymo sistemose normatyviniai modeliai sudaromi veiklos **planavimo metu**, o deskriptiniai – **apskaitos metu**. Atsiradus neatitikimams tarp normatyvinių ir deskriptinių trajektorijų (įėjimų, išėjimų, fazinių), gerai suprojektuotose valdymo sistemose turėtų atsirasti **signalinė informacija**, kuri savo keliu turėtų inicijuoti **reguliavimo mechanizmą**, besistengiančių likviduoti tuos neatitikimus, veiklą. Taip, pavyzdžiui, veikia **automatinio ir automatizuoto valdymo sistemų grįžtamojo ryšio mechanizmai**.

Kadangi ne visi asmenys suinteresuoti atskleisti savo motyvacinius modelius (o be jų žinojimo pareigūno deskriptinis modelis negali būti laikomas pakankamai pilnu), tai anksčiau minėtieji ryšiai tarp situacijos deskriptinio modelio, pareigybės normatyvinio modelio, pareigūno motyvacinio modelio ir pareigūno priimamo sprendimo duotoje situacijoje gali būti panaudoti pareigūno motyvacinio modelio nustatymui. Žinodami situacijos modelį, pareigybės normatyvinį modelį ir pareigūno priimtą sprendimą, mes, spręsdami loginių lygčių sistemą, galime apskaičiuoti, kokiais motyvais vadovavosi pareigūnas, priimdamas sprendimą duotoje situacijoje. Sukaupę pakankamai daug tokio tipo statistinės informacijos, mes galime gana tiksliai nustatyti pareigūno motyvacinį modelį ir spręsti apie pareigūno tinkamumą užimamoms pareigoms. Čia aprašytą metodą galima būtų panaudoti, atrenkant tinkamiausiais kandidatūras į aukštus valstybės postus ir atmetant netinkamus kandidatus, o tai visais laikais buvo ir liks gana aktualu.

Kaip jau buvo minėta šio skyrelio pradžioje, be sisteminių modelių, kuriuos mes trumpai jau aprašėme, egzistuoja ir taip vadinami optimizaciniai modeliai. Jie sudaromi, sprendžiant optimizavimo uždavinius, ieškant optimalių sistemos parametrų reikšmių, optimalių jų funkcionavimo trajektorijų ir pan.. Tai gana svarbu sistemų projektavime, jų **optimalių normatyvinių modelių** kūrimė.

Sudarant **optimizacinį modelį**, pradžioje pasirenkamas tas sistemos **efektyvumo rodiklis**, kurio atžvilgiu sistema bei jos veikla bus optimizuojama. Pasirinkus tą rodiklį, vadinamą **optimalumo rodikliu**, nustatomas **optimalumo kriterijus** – tas požymis (ar taisyklė), kuo naudodamiesi sprendžiame, kada optimalumo rodiklis pasiekia optimalią reikšmę. Dažniausiai optimalumo kriterijumi pasirenkamas optimalumo rodiklio maksimumas ar minimumas (pavyzdžiui, pelno maksimumas, išlaidų minimumas ir pan.) Žinodami optimalumo rodiklį, nagrinėjame, nuo kokių **valdomų bei nevaldomų kintamųjų** tas rodiklis priklauso, ir bandome nustatyti bei matematiškai išreikšti tą priklausomybę, t.y. sudaryti taip vadinamąją **tikslo funkciją**. Po to nustatomos valdomųjų kintamųjų leistino kitimo ribos, kurios matematiškai išreiškiamos kaip nelygybių bei lygybių sistema, vadinama **apribojimų sistema**. Iš to, kas buvo pasakyta apie optimizacinio modelio sudarymą, išplaukia, jog **optimizacinį modelį sudaro:**

1) **tikslo funkcija** – optimalumo rodiklio priklausomybės nuo valdomųjų ir nevaldomųjų kintamųjų matematinė išraiška – su optimalumo kriterijaus nurodymu, pavyzdžiui:

$$z = F(\vec{X}, \vec{Y}) \rightarrow \min_{\vec{x}}, \text{ kur } z - \text{optimalumo rodiklis, } \vec{X} - \text{valdomųjų kintamųjų vektorius, } \vec{Y} - \text{nevaldomųjų kintamųjų vektorius, } z \rightarrow \min_{\vec{x}} - \text{optimalumo kriterijus, } F - \text{optimalumo rodiklio } z \text{ priklausomybės nuo } \vec{X} \text{ ir } \vec{Y} \text{ matematinė išraiška;}$$

2) **apribojimai** – nelygybių bei lygybių sistema, kurią turi tenkinti valdomųjų kintamųjų vektoriaus $\vec{X} = (x_1, x_2, \dots, x_i, \dots, x_m)$ koordinačių x_i reikšmės.

Kai kuriais atvejais optimizacinis modelis gali ir neturėti apribojimų sistemos, t.y. turėti savo sudėtyje tik tikslo funkciją.

Sprendžiant optimizavimo uždavinį, prognozuojamos nevaldomųjų kintamųjų vektoriaus $\vec{Y} = (y_1, y_2, \dots, y_l, \dots, y_s)$ koordinačių y_l reikšmės, o po to, naudojantis atitinkamais metodais (pav. tiesinio, netiesinio, diskretaus, dinaminio, stochastinio programavimo), ieškomos tokios valdomųjų kintamųjų vektoriaus $\vec{X} = (x_1, x_2, \dots, x_i, \dots, x_m)$ koordinačių reikšmės $x_i^{(opt)}$, kurios tenkintų apribojimų sistemą ir optimizuotą (minimizuotą ar maksimizuotą) tikslo funkcijos reikšmę, esant prognozuojamoms nevaldomųjų kintamųjų reikšmėms.

Neretai, sudarius sudėtingą optimizacinį modelį, optimizavimo uždavinį, naudojantis analitiniais optimizavimo metodais (algoritmais), nepavyksta išspręsti. Tada naudojamos matematiniai eksperimentai su optimizaciniu modeliu metodu, ir optimalus sprendinys (tikslus ar artutinis) ieškomas, pasitelkiant imitacinį modeliavimą.

Tokiu būdu, iš vienos pusės, optimizaciniai modeliai ir optimizavimo uždavinių sprendimas taikomi, projektuojant normatyvinius sisteminius modelius bei tobulinant funkcionuojančias sistemas, o iš kitos pusės, sprendžiant optimizavimo uždavinius, neretai tenka naudotis sisteminiiais modeliais ir matematiniais eksperimentais su sisteminiiais ir optimizaciniais modeliais metodu. **Sisteminiai ir optimizaciniai modeliai vienas kitą papildo.** Tai ypatingai ryškiai pastebime ne tik sistemų projektavime, bet ir jau minėtame operacijų tyrime, optimizuodami sudėtingų sistemų funkcionavimą.

5. SISTEMINĖ TYRIMŲ, MODELIAVIMO IR PROJEKTAVIMO METODOLOGIJA. SISTEMINIS MĄSTYMAS.

Sisteminė metodologija susiformavo, kai tikslieji mokslai pradėjo tirti, modeliuoti ir

projektuoti ne tik paprastas, bet ir sudėtingas sistemas.

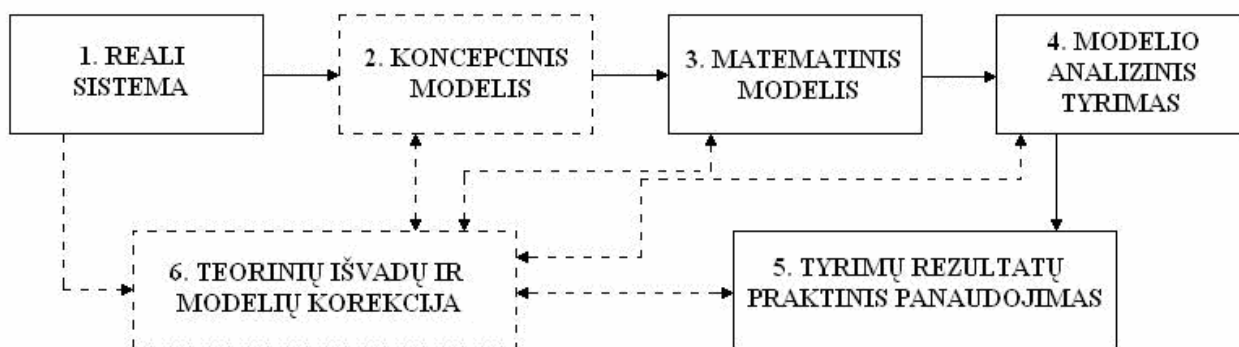
Sistema vadinama **paprasta**, jei ją vienas žmogus gali pakankamai pilnai ir giliai apžvelgti savo intelektu, t.y. per praktikoje priimtina laiką gali susidaryti pakankamai adekvatų tikrovei tos sistemos koncepcinį modelį, o reikalui esant – ir matematinį modelį.

Sistema vadinama **sudėtinga**, jei ją vienas žmogus jau nebepajėgus aprėpti savo intelektu, t.y. nebepajėgus tiesiogiai (be tarpinių etapų ir pagalbinių priemonių) susidaryti pakankamai adekvatų tikrovei tos sistemos koncepcinį, o po to – ir matematinį modelį.

Paprastos ir sudėtingos sistemos sąvokos yra **reliatyvios**, nes viena ir ta pati sistema vienam žmogui gali būti sudėtinga, o kitam paprasta; sudėtinga sistema, laikui bėgant ir įsigijus naujų žinių, gali tapti paprasta, o kartais gali atsitikti ir taip, kad, laikui bėgant ir įsigyjant naujų žinių, paaiškėja, jog ta sistema, kurią laikėme paprasta, tikrovėje yra sudėtinga.

Tačiau šis paprastos ir sudėtingos sistemos sąvokų reliatyvumas pats yra reliatyvus, nes kiekvienu konkrečiu momentu konkrečiam žmogui kokia nors konkreti sistema **objektyviai** yra arba tik paprasta, arba sudėtinga.

Paprastų sistemų tyrimui tikslieji mokslai naudojami taip vadinama **klasikine tyrimų metodologija**, kuri schematiškai pavaizduota 3 pav.



3 pav.

Šioje scheme blokas, vaizduojantis koncepcinio modelio sudarymą, pavaizduotas punktyru, nes, kuriant paprastos sistemos matematinį modelį, susidaro išpūdis, jog tos sistemos matematinis modelis gaunamas lyg ir tiesiogiai be tarpinio koncepcinio modeliavimo etapo. Taip atsitinka todėl, kad koncepcinis modeliavimas, kuris vyksta žmogaus galvoje, kuriant paprastos sistemos matematinį modelį, vyksta beveik automatiškai ir beveik kartu su matematinio modelio sudarymu. Jis neišskiriamas į atskirą mąstymo etapą ir niekur, išskyrus žmogaus sąmonę bei pasąmonę, nefiksuojamas. Dažnai žmogus net neatkreipia dėmesio į tai, kad jis kuria paprastos sistemos koncepcinį modelį: jam atrodo, jog jis, nagrinėdamas realią paprastą sistemą, iškart gauna jos matematinį modelį.

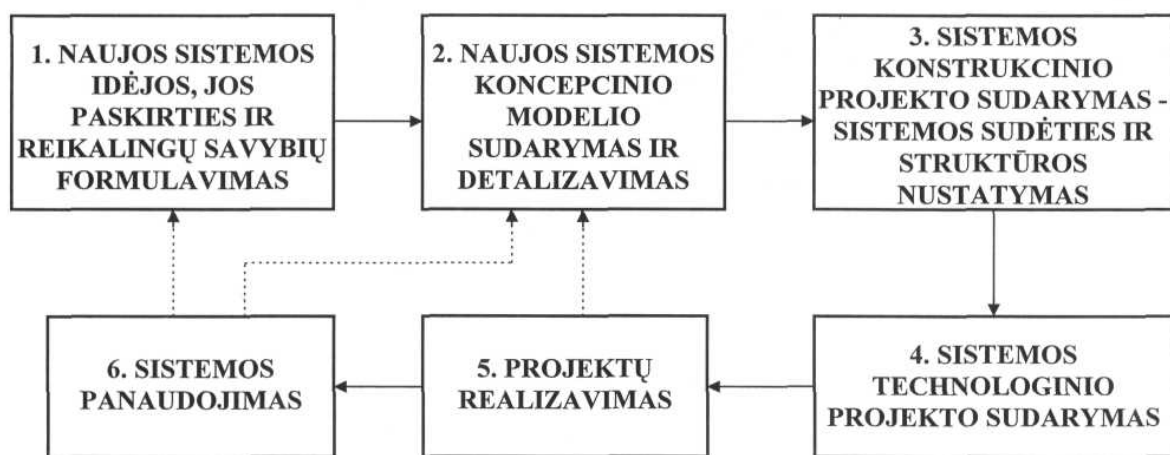
Analizuodamas matematinį modelį (o kartu ir galvoje esantį koncepcinį modelį), žmogus padaro tas ar kitas išvadas, kurias po to gali taikyti savo ir kitų žmonių praktinėje veikloje (žr. nagrinėjamos schemos ketvirtąjį ir penktąjį blokus).

Šeštasis blokas taip pat pavaizduotas punktyru, nes jis nagrinėjamoje scheme ne visada yra būtinas: paprasta sistema, todėl ir vadinama paprasta, kad, ją nagrinėjant, neretai iškart pavyksta susidaryti pakankamai adekvatų jos koncepcinį bei matematinį modelius, kurių koreguoti jau nebereikia.

Tačiau dažnai šeštasis blokas būna reikalingas, nes neretai praktinėje veikloje susiduriame su kai kuriomis iškart nepastebėtomis paprastos sistemos savybėmis, į kurias atsižvelgdami, mes šiek tiek pakoreguojame susidarytą sistemos įvaizdį ir jos matematinį modelį bei iš to modelio analizės išplaukiančias išvadas.

Paprastų sistemų **projektavimo klasikinė metodologija** taip pat gana paprasta:

pradžioje nustatoma projektuojamos sistemos paskirtis ir tos jos savybės, kurios reikalingos, kad sistema atitiktų savo paskirtį, o po to sudaromas tos sistemos normatyvinis modelis – projektas, kuris pradžioje atsiranda projektuotojo galvoje kaip projektuojamos sistemos koncepcija, po to palaipsniui detalizuojasi, kol, galų gale, tampa būsimos sistemos **inžineriniu konstruktoriniu projektu** (ar matematiniu modeliu, kuris vėliau išreiškiamas inžineriniais terminais). Sudarius konstruktorinį projektą, sprendžiami suprojektuotos sistemos realizavimo (pagaminimo) klausimai ir sudaromas konstruktorinio projekto **realizavimo technologinis projektas**. Projektuojant paprastas sistemas, konstruktorinis ir technologinis projektavimas neretai vyksta kartu ir vienas kitą papildo ir koreguoja. Turėdami konstruktorinį ir technologinį projektus, mes galime juos realizuoti ir sukurtą naują realią sistemą panaudoti savo reikmėms. Naudojimo metu gali kilti sukurtos sistemos patobulinimo idėjos ir gali susidaryti patobulintos sistemos koncepcinis modelis, kuris, reikalui esant, gali būti realizuotas jau aprašytu būdu. Paprastos sistemos projektavimo blokinė schema pavaizduota 4 pav.



4 pav.

Klasikinės sistemų tyrimo ir projektavimo metodologijos, pavaizduotos 3 ir 4 pav., jau nebepakanka, nagrinėjant ir projektuojant sudėtingas sistemas, nes sudaryti tokių sistemų pakankamai detalius ir pilnus koncepcinius ir matematinius modelius dėl sudėtingų sistemų neapžvelgiamumo iškart nepavyksta. Todėl, prieš sudarant sudėtingos sistemos modelį, tą sistemą tenka **dekomponuoti** į posistemius, o šiuos – į smulkesnius posistemius ir, galų gale, į elementus, sudaryti elementų išorinius matematinius modelius, o po to įvykdyti sistemos vidinio modelio **struktūrinę sintezę** – elementų matematinius modelius sujungti taip, kad vidinio modelio struktūra sutaptų su realios sistemos struktūra. Eksperimentuodami su sudarytu vidiniu matematiniu modeliu, kuris kartais gali turėti analizinio, o kartais ir imitacinio modelio formą, mes galime gauti vienokias ar kitokias išvadas apie modelio funkcionavimą. Lygindami matematinių eksperimentų su modeliu rezultatus su informacija apie realios sistemos funkcionavimą, mes galime spręsti apie sudaryto sistemos vidinio modelio adekvatumą tikrovei ir, reikalui esant, tą modelį koreguoti. Dėl modeliujamos sudėtingos sistemos sudėtingumo beveik niekada nepavyksta iškart gauti pakankamai tobulą jos modelį, todėl **sudėtingos sistemos modelio kūrimo proceso iteratyvumas – būdinga sudėtingų sistemų modeliavimo savybė**. Kuriant sudėtingos sistemos deskriptinį modelį ar naujos sudėtingos sistemos projektą, tokių iteracijų, kuomet, analizuodami vėlesnių to kūrimo etapų rezultatus, mes susiduriame su reikiamybe grįžti į ankstesnius etapus ir koreguoti juos, o po to – ir į po jų sekančius etapus ir juos koreguoti, būna gana daug, ir niekuomet negalima garantuoti, jog neprireiks naujų iteracijų. Neveltui sudėtingų sistemų projektuotojų tarpe paplitęs toks liūdnokas aforizmas: „Kiekviena

paskutinė projektavimo klaida yra priešpaskutinė“. Tačiau ši liūdnoka perspektyva nėra beviltiška: modeliavimo bei projektavimo procese koreguojančių iteracijų dažnis palaipsniui mažėja, ir, galų gale, mes gauname pakankamai mus tenkinantį modelį (projektą). Tokį projektą jau galima bandyti realizuoti, o su modeliu – eksperimentuoti, prognozuojant, kaip reali sistema funkcionuos tomis ar kitomis sąlygomis. **Benagrinėjant sudarytą vidinį sudėtingos sistemos deskriptinį modelį ir jo funkcionavimą, neretai pavyksta žymiai giliau ir detaliau suprasti realioje sistemoje vykstančius procesus, surasti jų dėsningumus ir, galų gale, susidaryti pakankamai pilną ir pakankamai tikslų sudėtingos sistemos koncepcinį modelį.** Tokiais atvejais sudėtinga sistema jos modeliuotojams tampa paprasta. Šis reiškinys yra labai pageidautinas, bet nėra būtinas: **su sudėtingos sistemos vidiniu modeliu galima operuoti (pavyzdžiui, eksperimentuoti) ir neturint pakankamai pilno tos sistemos koncepcinio modelio.**

Sudėtingų sistemų modeliavimą ir projektavimą, teoriškai mąstant, galėtų atlikti vienas žmogus, tačiau praktiškai, taupant laiką ir atsižvelgiant į tai, jog kiekvieno žmogaus gyvenimas ir galimybės yra riboti, taip nedaroma: sudėtingų sistemų modeliavimui ir projektavimui sudaromi **specialūs modeliuotojų bei projektuotojų kolektyvai** (kartais net gana dideli).

Yra dar ir kita priežastis, dėl kurios sudėtingų sistemų modeliavimas ir projektavimas turi būti kolektyvinis darbas: dauguma sudėtingų sistemų yra **polifunkcionalios** (atliekančios daug įvairių funkcijų) ir tokios sistemos gali būti nagrinėjamos **daugeliu įvairių aspektų**, o tam reikalingi **įvairių sričių specialistai**, nes šiuolaikinis mokslo ir technikos išsivystymo lygis jau yra toks, kad vienas žmogus jau nebegali būti visų sričių specialistu. Iš čia išplaukia, kad sudėtingų sistemų tyrimo, modeliavimo ir projektavimo kolektyvai turi būti **kompleksiniai**, t.y. sudaryti iš įvairių sričių specialistų, o ir patys **sudėtingų sistemų modeliai ir projektai taip pat turi būti kompleksiniai, daugiaspektiniai**, t.y. ne tik atskirų elementų modelių jungimo į posistemių modelius, o šių – į sudėtingos sistemos vidinį modelį, bet ir **įvairių aspektų sintezės** rezultatas: skirtingų profesijų specialistų sudaryti vienaaspektiniai modeliai ir projektai turi būti derinami ir jungiami į bendrą **sisteminį sudėtingos sistemos modelį ar projektą.**

Kompleksiniuose modeliuotojų ir projektuotojų kolektyvuose iškyla **bendravimo ir susikalbėjimo** bei **darbo organizavimo** problemos.

Susikalbėjimo problema atsiranda todėl, kad įvairių sričių specialistai naudojami savo sričių **specialia terminologija**, kuri nėra žinoma kitų sričių specialistams. Todėl kompleksinių kolektyvų nariams tenka ilgainiui padirbėti kartu, kol jie pradeda pakankamai gerai vienas kitą suprasti. Savitarpio supratimo atsiradimui gali padėti **bendrasisteminė terminologija**, kurios sukūrimas ir įdiegimas įvairiose sferose – vienas iš svarbių jau minėtosios bendrosios sistemų teorijos uždavinių, kurių sprendžiant jau nemažai padaryta. Vienas šios metodinės priemonės tikslų yra SIS kūrėjų kolektyvo supažindinimas su kai kuriais bendrasisteminiais terminais, kurie reikalingi sisteminės modeliavimo ir projektavimo metodologijos įsisavinimui.

Organizuojant kompleksinio kolektyvo darbą, vien tik specialistų nebepakanka. Kompleksiniame kolektyve turi būti bent vienas **dženeralistas**. Dženeralistas skiriasi nuo specialisto tuo, kad specialistas gana detalai ir gana giliai žino ir supranta kokią nors vieną specialią (ir todėl gana siaurą) sritį, o dženeralistas privalo orientotis (nors galbūt nepakankamai žinodamas detales) įvairiose srityse, mokėti susikalbėti su įvairiais specialistais, suprasti juos, derinti ir organizuoti įvairių specialistų darbą, siekiant bendrų visam kompleksiniam kolektyvui tikslų.

Kad žmogus galėtų tapti dženeralistu, jis turi ne tik įsigyti gana daug gana įvairių žinių, bet ir sugebėti jas sujungti į specialaus mąstymo, vadinamo **sisteminio mąstymu**, sistemą, kuri leistų tam žmogui konstruoti sudėtingas sistemas ir organizuoti jų kūrimą. Sisteminio mąstymo pagrindai reikalingi ir specialistams – kompleksinio kolektyvo nariams, kuriantiems sudėtingos sistemos modelį ar projektą. Su kai kuriais sisteminio mąstymo

principais (fundamentiniais ir instrumentiniais) galima susipažinti dviejose šios metodinės priemonės autoriaus knygelėse.¹ Pirmoje knygelėje išdėstyta 30 bendrasisteminių fundamentinių ir instrumentinių sisteminio mąstymo ir sisteminės modeliavimo bei projektavimo metodologijos principų, o antroje knygelėje išdėstyti principai, kurių žinojimas naudingas sudėtingų automatizuoto valdymo sistemų (AVS) organizacijose kūrėjams, tame skaičiuje ir organizacijų vadovams. Visiems, norintiems įsisavinti sisteminio mąstymo ir sudėtingų sistemų tyrimo, modeliavimo ir projektavimo sisteminę metodologiją, labai rekomenduotina susipažinti su tais principais, kurių aprašymas šioje metodinėje priemonėje užimtų per daug vietos. Čia apsiribosiu tik pastaba, jog sisteminio mąstymo esmę sudaro sugebėjimas, nagrinėjant bet kokią objektą ar reiškinį, matyti jame sistemą, egzistuojančią erdvėje ir laike, kuri savo ruožtu susideda iš mažesnių sistemų ir sąveikaudama su eile kitų sistemų, sudaro dar sudėtingesnę sistemą, bei iš to sugebėjimo išplaukiantis suvokimas, jog, norint pilnai suprasti nagrinėjamą sistemą, reikia nagrinėti ne tik tą sistemą, bet ir joje vykstančių procesų sąryšį su procesais, vykstančiais jos sudėtinėse dalyse ir toje dar sudėtingesnėje sistemoje, į kurią nagrinėjamoji sistema įeina kaip sudėtinė dalis. Toks sugebėjimas įsigijamas ne iškart ir nelengvai, tačiau jį įsigijęs žmogus pradeda žymiai giliau ir žymiai lengviau susivokti jį supančiame sudėtingame pasaulyje, nes pastebėti ir išsiaiškinti sisteminiai dėsniumai padeda pastebėti ir suvokti visame kame tam tikrą tvarką ir priežastinius ryšius, kas žmogui, nesugebančiam sistemiškai mąstyti, gali atrodyti kaip nesuvokiamas chaosas, kuriame nėra jokių dėsniumų, ribojančių galinčius tame „chaose“ vykti procesus. Žmogus, nesugebantis sistemiškai mąstyti, gali ir nepastebėti to, kad, iš vienos pusės, koks nors objektas, tapdamas kokios nors sistemos elementu, netenka dalies savo „elgesio“ laisvių ir todėl jo veikla tampa lengviau prognozuojama, ir kad, iš kitos pusės, objektai, jungdamiesi vienas su kitu, emerđžencijos reiškinį dėka gali „pagimdyti“ visai naujus objektus su naujomis sisteminėmis savybėmis, kurių nebuvo besijungiančiuose objektuose. Todėl toks žmogus, stebėdamas kokį nors sudėtingą reiškinį, neretai pastebi ir suvokia žymiai mažiau, negu žmogus, sugebantis sistemiškai mąstyti. **Sistemiškai mąstančiam žmogui pasaulis atrodo, iš vienos pusės, žymiai sudėtingesnis, nes toks žmogus pastebi žymiai daugiau pasaulyje egzistuojančių ryšių, iš kitos pusės – žymiai suprantamesnis, žymiai lengviau prognozuojamas, net kokia tai prasme paprastesnis, nes sistemiškai mąstantis žmogus išvelgia iš sisteminių ryšių išplaukiančius dėsniumus, ribojančius galimų situacijų spektrą ir surišančius atskiras dalis į vieningą visumą.**

Nagrinėjant ir modeliuojant sudėtingas sistemas, vietoj schemas, pavaizduotos 3 pav., tenka naudotis žymiai sudėtingesne schema, pavaizduota 5 pav.

Šioje schemoje panaudotų mnemoninių žymėjimų reikšmės:

RS – nagrinėjamoji reali sistema;

SA – nagrinėjamos sistemos aplinka;

IP – iteracinis procesas, kurio metu, naudojantis ankstesnių (turinčių mažesnius numerius) blokų rezultatais, bandoma gauti vėlesnio bloko rezultatus, į kuriuos atsižvelgiant, koreguojami ankstesni blokai, po to – vėlesnis blokas, po to vėl grįžtama į ankstesnius blokus ir t.t., kol gaunami pakankamai suderinti rezultatai;

SAD – nagrinėjamos sistemos sisteminė analizė ir dekompozicija, kurios metu išskiriami sistemos posistemiai ir elementai (juos ir vaizduoja pats blokas SAD); sisteminės analizės ir dekompozicijos procesą, kuris gali vykti įvairiais būdais, pav., naudojantis funkciniu ar/ir objektiniu principu, vaizduoja iteracinis procesas IP, jungiantis RS ir SAD blokus;

¹ 1. A.Šukys. Sistemnyj podchod i jego osnovnyje principy. Metodičeskaja razrabotka po kursu „Teorija sistem“. Vypusk I. Barnaul:API, 1980 (rusų k.)

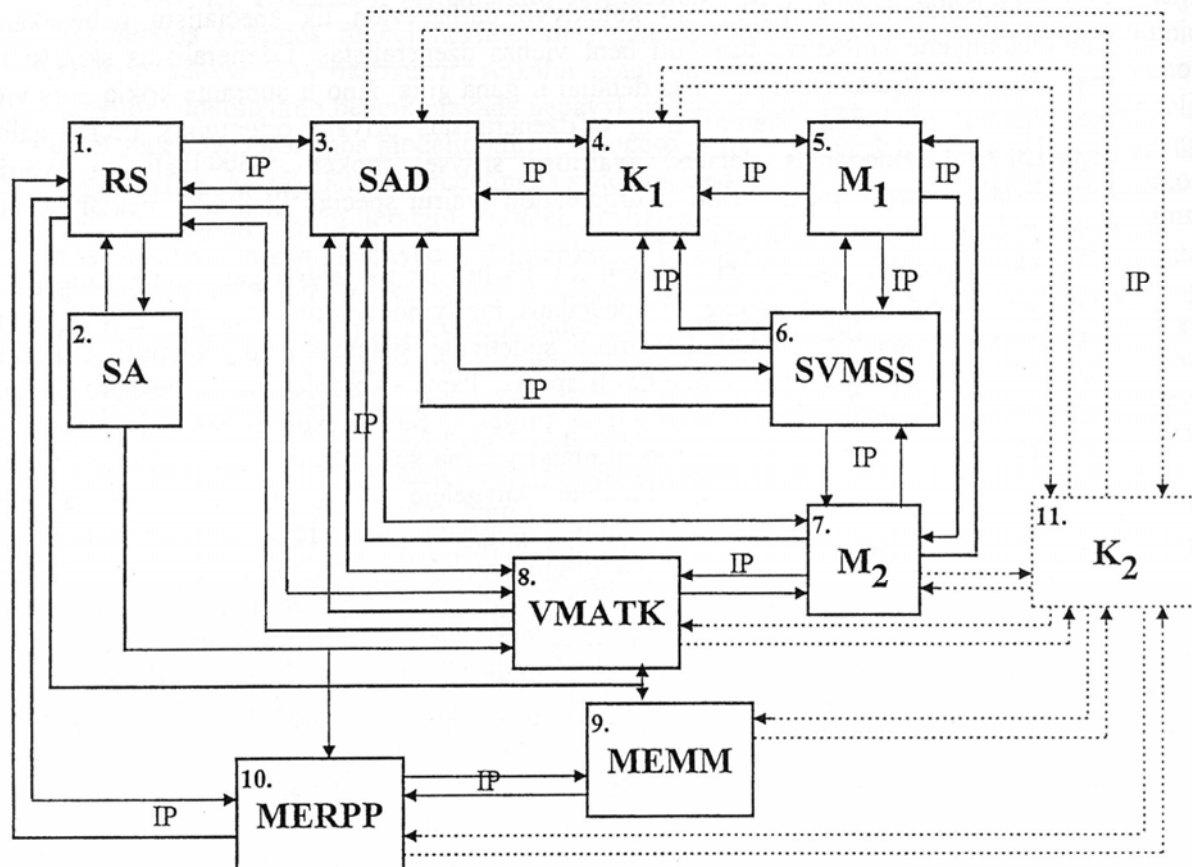
2. A.Šukys. Pirmosios kartos ĮAVS neefektyvumo priežastys. Antrosios kartos ĮAVS projektavimo principai. Vilnius: VU, 1987 (lietuvių kalba).

K_1 – nagrinėjamos sistemos elementų koncepciniai modeliai, pirmosios konceptualizacijos, vykstančios kaip iteracinis procesas IP tarp SAD ir K_1 bloků, rezultatas; ir K_1 , ir SAD vėliau gali būti koreguojami, sprendžiant vėlesnių bloků problemas;

M_1 – nagrinėjamos sistemos elementų išoriniai matematiniai modeliai, kurie sudaromi iteracinio proceso IP, vykstančio tarp K_1 ir M_1 bloků, metu;

SVMSS – nagrinėjamos sistemos vidinio modelio struktūrinės sintezės rezultatas – sistemos vidinio modelio struktūra; pačios struktūrinės sintezės procesą vaizduoja IP, jungiantys SVMSS bloką su SAD, K_1 ir M_1 blokais;

M_2 – nagrinėjamos sistemos vidinis matematinis modelis, kuris gaunamas jungiant elementų (ar juose vykstančių procesų) matematinius modelius M_1 į struktūrą, gautą SVMSS bloke; modelio M_2 gavimas ir tobulinimas – taip pat gana sudėtingas iteracinis procesas;



5 pav.

VMATK – vidinio nagrinėjamos sistemos modelio atitikimo tikrovei tikrinimo ir korekcijos blokas, vaizduojantis tas M_2 korektyvas, kurios atsiranda, lyginant M_2 funkcionavimą su RS funkcionavimu besikeičiančioje aplinkoje SA;

MEMM – matematinių eksperimentų su matematiniais modeliais M_2 ir M_1 blokas; tie eksperimentai gali būti ir išoriniai, ir vidiniai; jų paskirtis – padėti spręsti RS tobulinimo bei praktinio panaudojimo problemas, prognozuoti tų ar kitų išorinių ar/ir vidinių poveikių į RS pasekmes; blokas MEMM vaizduoja ne tik pačius eksperimentus bei matematinio eksperimentavimo reikmėms sudarytą kompiuterinę sistemą, bet ir tų eksperimentų rezultatus: gautas išvadas ir rekomendacijas;

MERPP – matematinių eksperimentų rezultatų praktinis panaudojimas, t.y. tie vidiniai ir išoriniai poveikiai į RS, kuriuos realizuodami siekiame RS tyrėjams ir jų užsakovams reikiamų tikslų;

K_2 – koncepcinis visos sudėtingos sistemos RS modelis; jis ir jo ryšiai pažymėti punktyrais, nes RS tyrimo ir modeliavimo metu gali pavykti susidaryti K_2 (o tai labai pageidautina, nes labai padeda realizuojant daugelį kitų šios sistemos bloků, koreguojant

juos), tačiau gali ir nepavykti, ir tuomet sisteminio tyrimo ir modeliavimo procesas funkcionuoja be bloko K_2 .

Sisteminė tyrimo ir modeliavimo metodologija skiriasi nuo klasikinės visų pirma tuo, kad reikalauja nagrinėti sistemą RS kartu su jos aplinka SA; klasikinė metodologija to nereikalauja, nors, žinoma, ir nedraudžia.

Antrąjį tų metodologijų skirtumą sudaro tai, kad sudėtingų sistemų tyrimas ir modelių sudarymas yra iteracinis procesas, o paprastoms sistemoms tai nėra būtina, o tik leistina.

Trečiasis skirtumas – sudėtingų sistemų dekompozicija, prieš sudarant jų modelius. Kadangi, dekomponuojant sistemą, nutraukiami kai kurie vidiniai jos ryšiai ir tuo pačiu metu prarandamos kai kurios sisteminės savybės, atsirandančios sistemos dalims jungiantis į sistemą, tai reikia stengtis pasirinkti tokią dekompoziciją, kuri nutrauktų kuo mažiau esminių ryšių. Neiškart ir nevisuomet pavyksta surasti tinkamą dekompoziciją, todėl neretai dekompozicija vykdoma keliais būdais, analizuojant sudėtingą sistemą įvairiais aspektais. Sudaromi keli sudėtingos sistemos vidinio modelio variantai, kurie vėliau sintezuojami į kompleksinį modelį, tokiu būdu stengiantis kuo pilniau atvaizduoti realią sistemą, palikti kuo mažiau nepastebėtų ir neatvaizduotų modelyje M_2 realios sistemos RS vidinių ryšių.

Daugaspektinė sudėtingos sistemos analizė ir po jos sekanti tos sistemos modelio sintezė reikalauja daug laiko ir įvairių sričių specialistų, o iš to išplaukia ketvirtasis sisteminės ir klasikinės metodologijų skirtumas: sisteminės metodologijos realizavimui reikalingi kompleksiniai kolektyvai, sudaryti iš įvairių sričių specialistų, kuriems vadovauja dženeralistas, o klasikinės metodologijos realizavimui dažniausiai pakanka vieno specialisto, nors leidžiamas ir kelių specialistų dalyvavimas.

Penktąjį skirtumą sudaro tai, kad paprastos sistemos koncepcinis modelis susidaro žmogaus galvoje anksčiau už matematinį, o sudėtingos sistemos koncepcinis modelis K_2 gali būti sudarytas tik po to, kai jau sudarytas tos sistemos matematinis modelis M_2 , o kartais tenka operuoti su sudėtingos sistemos matematiniu modeliu, net nesusidarius pakankamai adekvataus jos koncepcinio modelio.

Šeštojo skirtumo esmė ta, kad, nagrinėjant paprastą sistemą, dažniausiai sudaromas tik jos išorinis matematinis modelis ir jau po to, reikalui esant, jos vidinis modelis, o nagrinėjant sudėtingą sistemą, pradžioje sudaromas jos vidinis matematinis modelis, kuris gaunamas, jungiant tos sistemos elementų išorinius modelius, ir tik po to tas vidinis modelis gali būti panaudotas išorinio modelio vaidmenyje ar išorinio sistemos modelio išvedimui.

Sisteminė metodologija būtinai reikalauja gauto sudėtingos sistemos matematinio modelio adekvatumo tikrinimo, pravedant specialius eksperimentus, o paprastų sistemų matematiniai modeliai tikrinami, tik bandant juos taikyti praktikoje bei teorijoje, ir specialių tikrinančių eksperimentų pravedimas jiems nėra būtinas, nors kartais tokie eksperimentai ir yra daromi. Tai septintasis sisteminės ir klasikinės metodologijų skirtumas.

Gana artimas šiam yra ir aštuntasis šių metodologijų skirtumas, išplaukiantis iš sisteminio mąstymo principų: mes niekad nesame tikri, kad sudėtingos sistemos modelis yra pakankamai adekvatus tikrovei, nes prie absoliučios tiesos galima tik vis labiau artėti, tačiau pasiekti tokią būseną, kurioje galėtume būti pilnai įsitikinę, jog tą absoliučią tiesą jau žinome, niekad neįmanoma. Bet koks mūsų žinojimas yra reliatyvus. Gilesnis to reliatyvumo suvokimas ir atsižvelgimas į jį sistemų tyrimo, modeliavimo ir projektavimo praktikoje taip pat skiria sisteminę metodologiją nuo klasikinės, sisteminį mąstymą nuo dogmatinio mąstymo.

Nors sisteminė ir klasikinė metodologijos žymiai skiriasi, jos nėra antagonistinės. Sisteminė metodologija yra klasikinės metodologijos vystymosi, jos papildymo ir koregavimo, pritaikant sudėtingų sistemų tyrimo, modeliavimo ir projektavimo reikmėms, rezultatas. Klasikinė gi metodologija yra dalinis sisteminės metodologijos atvejis, jos supaprastintas variantas, kuriuo galima naudotis, modeliuojant bei projektuojant paprastas sistemas.

Tačiau ir sisteminė metodologija nėra visagalė. Egzistuoja tokios sistemos, kurioms mes šiandien turimomis jėgomis per praktiškai priimtina laiką dar nesugebėtume sudaryti pakankamai adekvačius matematinius modelius, net ir naudodamiesi sistetine metodologija bei visais šiuolaikiniais mokslo ir technikos pasiekimais. Tokias sistemas vadinsime **labai sudėtingomis**, arba **supersudėtingomis**. Supersudėtingos sistemos sąvoka taip pat yra reliatyvi (kaip ir paprastos bei sudėtingos sistemų sąvokos), nes, laikui bėgant, kai kurios supersudėtingos sistemos gali tapti sudėtingomis, o kartais – net ir paprastomis. Tačiau visuomet egzistuoja ir egzistuos tokios sistemos, kurios duotuoju momentu duotosioms pajėgoms bus supersudėtingos.

Susipažinę su sudėtingų sistemų deskriptinių modelių sudarymo sisteminės metodologijos pagrindiniais bruožais, pereisime prie sudėtingų sistemų projektavimo – jų normatyvinių modelių sudarymo – metodologijos aprašymo.

Projektuojant sudėtingas sistemas, klasikinės metodologijos, kurios panaudojimo projektavime blokinę schemą vaizduoja 4 pav., jau nebepakanka. Tenka pereiti prie sisteminės metodologijos, ir projektavimo procesas tampa žymiai sudėtingesnis. Jo blokinę schemą vaizduoja 6 pav.

Šioje schemoje panaudotų mnemoninių žymėjimų reikšmės:

NSA – naujos (projektuojamos) sistemos aplinka, kurioje turės funkcionuoti suprojektuota sistema ir iš kurios analizės išplaukia naujos sistemos sukūrimo tikslingumo idėja;

NSIF – naujos sistemos idėjos (jos misijos) ir įvaizdžio (vizijos) formavimas ir formulavimas; blokas NSIF kaip tik vaizduoja tą idėją ir įvaizdį, o jų formavimo ir formulavimo procesą vaizduoja iteracinis procesas IP, jungiantis blokus NSA ir NSIF;

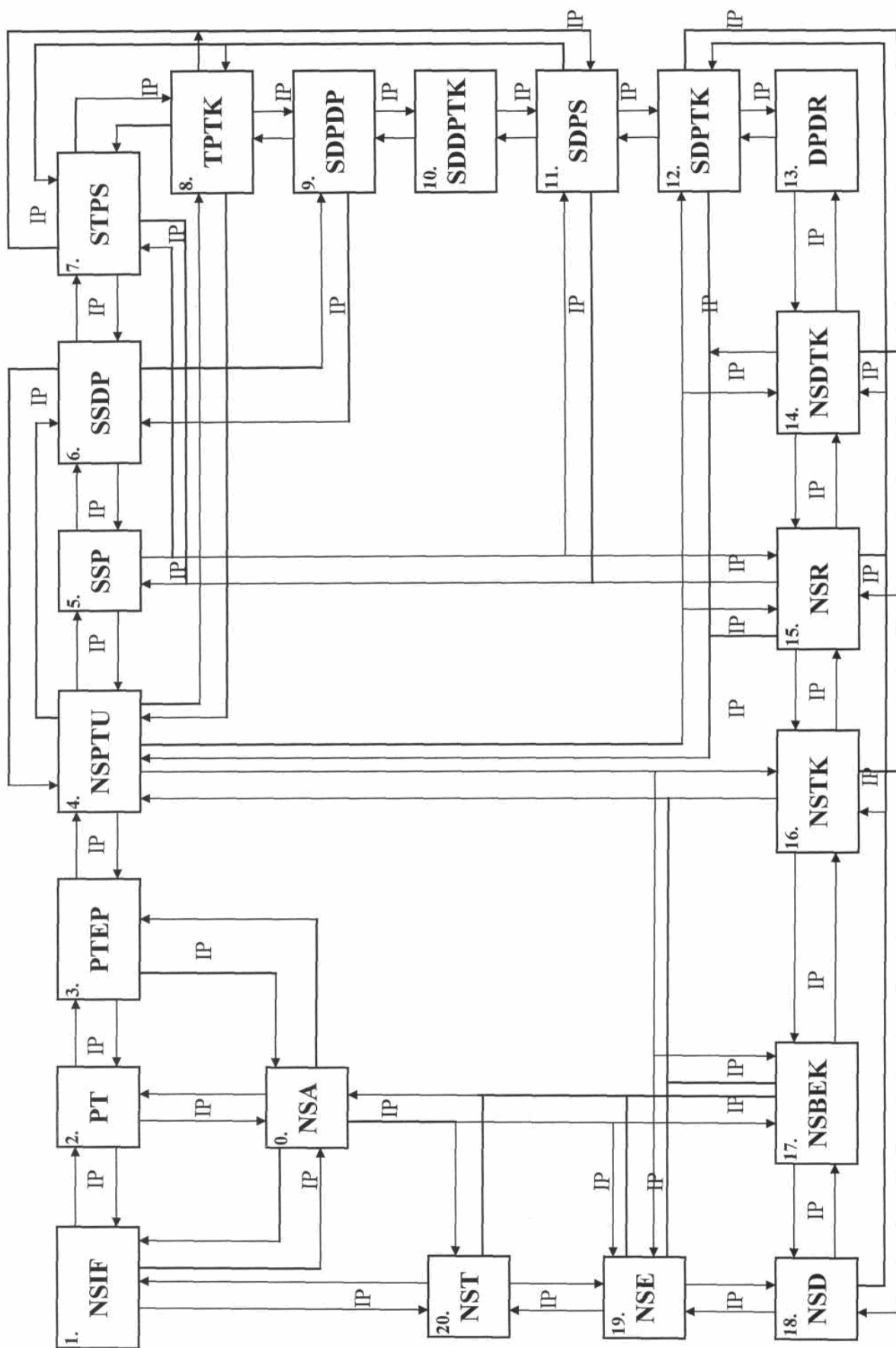
PT – priešprojektinių tyrimų rezultatų blokas; pačius priešprojektinius tyrimus inicijuoja ir realizuoja iteraciniai procesai, jungiantys bloką PT su blokais NSIF ir NSA, o jų rezultatai naudojami naujos sistemos idėjos ir įvaizdžio tikslinimui bei turtinimui, o taip pat iteraciniame procese, jungiančiame PT bloką su PTEP bloku, sprendžiant naujos sistemos sukūrimo idėjos pagrįstumo klausimus;

PTEP – projektavimo techninis bei ekonominis pagrindimas, atsirandantis ieškant atsakymo į klausimus: a) ar įmanoma („techniškai“) pageidaujamą sistemą sukurti, b) ar ekonomiškai ir socialiai apsimoka tokią sistemą kurti; atsakymai į tuos klausimus randami iteraciniuose procesuose, jungiančiuose PTEP ir PT, o taip pat PTEP ir NSA blokus; jei į pirmąjį ar antrąjį klausimą gaunamas neigiamas atsakymas, tai tolimesnis projektavimo procesas nebetešiamas (bent jau iki tol, kol, pasikeitus sąlygoms, bus gautas teigiamas atsakymas); projektavimas tęsiamas tik tuomet, jei gaunamas teigiamas atsakymas į abu minėtus klausimus;

NSPTU – naujos sistemos projektavimo techninė užduotis (TU); ji sudaroma iteracinio proceso, jungiančio šį bloką su PTEP bloku, metu; TU turinį sudaro pakankamai pilnas vienareikšmiškai suformuluotų reikalavimų, kuriuos turi tenkinti projektuojamoji sistema, sąrašas;

SSP – sistemos struktūros projektas; jį sudarant, būtina surasti (ar išrasti) projektuojamos sistemos veikimo idėją (principą) bei to veikimo įvaizdį (konceptinį modelį), jį dekomponuoti į paprastesnius (jau žinomus, ar net anksčiau suprojektuotus) paprocesius, nustatyti tų paprocesių vykdytojus (projektuojamos sistemos posistemius bei elementus), jų medžiaginius, energetinius ir informacinius ryšius, o taip pat ir kitus santykius (pavaldumo, priežasties ir pasekmės ir t.t.) tarp jų; projektuojant sistemos struktūrą, naudojamosi objektinės ir funkcinės dekompozicijos ir sintezės metodais, apie kuriuos plačiau kalbėsime kituose šios metodinės priemonės skyreliuose;

SSDP – sistemos sudėtinių dalių (posistemų ir elementų) projektavimas, jų normatyvinių modelių sudarymas; pats SSDP blokas vaizduoja sistemos sudėties projektą – sistemos sudėtinių dalių normatyvinius modelius, o jų projektavimą – iteraciniai procesai, jungiantys NSPTU ir SSP blokus su SSDP bloku;



6 pav.

STPS – sistemos techninio (principinio, konstrukcinio) projekto sintezė; STPS blokas vaizduoja sistemos techninį projektą, o jo sintezę iš SSDP bloke esančių sudėtinių dalių projektų pagal SSP bloke sudarytą sistemos principinį struktūros projektą – iteraciniai procesai, jungiantys SSDP ir SSP blokus su STPS bloku; sistemos techninis projektas – tai jos konstrukcijos ir veikimo principinė schema, analogiška techninių sistemų (pvz. kompiuterių, televizorių ir pan.) principinėms schemoms;

TPTK – techninio projekto testavimas ir koregavimas, to testavimo ir koregavimo rezultatas – patobulintas techninis projektas, atitinkantis TU reikalavimus;

SDPDP – sistemos darbinio projekto dalių darbiniai projektai, t.y. tų dalių „montažinės schemas“ ir sistemos dalių sukūrimo (pagaminimo, organizavimo) procesų technologiniai projektai;

SDDPTK – sistemos dalių darbinių projektų testavimas ir koregavimas, patobulinti darbinio projekto sudėtinių dalių darbiniai projektai;

SDPS – sistemos darbinio projekto sintezė ir jos rezultatas – sistemos darbinis projektas, t.y. visos suprojektuotos sistemos „montažinė schema“ ir sistemos konstrukcinio projekto realizavimo technologinis projektas;

SDPTK – sistemos darbinio projekto testavimas ir koregavimas, to testavimo ir koregavimo rezultatas – patobulintas sistemos darbinis projektas, tenkinantis TU reikalavimus;

DPDR – darbinio projekto dalių realizavimas, pagamintos (sukurtos) naujos sistemos dalys (jos elementai bei posistemiai);

NSDTK – naujos sistemos dalių testavimas ir koregavimas, patobulintos naujos sistemos dalys, atitinkančios TU bei techninio ir darbinio sistemos projektų reikalavimus;

NSR – naujos sistemos realizavimas, t.y. jos sintezė iš sudarytų (pagamintų) dalių į struktūrą, numatytą SSP ir SDPS blokuose, o taip pat tos sintezės rezultatas – sukurta (pagaminta) sistema ar bandomasis jos egzempliorius (bandomoji serija);

NSTK – naujos sistemos testavimas ir koregavimas, o taip pat šių procesų rezultatas – patobulinta reali nauja sistema, teoriškai atitinkanti TU reikalavimus;

NSBEK – naujos sistemos bandomoji eksploatacija ir koregavimas, o taip pat šių procesų rezultatas – išbandyta ir patobulinta nauja sistema, praktiškai atitinkanti TU reikalavimus;

NSD – naujos sistemos diegimas į „pramoninę eksploataciją“ ir šio proceso rezultatas – paruošta „pramonei eksploatacijai“ nauja sistema; pats diegimas priklausomai nuo sistemos ir jos diegimo proceso sudėtingumo gali vykti ir „eilėmis“, kai diegiami ne iškart visi sistemos posistemiai, o posistemių grupės („eilės“) viena po kitos;

NSE – naujos sistemos eksploatacija;

NST – naujos sistemos tobulinimas eksploatacijos eigoje.

Kaip matyti iš 4 pav. ir 6 pav. blokinių schemų, sisteminė sudėtingų sistemų projektavimo metodologija žymiai skiriasi nuo klasikinės.

Sudėtingos sistemos projektas dėl tokios sistemos neapžvelgiamumo negali būti gautas iškart kaip vieno projektavimo etapo rezultatas, kaip tai būdinga paprastų sistemų projektavimui. Todėl sisteminė sudėtingų sistemų projektavimo metodologija remiasi sudėtingu **projekto kūrimo palaipsniškumo principu**, teigiančiu, jog sudėtingos sistemos kūrimas – **daugiaetapis procesas**, kuriame, pereinant nuo vieno etapo prie kito, palaipsniui išryškėja vis daugiau sistemos projekto detalių ir ryšių tarp jų. Tų detalių ir ryšių pradžioje numatyti nebuvo įmanoma, ir todėl nebuvo tikslinga to iškart siekti. Kiekvienas etapas, pasiekdamas sisteminėje projektavimo metodologijoje numatytus dalinius rezultatus, sudaro sąlygas naujam etapui ir t.t. Tokiu būdu, žingsnis po žingsnio, pradedant nuo pačių bendriausių ir pradžioje net nelabai aiškių euristinių idėjų, jas konkretizuojant, detalizuojant, papildant smulkesnėmis ir konkretesnėmis euristinėmis idėjomis (atradimais bei išradimais), pamažu pereinama prie sistemos techninio projekto, po to – prie jos darbinio projekto ir to projekto realizavimo – naujos sistemos sukūrimo.

Naujos sudėtingos sistemos sukūrimo procese įprasta išskirti šias **stadijas**:

1) priešprojektinę, 2) projektavimo, 3) projekto realizavimo, įdiegimo ir eksploataavimo.

Priešprojektinės stadijos sudėtį sudaro keli jos **etapai**, kuriuos vaizduoja 6 pav. blokai, pažymėti numeriais 1–4. Priešprojektinė stadija baigiama projektavimo techninės užduoties sudarymu.

Projektavimo stadijoje išskiriami trys pagrindiniai etapai: 1) techninio projekto sudarymo, 2) techninio projekto sprendimų patikrinimo, 3) darbinio projekto sudarymo.

Techninio projektavimo etapą sudaro 6 pav. blokai, pažymėti numeriais 5–7. Šis etapas baigiasi sistemos techninio projekto sudarymu.

Techninio projekto sprendimų patikrinimo etapą sudaro 6 pav. blokas, pažymėtas numeriu 8. Techninio projekto sprendimai gali būti patikrinti natūriniais ir matematiniais eksperimentais (imituojant projektuojamos sistemos funkcionavimą), numatytais techninio projekto testavimo programoje. Retkarčiais gali atsitikti taip, kad techniniai sprendimai, priimti sudarant naujos sistemos projektą, jau buvo anksčiau patikrinti ir projektuotojams jokių abejonų bei neiškumų nekelia. Tuomet šis etapas gali būti praleistas, ir, sudarius sistemos techninį projektą, galima iškart pereiti prie darbinio projektavimo.

Darbinio projektavimo etapą sudaro 6 pav. blokai, pažymėti numeriais 9–12. Šis etapas baigiamas sistemos darbinio projekto sudarymu. **Darbinis projektas skiriasi nuo techninio projekto tuo, kad techninis projektas – tai principinė schema, vaizduojanti sistemos struktūrą ir sudėtį bei nusakanti (išreiškianti, aprašanti) projektuojamos sistemos funkcionavimo principus, o darbinis projektas – tai technologinis naujos sistemos sukūrimo (pagaminimo) projektas.** Kitaip tariant, sistemos **techninis projektas – tai normatyvinis išorinis ir vidinis informacinis kuriamos sistemos modelis, o darbinis projektas – normatyvinis sistemos sukūrimo (pagaminimo) proceso modelis**, kartu detalai vaizduojantis tai, kokia turėtų būti kuriamoji nauja sistema.

Vienas pagrindinių techninio projektavimo rezultatų yra kuriamos sistemos **principinė schema**, kuri darbinio projektavimo metu transformuojama į **montažinę schema – tokią principinės schemos formą, kuri ne tik topologiškai, bet ir topografiškai modeliuoja būsimos sistemos sudėtį ir struktūrą**, atsižvelgdama į tos sistemos dalių ir pačios sistemos pagaminimo proceso ypatumus ir reikalavimus, tų dalių erdvinį išdėstymą.

Projektuojant **tipines sudėtingas sistemas**, kurias kuriant galima pasinaudoti jau anksčiau gautais ir patikrintais praktikoje **tipiniais projektavimo sprendimais (sprendiniais)**, techninio ir darbinio projektavimo etapus neretai būna tikslinga sujungti į vieną etapą, tuo sutrumpinant projektavimo stadijos trukmę ir sumažinant projektavimo išlaidas.

Projekto realizavimo, diegimo ir eksploataavimo stadiją (trumpiau vadinamą **poprojektine stadija**) sudaro 6 pav. blokai, pažymėti numeriais 13–20. Šios stadijos sudėtis labai priklauso nuo suprojektuotos sistemos charakterio, nuo to, ar ji – techninė sistema, ar organizacija, nuo sistemos dydžio, jos paskirties bei ilgaamžiškumo, nuo sistemą aptarnaujančio personalo pasiruošimo lygio. Todėl 6 pav. pavaizduoti šios stadijos blokai – tik vienas galimų šios stadijos blokinės schemos variantų, kurių gali būti gana daug ir gana įvairių.

Lygindami sisteminę ir klasikinę metodologijas, galime pastebėti, kad kai kurie jų skirtumai sutampa su jau aprašytais realių sistemų tyrimo ir modeliavimo sisteminės ir klasikinės metodologijų skirtumais (žr. pirmąjį, antrąjį, trečiąjį ir ketvirtąjį skirtumus), nes sudėtingų sistemų projektavimo procesas taip pat yra iteracinis, projektavime dalyvauja įvairių sričių specialistai, projektuotojų kompleksinių kolektyvų vadovavimui taip pat reikalingi dženeralistai, projektavimui, kaip ir deskriptiniam modeliavimui, būdingi vienas kitą papildantys ir vienas kitą koreguojantys dekompozicijos ir sintezės procesai, projektuojant sudėtingą sistemą, būtina atsižvelgti į jos būsimą sąveikavimą su aplinka ir jo pasekmes.

Sisteminė metodologija žymiai sudėtingesnė už klasikinę, todėl jos įsisavinimas ir išmokymas ją taikyti praktiniame darbe, tiriant ir kuriant sudėtingas sistemas, reikalauja nelengvo ir ilgo darbo. Šiame darbe gali padėti jau minėtų sisteminio mąstymo bei sisteminės metodologijos ir jos taikymo principų, o taip pat iš jų išplaukiančių išvadų įsisavinimas

SIS kūrėjams naudingos informacijos galėtų suteikti šie bendrasisteminiai principai:

1. **Elementų suderinamumo sistemoje principas**, teigiantis, jog **objektų bei subjektų, iš kurių norima sudaryti mums reikalingą sistemą, savybės, santykiai ir tikslai turi būti tokie, kad juos būtų įmanoma suderinti tarpusavyje, o taip pat ir su kuriamos sistemos misija bei jos tikslais**, nes iš esybių, netenkinančių šio principo reikalavimų, negali būti sudaryta (susidaryti) mums reikalinga efektyviai veikianti sistema.

Jau minėtose šios metodinės priemonės autoriaus knygutėse elementų suderinamumo principo aprašyme pateikiami penki derinimo būdai, kuriuos jau minėtas M.Mesarovičius pavadino **koordinavimo principais**: 1) interfeisų prognozavimo (planavimo), 2) interfeisų laisvo pasirinkimo, 3) interfeisų leistino kitimo ribų įvertinimo, 4) atsakomybės suteikimo, 5) koalicijų organizavimo. Su šiais koordinavimo principais ypatingai naudinga susipažinti stambių organizacijų (valstybių ir jų regionų, ministerijų, stambių įmonių ir t.t.) aukšto lygio vadovams. Jie aprašyti M.Mesarovičiaus, D.Tako ir J.Takahara knygoje „Hierarchinių daugialygiinių sistemų teorija“ (yra vertimas iš anglų kalbos į rusų kalbą), o taip pat šių eilučių autoriaus knygutėje, parašytoje lietuvių kalbą (žiūr. šios metodinės priemonės 17–jo puslapio išnašą, kurioje ta knygutė minima).

2. **Organizavimosi ir organizuotumo principas**, teigiantis, jog pirmojo tipo sistemos atsiranda ir efektyviai funkcionuoja tik tada, kai jų elementai, tenkinantys suderinamumo principo reikalavimus, **jungiasi ne bet kaip, o tam tikra tvarka į tam tikrą struktūrą ir pradeda santykiauti tarpusavyje ne bet kaip, o tenkindami tam tikrus sisteminius dėsningumus** (laikydami būdingą tai sistemai taisyklių, organizuojančių sistemos elementų veiklą ir ribojančių jų veiksmų bei poelgių laisvę). Tik tenkinant šio principo reikalavimus, gali atsirasti tos sisteminės savybės, dėl kurių buvo kuriama ar kūrėsi mus dominanti pirmojo tipo sistema.

Pirmojo tipo sistema gali susiformuoti ir „stichiškai“, evoliucijos keliu, veikiant mutacijų (atsitiktinių pokyčių), atrankos ir paveldimumo faktoriams, tačiau tam reikia ilgo laiko ir palankių aplinkybių. Daug greičiau ir su daug didesne tikimybe mes galime gauti norimą rezultatą – sistemą su mums reikalingomis savybėmis, jei mes sąmoningai parinksime tinkamus (tenkinančius pirmojo principo reikalavimus) elementus ir sąmoningai sujungsime (suorganizuosime) juos į atitinkamą struktūrą. Net ir tuo atveju, kai sistema vystosi evoliucijos keliu, mes galime paspartinti evoliucijos procesus bei nukreipti juos mums labiau pageidautina linkme, imdamiesi sąmoningo organizacinio darbo, iš anksto suprojektavę mums naudingų sistemos tobulinimo procesų eigą. Taip, pavyzdžiui daroma augalų bei gyvūnų selekcijoje. Žinoma, prieš imantis sąmoningo tobulinimo, reikia tuos procesus pamodeliuoti, paprognozuoti ir gerai pasverti mūsų sąmoningų poveikių pasekmes, kad nepadarytume daugiau žalos negu naudos. Tai kartais pasitaiko, kai „tobulinimų“ (ir, bendrai „reformų“ ar net „revoliucijų“) imasi ryžtingi, tačiau netoliaregiški „reformatoriai“ ar „revoliucionieriai“. Todėl **tobulinant sistemas bei vykdant reformas reikia ne tik proto, bet ir išminties**. Išmintingas žmogus skiriasi nuo protingo, bet neišmintingo tuo, kad sugeba surasti ne tik racionalius sudėtingų problemų sprendimus, ką sugeba ir pastarasis, bet ir numatyti tolimas priimamų sprendimų pasekmes, ko protingas, bet neišmintingas žmogus dažnai nesugeba.

3. **Emerdžentiškumo principas**, teigiantis, jog **objektams ir subjektams jungiantis į sistemą ir atsirandant naujiems ryšiams, gali atsirasti ir naujų savybių, priklausančių visai sistemai, o ne tik atskiriems jos elementams, ir kad dekomponuojant sistemą, kai nutraukiami kai kurie vidiniai jos ryšiai, kai kurios sisteminės savybės gali išnykti**. Šito principo nežinojimas bei jo nepaisymas – viena iš svarbiausių priežasčių tų klaidų, kurias neretai daro sudėtingų sistemų projektuotojai bei organizacijų vadovai. Emerdžencijos reiškinytis yra viena svarbiausių priežasčių, dėl kurių sunku numatyti sudėtingose situacijose

priimamų sprendimų galimas tolimas pasekmes. Tačiau emerđžencijos reiškinys labai naudingas tuo, kad sistemų jungimo metu galinčios atsirasti naujos savybės kaip tik ir sudaro potencialų pagrindą naujų objektų su naujomis savybėmis konstravimui. Be to, kaip tik dėl emerđžencijos reiškinio, sujungiant darbuotojų pastangas ir pajėgas bendro darbo metu, gali atsirasti papildomas efektas, vadinamas **sinergijos efektu**, dėl kurio darbo rezultatas gali tapti gerokai didesnis negu tų rezultatų, kuriuos būtų galima pasiekti pavienėmis pastangomis, suma. Vargu ar įmanoma pervertinti emerđžencijos reiškinio vaidmenį pasaulio evoliucijoje: jei nebūtų emerđžencijos reiškinio, vargu ar būtų galėjusi mūsų planetoje atsirasti gyvybė, sąmonė ir kiti sudėtingi reiškiniai.

4. Sistemos savybių priklausomybės nuo jos struktūros principas, išplaukiantis iš emerđžentiškumo principo ir teigiantis, jog **sistemos savybės priklauso ne tik nuo jos sudėties, nuo jos elementų individualių savybių, bet ir nuo ryšių tarp tų elementų, jų vietos sistemoje, t.y. nuo sistemos struktūros.**

Tokios priklausomybės pavyzdžių apstu ne tik techninėse sistemose ir organizacijose, kurias žmonės specialiai kuria, sąmoningai jungdami elementus į struktūras taip, kad būtų gaunamos reikiamos techninių sistemų bei organizacijų sisteminės savybės, bet ir gamtoje. Pavyzdžiui, vanduo, būdamas įvairiose fazėse (garų, skysto kūno ir kieto kūno, vadinamo ledu) turi skirtingas struktūras ir skirtingas savybes, nors cheminė vandens sudėtis lieka ta pati, o toks cheminis elementas kaip anglis, net būdamas kieto kūno fazėje, bet turėdamas skirtingas kristalines struktūras, gali būti ir gana minkštu grafitu, ir labai kietu deimantu.

Iš šio principo išplaukia, kad mes, **keisdami sistemos struktūrą, net nekeisdami jos elementų, galime keisti sistemos sisteminės ir jos elementų struktūrinės savybes**, kad blogai organizuotoje organizacijoje net geri darbuotojai gali dirbti nepakankamai efektyviai, o gerai organizuotoje organizacijoje net nelabai geri darbuotojai tampa priversti arba normaliai dirbti, arba palikti tą organizaciją.

5. Sistemos elementų funkcijų aktualizavimo principas, teigiantis, jog elementams jungiantis į sistemą, jų funkcijų sąrašas ir jų veikla gali pasipildyti naujomis funkcijomis, aktualiomis (reikalingomis) visos sistemos funkcionavimui, ir kad **elementų funkcijų aktualizavimo laipsnis gali charakterizuoti sistemos organizuotumo laipsnį, jos struktūros atitikimo sistemos paskirčiai laipsnį.** Gerai organizuotoje sistemoje jos elementai neturi būti apkraunami funkcijomis, neišplaukiančiomis iš sistemos paskirties: elementų funkcijos turi būti aktualios sistemai, o sistemos funkcijos turėtų būti aktualios jos elementams ir tos sistemos vartotojams.

6. Funkcijų labilizavimo principas, teigiantis, jog **besivystant sistemoms ir didėjant jų sudėtingumui, jų struktūra ir sistemose vykstantys procesai gali įgyti labilumo savybę, t.y. savybę greitai keisti vykdomas funkcijas bei pradėti vykdyti naujas funkcijas, nekeičiant arba beveik nekeičiant sistemos sudėties ir struktūros.** Sistemos labilumą galima matuoti sistemos funkcijų labilumo ir jos sudėties bei struktūros stabilumo, keičiant funkcijas, santykiu. Pavyzdžiui, norint pertvarkyti upėmis beplaukiojantį katerį į kombainą javų derliui nuimti, reikia jį kardinaliai perkonstruoti, iš esmės keičiant sudėtį ir struktūrą, o perkvalifikuoti upeivį, vairuojantį tą katerį, į kombainininką galima, visiškai nekeičiant tokios sistemos kaip žmogus anatinės sudėties ir struktūros, pakeičiant tik sudėtį ir struktūrą kai kurių veiklos programų, užrašytų ir saugomų žmogaus atmintyje, papildant tų programų banką naujomis programomis. Tai rodo, jog tokia sudėtinga sistema, kaip žmogus, yra žymiai labilesnė už katerį. **Labilumo savybė gali atsirasti tik tokiose sistemose, kurios turi atmintį ir sugeba saugoti joje sistemos veikimą valdančią informaciją – veiklos programą. Keisdami tą programą, mes galime keisti sistemos veikimo algoritmą, nekeisdami arba beveik nekeisdami sistemos sudėties ir struktūros.**

Naudodamiesi sistemos elementų funkcijų aktualizavimo ir sistemos funkcijų labilumo tyrimų rezultatais, mes galime įvertinti **sistemų organizuotumo lygį ir laipsnį.** Pavyzdžiui, lygindami skruzdėlyną ir žmonių kolektyvą (sakykim, Statistikos departamento kolektyvą) organizuotumą, mes rasime, jog skruzdėlyną organizuotumo laipsnis yra aukštesnis, nes

skruzdėlių vykdomos funkcijos labiau aktualizuotos, tačiau jo organizuotumo lygis yra žymiai žemesnis, nes tokios sistemos, kaip skruzdėlynas, labilumas, lyginant jį su žmonių kolektyvų labilumu, yra labai žemas.

Iš funkcijų labilumo principo išplaukia, jog **projektuojant bei kuriant (organizuojant) sudėtingas sistemas, reikia stengtis ne tik aktualizuoti sistemų elementų funkcijas, bet ir, siekti, kad sistema būtų pakankamai labili.** Pavyzdžiui, kompleksinės „žmogus + technika“ tipo sistemos yra žymiai labilesnės, negu grynai techninės sistemos. Į tai reikia atsižvelgti, kuriant automatizuotas sistemas: ne viską, ką galime automatizuoti, yra tikslinga automatizuoti, nes kartais automatizavimas gali nepasiteisinti ekonomiškai, o kartais jis gali būti netikslingas labilumo išsaugojimo sumetimais. Tačiau reikia turėti omenyje, kad kompiuteris dialogo su žmogumi režime ne tik nesumažina žmogaus labilumo, bet gali ir žymiai padidinti jį.

7. **Disfunkcijų neutralizavimo principas**, teigiantis, jog **sistemose gali atsirasti ir egzistuoti ne tik aktualios funkcijos, bet ir disfunkcijos** – įvairus nesuderinamumai bei sutrikimai sistemos ir jos dalių veikloje, todėl **sistemoje turi būti sudaryti ir joje egzistuoti tų disfunkcijų neutralizavimo „mechanizmai“.** Tokių „mechanizmų“ sudarymas ir jų veiklos palaikymas viena iš aktualių sistemos vidinių funkcijų. Tokie „mechanizmai“ turi būti projektuojami jau sistemos projektavimo metu, nes be jų sistema negalės efektyviai funkcionuoti. Tačiau dėl sistemų sudėtingumo viską iš anksto numatyti yra sunku, todėl gali atsirasti reikiamybė kurti tokius „mechanizmus“ papildomai, jau sistemos funkcionavimo metu. O tai reiškia, jog jau sistemos kūrimo metu reikėtų atsižvelgti į tokia reikiamybę ir numatyti sistemos projekte galimybes tokių papildomų „mechanizmų“ sudarymui.

8. **Sudėtingų techninių sistemų bei organizacijų polifunkcionalumo principas**, teigiantis, jog dauguma sudėtingų organizacijų, o neretai – ir sudėtingos techninės sistemos yra polifunkcionalios, t.y. galinčios turėti ne vieną, o kelias paskirtis, vykdyti ne vieną, o kelias bendrasistemines funkcijas, iš ko išplaukia, kad tokių sistemų veiklos nagrinėjimui bei tobulinimui, o taip pat tokių sistemų projektavimui reikalingi **kompleksiniai sisteminių analitikų bei projektuotojų kolektyvai**, sudaryti iš įvairių sričių specialistų, vadovaujamų dženeralistų.

9. **Funkcijų specializavimo (diferencijavimo) bei jų jungimo (integravimo) ir šių procesų sintezės principas**, teigiantis, jog didėjant sistemų organizuotumui vyksta du priešingos krypties, tačiau vienas kitą papildantys procesai – **funkcijų diferencijavimosi ir funkcijų integravimosi procesai, didinantys sistemos funkcionavimo efektyvumą.** Funkcijų diferenciacija pasireiškia tuo, kad kai kurios sudėtingos funkcijos suskyla į paprastesnes, o sistemos elementai ir jų grupės specializuojasi tam tikrų funkcijų ir jų grupių vykdyme, didindami savo veiklos efektyvumą bei meistriškumą. Funkcijų integracija pasireiškia tuo, kad kai kurios giminingos funkcijos jungiasi, integruojasi, atsiranda naujos integralinės funkcijos (pavyzdžiui, elementų ir jų grupių veiklos koordinavimo funkcijos), ir viso to rezultate didėja sistemoje vykstančios veiklos organizuotumas ir efektyvumas. Sistemų projektuotojams ir organizatoriams reikia išmokti vertinti ir skirti, kada ir kur funkcijų diferenciacija ar integracija yra naudingos, nes didina sistemos veiklos efektyvumą, o taip pat kada ir kur ta diferenciacija ar integracija galėtų būti žalingos. Toks vertinimas padėtų racionaliai spręsti sistemos **struktūros ir veiklos centralizavimo ir decentralizavimo santykio pasirinkimo** klausimus, kurie būtinai išskyla, projektuojant ir tobulinant sistemų struktūras. Sistemotyros rėmuose susiformavo net specialus **grupinės dinamikos** mokslas, nagrinėjantis funkcijų centralizavimo ir decentralizavimo procesus bei jų santykį, racionalizuojant sistemų struktūras. Funkcijų decentralizavimo ir jos centralizavimo klausimai turėtų būti gana aktualūs ir SIS projektuotojams, nes šiuo metu funkcijų paskirstymas departamente nėra pakankamai racionalus. Decentralizuotos struktūros sugeba užtikrinti demokratiškumą ir visapusiškumą, sumažina galimybę priimti vienašališkus sprendimus, todėl tokios struktūros rekomenduotinos svarbius ir ilgalaikius sprendimus priimančioms sistemoms. Centralizuotos struktūros rekomenduotinos jau priimtų sprendimų vykdymo

valdymui, nes jos veikia operatyviau ir vieningiau. Todėl **įstatymus leidžiančiai valdžiai turėtų būti būdinga decentralizacija, o vykdomajai valdžiai – centralizacija.**

10. **Izomorfizmo principas**, teigiantis, jog **sistemos, turinčios izomorfiškas (paprastiau kalbant, atitinkamai vienodas ar labai panašias) struktūras, būdamos sudarytos net iš skirtingų elementų, gali turėti bendrų savybių, jose vykstantiems procesams gali būti būdingi analogiški ar net vienodi dėsningumai, ir todėl tokios sistemos gali būti viena kitos modeliais.** Ištyrę vieną iš izomorfiškų sistemų, mes galėtume daug ką pasakyti apie visų kitų jai izomorfiškų sistemų funkcionavimo ypatumus. Iš izomorfizmo principo kaip tik ir išplaukia sistemų modeliavimo ir eksperimentų su modeliais tikslingumas, o taip pat ir tokio mokslo, kaip sistemotyra egzistavimo galimybė bei prasmingumas. Izomorfizmas būdingas ne tik sistemoms – konstrukcijoms, bet ir sistemoms – procesams. Pavyzdžiui, organizacijų projektavimo procesas daugeliu atžvilgių yra gana izomorfiškas sudėtingų techninių sistemų projektavimo procesui, todėl nemažai reiškinių ir savybių abiejuose procesuose arba sutampa, arba analogiški. **Kadangi sudėtingų techninių sistemų projektuotojai yra sukaupę gana turtingą patirtį ir sukūrę nemažai efektyvių metodų, tai organizacijų projektuotojams, turintiems kur kas mažesnę patirtį, būtų naudinga pasimokyti iš techninių sistemų projektuotojų.** Iš izomorfizmo principo išplaukia, jog ir organizacijas galima ir tikslinga projektuoti bei kurti, naudojantis inžineriniais metodais, kaip projektuojamos ir kuriamos sudėtingos techninės sistemos.

11. **Adaptyvumo principas**, teigiantis, jog **sudėtinga sistema, orientuota į tam tikrų tikslų pasiekimą ir egzistuojanti besikeičiančioje aplinkoje, gali ir privalo turėti adaptyvumo savybę, t.y. sugebėjimą pertvarkyti savo struktūrą, savo parametrus ir savo elgesį, prisitaikant prie aplinkos sąlygų.** Adaptyvumo savybės tikslingumas ir net būtinumas išplaukia iš aplinkos kintamumo, o adaptyvumo galimybė – iš jau minėtojo sudėtingų sistemų labilumo, o taip pat ir iš emerđžencijos reiškinių egzistavimo.

Adaptacija gali pasireikšti gana įvairiai: kartais gali keistis tik sistemos funkcionavimas (**funkcinė arba elgesio adaptacija**), tačiau gali pasikeisti ir sistemos parametrai (**parametrinė adaptacija**), o kartais – ir sistemos struktūra ar net sudėtis (**struktūrinė adaptacija**). Gana dažnai **adaptacija būna mišri**, kai, pavyzdžiui, keičiasi ir sistemos elgesys ir jos parametrai bei jos struktūra.

Sistemos, sugebančios vertinti savo poelgius bei jų pasekmes ir turinčios atmintį, gali kaupti savo patirtį. Todėl jos gali ne tik adaptuotis, bet ir tobulinti adaptacijos procesus: gali didinti prisitaikymo greitį, gerinti prisitaikymo kokybę. Tokį reiškinį vadina sistemos **mokymusi**. Yra sistemų, kurios gali mokytis tik būdamos mokomos (tokioms sistemoms reikalingi mokytojai), tačiau yra ir tokių, kurios gali mokytis pačios, būdamos savęs pačių mokytojais. Žinoma, yra ir tokių sistemų, kurios mokymosi savybės neturi, tačiau visos organizacijos ir jų nariai gali ir privalo mokytis, tobulinti save. Tai tampa ypač aktualu informacinėje visuomenėje, kuri pradėjo formuotis XX a. pabaigoje ir intensyviai vystosi dabar.

Šiuo metu Lietuva, jos žmonės ir įvairios jose esančios valstybinės ir privačios organizacijos pergyvena didelių pasikeitimų metą, todėl visiems gana svarbu pakankamai greitai ir pakankamai lengvai adaptuotis, prisitaikant prie naujų sąlygų visais tais atvejais, kai nepavyksta ar neįmanoma tas sąlygas pakeisti ir pritaikyti prie savo poreikių (ką irgi būtų galima pavadinti adaptacija, kuri skiriasi nuo paprastos adaptacijos tuo, kad sistema, pavyzdžiui, **žmogus prisitaiko prie aplinkos, keisdamas ne tik save, bet ir pačią aplinką**). Adaptuojantis svarbu racionaliai naudotis kitų sukaupta patirtimi ir efektyviai mokytis ne tik iš savo klaidų bei pasiekimų, bet ir iš to, ką patyrė ir pasiekė kiti.

Adaptacijos principas ir iš jo išplaukiančios išvados turėtų dominti ir Statistikos departamento darbuotojus ir, tuo labiau, SIS projektuotojus, kuriančius tokią sistemą, kuri turės dirbti nuolat besikeičiančioje aplinkoje. Būtų blogai, jei būtų suprojektuota sistema, turinti dabartiniu metu optimalią struktūrą ir esamomis sąlygomis funkcionuojanti optimaliai, tačiau nelanksti, nelabili, nesugebanti adaptuotis. Pasikeitus sąlygoms, tokia sistema galėtų

tapti ne tik neoptimali, bet ir visai neracionali. Kuriant organizacinę sistemą, reikia specialiai pasirūpinti, kad jos aktyvieji elementai turėtų galimybę efektyviai mokytis ir tobulinti save ir savo aplinką, efektyviai adaptuotis ir adaptuoti savo sistemą, o esant galimybei – ir aktyviai veikti sistemos aplinką, keičiant ją pageidautina linkme.

Adaptuotis gali ne tik individai bei atskiros sistemos, bet ir iš individų sudarytos populiacijos bei jų rūšys, nes, veikiant aplinkai, gali keistis individų genai, ir tie pasikeitimai gali persiduoti iš kartos į kartą. Taip atsiranda evoliucijos procesai gyvojoje gamtoje.

12. Evoliucijos principas, teigiantis, jog **sistemos dėl išorinių ir vidinių poveikių gali keistis ir, susiklosčius tam tikroms sąlygoms, gali evoliucionuoti**. Šis principas reikalauja, kad organizacijų vadovai, sprenddami organizacijose išskylančius einamuosius klausimus, nepamirštų evoliucijos reiškinių egzistavimo ir nuolat stengtųsi, kad jų vadovaujamos organizacijos neatsiliktų nuo evoliucionuojančios aplinkos, kad evoliucijos procesai jų organizacijose vyktų reikiama linkme ir reikiamais tempais. Iš to išplaukia, jog organizacijų vadovai turi suprasti evoliucijos „mechanizmus“ ir mokėti jais naudotis.

Evoliuciją galima pastebėti ir gyvojoje gamtoje, ir visuomeninėje gyvenime, ir mokslo bei technikos vystymesi.

Gyvojoje gamtoje, veikiant išoriniams bei vidiniams poveikiams, individų genuose gali įvykti įvairūs pakitimai (mutacijos), dėl kurių individų savybės, pereinant iš kartos į kartą, gali įvairiai keistis. Vienos individų grupės, būdamos geriau prisitaikiusios prie aplinkos, gali lengviau išgyventi ir, palikdamos vis daugiau palikuonių, gali didėti, o kitos, būdamos blogiau prisitaikiusios, gali ir visai išnykti. Tokiu būdu, veikiant **mutacijų, atrankos ir paveldimumo** „mechanizms“, atsiranda įvairios populiacijų rūšys, prisitaikiusios prie savo užimamų „nišų“ gamtos pasaulyje; taip tose rūšyse stiprėja rūšies išlikimui ir augimui naudingos savybės ir silpnėja tai, kas žalinga; taip vyksta kai kurių populiacijų bei jų rūšių išsigimimo bei išnykimo procesai, kadangi mutacijos gali būti ne tik naudingos, bet ir žalingos, netenkinančios atrankos kriterijų.

Mutacijų, atrankos ir paveldimumo „mechanizmai“ veikia ne tik gyvojoje gamtoje, bet ir visuomeninėje gyvenime (ekonomikoje, politikoje ir t.t.), tik čia mutacijos vyksta ne individų genuose, bet visuomenėje vyraujančiose idėjose ir organizacijų struktūrose bei individų santykiuose, visuomenės kultūroje ir moralėje, o atranką realizuoja ekonominė, politinė ir t.t. konkurencija.

Analogiškus procesus ir „mechanizmus“ galime pastebėti ir mokslo bei technikos vystymesi, kur taip pat vyksta patirties atranka ir kaupimas, naujų idėjų (atradinų bei išradimų) atsiradimas, jų tikrinimas, sukaupytų žinių perdavimas naujoms kartoms.

Nesunku pastebėti, kad sudėtingų sistemų modeliavime ir projektavime plačiai taikomi eksperimentų su matematiniais modeliais metodai gana dažnai naudojami evoliucijos „mechanizmais“ bei algoritmais: daromi atsitiktiniai bandymai („mutacijos“), vyksta jų rezultatų kaupimas ir atranka, sukaupta informacija perduodama į sekantį bandymą, modeliavimo ir projektavimo etapą, kur vėl daromi, tačiau jau nauji bandymai ir t.t. Tokiu būdu palaipsniui artėjama prie siekiamo tikslo. Tame artėjime galimi ir gana tolygūs kokybiniai poslinkiai, ir staigūs kokybiniai šuoliai, kai emerđencijos reiškinių dėka gimsta naujos idėjos ir savybės.

13. Sudėtingumo (pirmojo informacinio barjero) principas, teigiantis, jog **egzistuoja bei, jungiant mažas sistemas į didesnes, gali būti sudarytos tokios sistemos, kuriose vykstančius procesus bei jų „mechanizmus“ ir dėsningumus apžvelgti ir suvokti vienas žmogus tampa jau nebepajėgus**. Tokias sistemas vadiname sudėtingomis, o tą jų sudėtingumo lygį, kurio vienas žmogus jau nebepajėgia įveikti – **pirmuoju informaciniu barjeru**. Dėl šio barjero vienas žmogus jau nebegali racionaliai valdyti sudėtingą sistemą bei joje vykstančius procesus.

Norint įveikti pirmąjį informacinį barjerą, sudėtingą sistemą tenka dekomponuoti į mažiau sudėtingas dalis, pavedant jų modeliavimą, projektavimą bei valdymą atskiriems žmonėms, o po to bendromis pastangomis, vadovaujant dženeralistams, įvykdyti atvirkštinį –

integracinį procesą. **Taip atsiranda sisteminė modeliavimo ir projektavimo metodologija, taip susiformuoja hierarchinės sudėtingų sistemų valdymo sistemos**, kuriose paprastoms sistemoms vadovauja žemutinio lygio vadovai, šių vadovų veiklą koordinuoja aukštesnio lygio vadovai (kurių kiekvienam priskiriama po keletą žemesnio lygio vadovų), aukštesnio lygio vadovų veiklą koordinuoja dar aukštesnio lygio vadovai ir t.t.

14. **Iteratyvumo principas**, teigiantis, jog **sudėtingų sistemų modeliavimo ir projektavimo bei jų valdymo sprendimų priėmimo procesai dėl tokių sistemų neapžvelgiamumo beveik neišvengiamai tampa iteratyviais**, t.y. besikartojančiais, artėjant prie siekiamo rezultato, nes jį pasiekti iškart beveik niekad nepavyksta. Tai žymiai prailgina ir apsunkina modeliavimo, projektavimo ir valdymo sprendimų priėmimo procesus, tačiau su tuo tenka taikstyti, norint pasiekti reikiamos modelių, projektų bei valdymo sprendimų kokybės. **Taip iteratyvumas tampa vienu iš specifinių sisteminės metodologijos bruožų**. Tai būdinga ir jos dėstymo bei įsisavinimo procesui: sudėtingą metodologiją be iteracijų aprašyti ir įsisavinti yra sunku, jei išvis tai įmanoma. Tik kelis kartus grįžtant prie to paties, pavyksta gerai išaiškinti ir išsiaiškinti sudėtingus klausimus.

15. **Antrojo informacinio barjero principas**, teigiantis, jog **egzistuoja toks sudėtingos sistemos valdymo sistemos sudėtingumo lygis bei sudėtingos problemos sprendimo sistemos sudėtingumo lygis, vadinamas antruoju informaciniu barjeru, nuo kurio tolimesnis valdymo personalo skaičiaus (bei problemą sprendžiančio kolektyvo narių skaičiaus) n didinimas ne tik nepalengvina valdymo bei problemos sprendimo sistemos darbo, bet net dar apsunkina jį**.

Antrasis informacinis barjeras atsiranda todėl, kad galimų abipusių ryšių tarp valdymo sistemos elementų (problemos sprendėjų) skaičius S_n , būdamas lygus $\frac{n(n-1)}{2} \approx \frac{n^2}{2}$, auga didėjant elementų skaičiui n , galima sakyti, „paraboliškai“, t.y. apytikriai kvadratu, kai tuo tarpu valdymo sistemos pajėgumas P_n geriausiu atveju gali augti proporcingai elementų skaičiui, t.y. „tiesiškai“: $P_n = kn$, kur $k = \text{const}$ – vieno elemento vidutinis pajėgumas (darbo našumo koeficientas).

Priimdami prielaidą, jog valdymo bei problemos sprendimo sudėtingumas S , iš vienos pusės, turėtų būti proporcingas galimų abipusių ryšių tarp valdymo sistemos elementų skaičiui S_n , o iš kitos pusės – valdymo problemų sprendimui reikalingo darbo apimčiai, mes galime valdymo sudėtingumą S ir tą darbą D , kurį gali padaryti valdymo personalas per laiko vienetą, matuoti vienodais darbo apimties vienetais ir grafiškai pavaizduoti **valdymo sudėtingumo**

$S = cS_n = c \frac{n(n-1)}{2}$, kur $c = \text{const}$ – proporcingumo koeficientas, ir **sistemos pajėgumo**

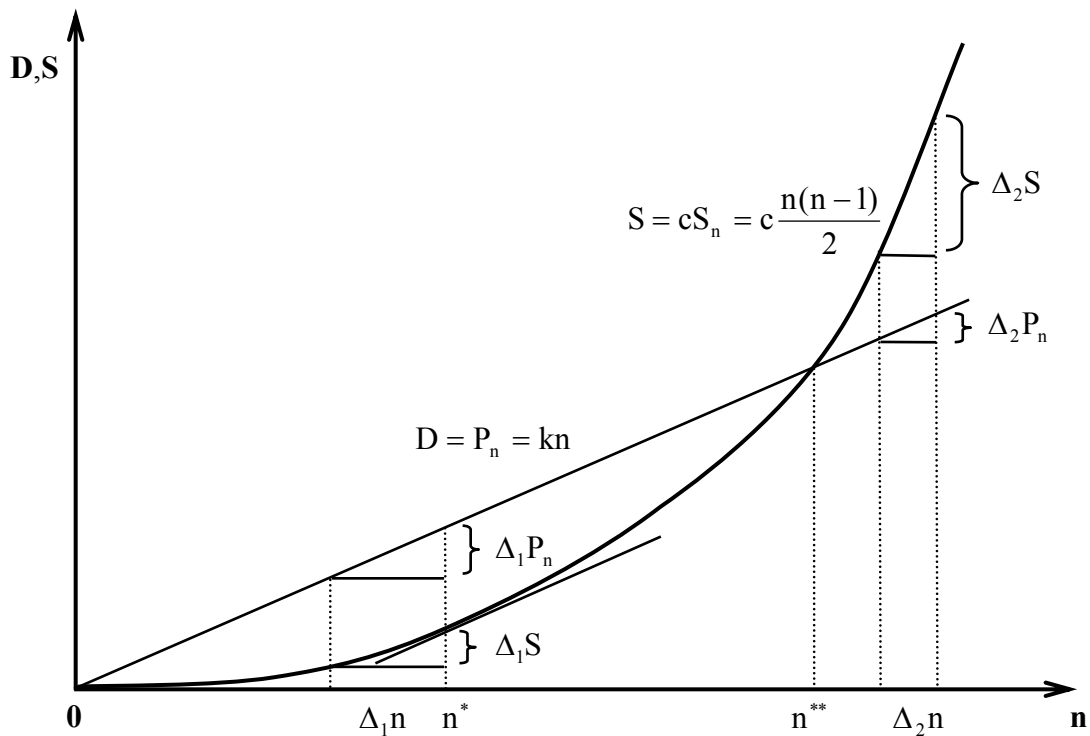
$P_n = kn$ priklausomybę nuo valdymo personalo skaičiaus n viename brėžinyje, kaip parodyta 7 pav.

Pažymėję n^* tokią n reikšmę, kuriai esant kreivės S liestinė tampa lygiagreti tiesei $P_n = kn$, galime pastebėti, jog **intervale $0 < n < n^*$, didėjant valdymo personalo skaičiui n , valdymo sistemos pajėgumas P_n auga greičiau, negu valdymo personalo bendravimo potencialus sudėtingumas S** , t.y., $\frac{dP_n}{dn} > \frac{dS}{dn}$, o **perėjus tašką $n = n^*$, kuriame $\frac{dP_n}{dn} = \frac{dS}{dn}$,**

intervale $n > n^*$ sudėtingumo S augimo greitis $\frac{dS}{dn}$ vis labiau lenkia pajėgumo P_n augimo

greitį (žiūr. 7 pav., kur matome, jog intervale $n \in [0; n^*]$ galioja nelygybė $\frac{\Delta_1 P_n}{\Delta_1 n} > \frac{\Delta_1 S}{\Delta_1 n}$, nes

$\Delta_1 P_n > \Delta_1 S$, o intervale $n > n^*$ – nelygybė $\frac{\Delta_2 P_n}{\Delta_2 n} < \frac{\Delta_2 S}{\Delta_2 n}$, nes $\Delta_2 P_n < \Delta_2 S$).



7 pav.

Visa tai rodo, jog **didinti valdymo personalo skaičių n efektyvu tik iki dydžio $n^* = \frac{k}{c} + \frac{1}{2}$** , nes tolimesnis personalo skaičiaus n didinimas didina valdymo personalo nuomonių ir veiksmų derinimo sudėtingumą, sprendžiant sudėtingas valdymo problemas, sparčiau, negu auga valdymo sistemos bendras pajėgumas.

Darbuotojų skaičių n padidinus iki dydžio $n^{**} = \frac{2k}{c} + 1$, nustatomo iš lygybės $P_{n^{**}} = S$, atitinkančios tiesės $P_n = kn$ ir kreivės $S = c \frac{n(n-1)}{2}$ susikirtimo tašką, visas problemas sprendžiančios sistemos R darbuotojų kolektyvo pajėgumas $P_{n^{**}}$ būtų išeikvotas darbuotojų tarpusavio bendravimui ir jo nebeliktų jokiame kitame darbe. Tolimesnis darbuotojų skaičiaus n didinimas taptų ne tik žalingas, bet ir pavojingas, nes pajėgumo P_n jau nebepakaktų net darbuotojų tarpusavio bendravimo pakankamumo užtikrinimui. Tokia sistema taptų nebegyvinga ir subyrėtų, kaip, pavyzdžiui, subyrėjo visos didžiosios imperijos, tapusios dėl savo dydžio neefektyvios ir nebevaldomos.

Antrojo informacinio barjero nagrinėjimas patvirtina garsųjį „Piterio dėsnį“: „Jei tūkstantis klerkų nepajėgia susidoroti su savo darbu ir todėl organizacijoje nėra reikiamos tvarkos, tai, pasiuntę jiems į pagalbą dar vieną klerką, situaciją tik pabloginsime: darbo ir nesusipratimų tik padaugės, o tvarkos – sumažės“.

Reikia atkreipti skaitytojo dėmesį į tai, kad skaičius n^* priklauso nuo vidutinio individualaus pajėgumo k, kolektyvo organizuotumo bei nuo kolektyviai sprendžiamos problemos charakterio ir gana dažnai yra visai nedidelis. Pavyzdžiui, kolektyviai ruošiant kokio nors sprendimo ir jį išreiškiančio dokumento projektą, netikslinga burti dideles grupes. Pilnai pakanka 3÷11 specialistų ir, tik paruošus pirminį projekto variantą, tikslinga jį apsvarstyti platesniuose kolektyvuose, surenkant pastabas bei pasiūlymus, po ko pirminį variantą tikslinga pakoreguoti toje pačioje ar, galbūt, dar mažesnėje grupėje ir pateikti viso kolektyvo galutiniam svarstymui ir priėmimui. Pradėti kokio nors projekto kūrimą nuo svarstymo dideliame kolektyve tikslinga tik tada, kada projektavimo vadovams trūksta idėjų ir

pradinės informacijos, reikalingos pradinės projekto koncepcijos sudarymui.

Žinomas vadybos specialistas, inžinierius ir matematikas V.Graičiūnas kartu su L.Arvilu 1933 m. yra matematiškai įrodę, kad vienas vadovas gali efektyviai vadovauti ne daugiau kaip 4–5 pavaldiniams. Įdomu, kad biurokratinės praktikos patvirtintas optimalus grupės dydis – 3÷11 narių – yra artimas skaičiams „7“ ir „9“, kurie įvairių tautų tautosakose simbolizuoja sąvoką „pakankamai daug“. O skaičius „3“ įvairiose pasakose simbolizuoja minimalų iteracijų, reikalingų norimam rezultatui pasiekti, skaičių. Tai rodo, jog šiuolaikinė sisteminė metodologija neblogai koreliuoja su gyvenimo praktikos pagimdyta liaudies išmintimi ir gali surasti savo ištakas ir pirmuosius savo daigelius ne tik kompiuterinės technikos ir informacinių technologijų amžiuje, bet ir gilioje senovėje.

Iš antrojo informacinio barjero principo išplaukia, jog tais atvejais, kai sudėtingos sistemos valdymui bei susijusių su tuo problemų sprendimui ribinių pajėgumų

$$P_{n^*} = kn^* = k\left(\frac{k}{c} + \frac{1}{2}\right) \text{ bei } P_{n^{**}} = k\left(\frac{2k}{c} + 1\right)$$

jau nebepakanka, reikia pradėti ne nuo valdymo personalo skaičiaus n , o nuo darbuotojų individualaus darbo našumo k didinimo. Didinant k , tiesė $P_n = kn$, sukdama apie koordinatinių sistemos pradžią – tašką 0 (žiūr. 7 pav.), kils į viršų ir antrąjį informacinį barjerą charakterizuojantys taškai n^* ir n^{**} slinks į dešinę, dėl ko atsiras galimybė didinti valdymo sistemos pajėgumą P_n ne tik didinant k , bet ir n iki naujo n^* bei n^{**} .

Kadangi žmogaus fizinės galimybės yra ribotos, tai žymiai padidinti valdymo personalo individualaus darbo našumą k pavyksta tik automatizuojant valdymo darbus, kuriant automatizuotas darbo vietas (ADV) ir jungiant jas į tinklus – kuriant automatizuoto valdymo sistemas (AVS) bei „kolektyvinio proto“ sistemas.*

Kadangi žmonės, sprenddami vis naujas problemas, susiduria su vis sudėtingesnių sistemų nagrinėjimu, projektavimu bei valdymu, tai jie visuomet susidurs ir su pajėgumo P_n nepakankamumo problema, kuri savo ruožtu reikalaus individualaus darbo našumo k didinimo ir taškų n^* bei n^{**} stūmimo į dešinę, nuo ko bendras ribinis pajėgumas P_n augs apytikriai proporcingai k augimo kvadratui. O tai reiškia, kad tolimesnis žmonijos progresas dėl antrojo informacinio barjero egzistavimo neįmanomas be intelektualaus darbo (tame skaičiuje ir valdymo) automatizavimo, kad **ADV ir AVS bei „kolektyvinio proto“ sistemų kūrimas yra ne tik pageidautina, bet ir neišvengiamai būtina sudėtinė žmonijos progreso dalis**. Tiesa, kai kurios anksčiau kurtos AVS nebuvo pakankamai efektyvios, tačiau tai buvo ne intelektualinio darbo automatizavimo ir AVS kūrimo idėjos klaidingumo išdava, o eilės klaidų, padarytų dėl subjektyvių ir objektyvių priežasčių**, kuriant tas AVS, rezultatas.

16. Problemos sudėtingumo mažinimo, mažinant nagrinėjamų santykių skaičių S_n , principas, teigiantis, jog **ne visi galimi santykiai S_n tarp sudėtingos sistemos elementų visada yra reikšmingi ir todėl, sugebant išskirti tik tuos santykius, kurie tikrai yra reikšmingi sprendžiant nagrinėjamą problemą, galima sumažinti tos problemos sprendimo sudėtingumą**.

Šio principo taikymo pavyzdys – sudėtingo darbų komplekso **tinkliniame grafike***** eilės darbų, priklausančių to grafiko kritiniam keliui, išskyrimas į pirmoje eilėje kontroliuojamųjų darbų grupę. Išskyrus kritinius darbus, komplekso darbų vadovui, kontroliuojančiam darbų vykdymo eigą, žymiai sumažėja pirmoje eilėje kontroliuojamųjų darbų skaičius, nes darbai, priklausantys kritiniams keliams, dažniausiai sudaro tik apie kelis ar keliolika procentų visų darbų. Kai darbų atlikimo eigoje dėl pasikeitusių aplinkybių keičiasi tinklinio grafiko kritiniai keliai, tai tuos pakeitimus tinkliniame grafike automatiškai atlieka kompiuterio programa, apdorojanti informaciją apie darbų eigą, ir ta pati programa

* A. Šukys. Matematinės sistemos tyros pradmenys (1.16.4. skyrelis). <http://www.mif.vu.lt/~sukys>

** A.Šukys. Pirmosios kartos [AVS neefektyvumo priežastys. Antrosios kartos [AVS projektavimo ir diegimo principai. Vilnius: VU, 1987

*** A. Šukys. Matematinės sistemos tyros pradmenys (1.9.2.16.–1.9.2.18. skyreliai). <http://www.mif.vu.lt/~sukys>

automatiškai pateikia darbų vadovui naują kritinių darbų sąrašą, tuo būdu padėdama vadovui atkreipti dėmesį į tuos darbus, į kuriuos tikrai reikėtų tą dėmesį atkreipti, negaištant laiko neaktualios šiuo momentu informacijos apie kitus darbus nagrinėjimui.

Kitas šio principo taikymo pavyzdys – **signalinės informacijos**, apie kurią čia bus kalbama šiek tiek vėliau, išskyrimas iš bendro informacijos srauto ir jos pateikimas vadovui, išrikiavus ją svarbos mažėjimo tvarka. Tai taikoma operatyviniuose panešimuose operatyvinio valdymo sistemose. Tuose pranešimuose signalinė informacija apie nepageidautinus nukrypimus nuo numatyto plano, kol tie nukrypimai dar nedideli, pradžioje pateikiama žemiausiojo ešelono vadovams ir tik tuomet, kai pastarieji nesugeba tų nukrypimų pašalinti ir nepageidautini nukrypimai toliau didėja, apie tai signalizuojama aukštesnio ešelono vadovams ir t.t. iki paties aukščiausiojo lygio. Todėl vadovai gauna informaciją apie nukrypimus (ir gauna būtinai) tik tada, kada jų nesugeba pašalinti jų pavaldiniai ir vadovo įsikišimas tampa būtinas. Toks signalinės informacijos pateikimo režimas taupo vadovų laiką, disciplinuoja pavaldinius ir reguliuoja vadovų įsikišimą į pavaldinių darbą taip, kad tas įsikišimas įvyksta nei per anksti, nei per vėlai.

Šis principas paaiškina, kodėl **patyrę darbuotojai, lyginant su nepatyrusiais, lengviau susidoroja su sudėtingomis problemomis**: jie, remdamiesi savo patirtimi, sugeba pakankamai greitai atskirti tai, kas tikrai svarbu, nuo to, kas nelabai arba visai nesvarbu, nepaskędami nereikšmingose ir nelabai reikšmingose smulkmenose. Kitaip tariant jie, sprenddami kolektyvui, susidedančiam iš n narių, iškilusią problemą, nagrinėja ne $S_n = \frac{n(n-1)}{2}$ abipusių ryšių, o žymiai mažiau, nes žino, jog ne visi galimi ryšiai būtinai yra svarbūs, ir sugeba atsirinkti tik svarbiuosius.

17. Problemų dekomponavimo ir lokalizavimo principas, teigiantis, jog sudarant sudėtingos sistemos išorinį modelį, kai dėl sistemos neapbrėptumo ir vidinių ryšių painumo nepavyksta nustatyti sistemos išėjimų ir jos naujos būsenos koordinačių reikšmių priklausomybės nuo įėjimų ir dabartinės būsenos koordinačių reikšmių, **tikslinga sudėtingas problemas dekomponuoti ir lokalizuoti**, t.y. tikslinga pereiti prie nagrinėjamos sistemos vidinio modelio sudarymo, išskiriant tos sistemos elementus, sudarant jų išorinius modelius ir po to juos sujungiant taip, kaip patys modeliuojamieji elementai yra sujungti realioje sistemoje, bei po to patikrinant, ar pasirinktas modelių sujungimo būdas tikrai modeliuoja emerđencijos reiškinius, egzistuojančius realioje sistemoje.

Šio principo taikymas suveda sudėtingų ir painių problemų, apimančių didelę sunkiai apžvelgiamą esybių, įvykių ir procesų aibę, surištą tarpusavyje painiais tiesioginiais ir grįžtamaisiais ryšiais, sprendimą į eilės **lokalių lengvai apžvelgiamų ir todėl lengviau sprendžiamų problemų spendimą**. Tokių lokalių problemų sprendimas yra žymiai paprastesnis ir lengvesnis, nes tampa apžvelgiamas ir jį gali atlikti kompleksinio kolektyvo nariai – atitinkamų sričių specialistai, vadovaujami sisteminio analitiko – dženeralisto, o sudarytų elementarių išorinių modelių jungimą į visos sistemos vidinį modelį ir to modelio funkcionavimo išbandymą, eksperimentuojant su juo bei jo dalimis įvairiuose režimuose galima kompiuterizuoti. Kolektyviai aptardami matematinių eksperimentų eigą ir jų rezultatus, gaunamus iš kompiuterio, projektuotojų bei modeliuotojų kolektyvo nariai žymiai greičiau ir lengviau gali išsiaiškinti reiškinius, vykstančius sudėtingoje sistemoje, jos funkcionavimo ypatumus ir dėsningumus, pastebėti ir ištaisyti padarytas klaidas.

Kolektyviai kuriant sudėtingos sistemos vidinį modelį, beveik neišvengiamai gali atsirasti praleidimų, nesuderinamumų ir kitokių klaidų. Jų lokalizavimo (aptikimo) palengvinimui ir jų skaičiaus sumažinimui ir skiriami sekantys 18÷22 principai bei iš jų išplaukiantys **problemų ir klaidų lokalizavimo metodai**, padedantys spręsti sudėtingumo problemą.

18. Terminalinės ir teleologinės metodologijų sintezės principas, teigiantis, jog terminalinė metodologija, besiremianti šiuolaikiniam mokslui žinomais gamtos dėsniais ir

besistengianti viską paaiškinti iš tų dėsnių išplaukiančiomis išvadamis, nustatydamą tarp reiškinių priežastinius ryšius, paimta atskirai, ir teleologinė metodologija, pradedanti bet kokią nagrinėjimą nuo subjektų tikslų nustatymo ir tų tikslų dekompozicijos, o po to pereinanti prie ryšių tarp subjektų tikslų ir jų veiksmų išsiaiškinimo bei tų veiksmų prognozavimo, paimtos atskirai, yra nepakankamos, tiriant organizacijų bei kitų sudėtingų sistemų, orientuotų į kokių nors tikslų pasiekimą, funkcionavimą ir bandant tą funkcionavimą suprasti bei prognozuoti, tačiau **tos metodologijos yra naudingos ir papildo viena kitą, todėl tikslinga ir reikalinga jų sintezė**, kaip tai ir daroma sisteminėje metodologijoje.

Naudojimasis šiuo principu padeda suprasti organizacijose bei kitose sudėtingose sistemose, orientuotose į kokių nors tikslų pasiekimą, vykstančius procesus ir tiksliau juos prognozuoti, efektyviau organizuoti eksperimentus su tokių sistemų imitaciniais modeliais, o taip pat suvokti metodologijų, naudojamų techninių sistemų projektavime, nepakankamumą, projektuojant organizacijas. **Nors dvejų metodologijų taikymas reikalauja daugiau laiko ir pastangų, tačiau tai atsiperka, nes padeda lokalizuoti galimas klaidas bei apsirikimus ir tokiu būdu sumažinti jų skaičių, praplečia požiūrį į sprendžiamą problemą, padidina sprendėjų euristines galimybes.**

19. Mechanistinės ir organizminės metodologijų sintezės principas, teigiantis, jog mechanistinė metodologija, kuria vadovaujamasi surenkant sistemą iš jos dalių, paimta atskirai, ir organizminė metodologija, traktuojanti organizaciją kaip organizmą, kuriame atskirų dalių veikimą galima suprasti, tik nustačius jų vaidmenį viso organizmo funkcionavime, paimtos atskirai, yra nepakankamos, tiriant ir projektuojant organizacijas bei kitas sudėtingas sistemas, **tačiau tos metodologijos yra naudingos ir papildo viena kitą, todėl tikslinga ir reikalinga jų sintezė**, kaip ir daroma, taikant sisteminę metodologiją.

Naudojimasis šiuo principu padeda sumažinti skaičių klaidų, galinčių atsirasti dėl emerđencijos reiškinių, vykdant sudėtingos sistemos dekompoziciją ir sintezę jos vidinio deskriptyvinio modelio bei projekto kūrimo procesuose.

20. Funkcinės ir objektinės metodologijų sintezės principas, teigiantis, jog funkcinė metodologija, kuria vadovaujamasi, sudarant organizacijos tikslų medžius ir funkcijų paskirstymo bei jų hierarchijos struktūrines schemas, paimta atskirai, ir objektinė metodologija, kuria vadovaujantis organizacijoje išskiriamos jos sudėtį charakterizuojančios esybės, nustatomi jų atributai bei esybių tarpusavio ryšiai, paimtos atskirai, yra nepakankamos, tiriant ir projektuojant organizacijas bei kitas sudėtingas sistemas, tačiau **tos metodologijos yra naudingos ir papildo viena kitą, todėl tikslinga ir reikalinga jų sintezė**, kaip ir daroma sisteminėje metodologijoje.

Naudojimasis šiuo principu padeda sumažinti skaičių tų klaidų, kurios gana dažnai padaromos, vykdant sudėtingos sistemos dekompoziciją ir sintezę jos vidinio deskriptyvinio modelio bei projekto kūrimo procesuose, apsiribojus vien tik struktūros ar vien tik sudėties elementų nagrinėjimu. Lyginant modelius, sudarytus taikant funkcinę ir objektinę metodologijas, **tos klaidos yra lokalizuojamos ir todėl nesunkiai surandamos ir ištaisomos.**

21. Deskriptyvinio ir normatyvinio modeliavimo metodų sintezės principas, teigiantis, jog tobulinant kokią nors sudėtingą sistemą bei projektuojant kokios organizacijos AVS, **tikslinga kurti tobulinamų bei projektuojamų sistemų ir jų dalių deskriptyvinius ir normatyvinius modelius, lyginti juos vieną su kitu, analizuoti randamus skirtumus ir, tokiu būdu išsiaiškinant padarytas projektavimo klaidas bei apsirikimus, koreguoti kuriamą sudėtingą projektą, artinti esamą sistemą prie pageidautinos.**

22. Metodų „nuo viršaus“ ir „nuo apačios“ sintezės principas, teigiantis, jog, kuriant bei tobulinant sudėtingą sistemą, nė vieno iš čia išvardintų metodų, paimtų atskirai, nepakanka, kad šie metodai gali ir privalo vienas kitą papildyti ir todėl **reikalinga abiejų šių metodų sintezė, kaip tai ir realizuojama sisteminėje metodologijoje.**

Priešprojektinėje stadijoje sudėtingos sistemos dekompozicija pradedama „nuo viršaus“, tačiau priešprojektinių tyrimų etape jau naudojamos abiejų šių vienas kitą

papildančių ir koreguojančių vienas kito rezultatus metodų sinteze.

Techninio projektavimo etape pirmenybė atiduodama metodui „nuo viršaus“, o metodu „nuo apačios“ naudojamosi tik fragmentiškai, kaip pagalbine priemone, išsiaiškinant bei patikslinant tuos projektinius sprendimus, kurie buvo nelabai aiškūs ir kėlė abejonių. Tokie išsiaiškinimai tęsiami ir techninių projektinių sprendimų testavimo bei jų modeliavimo etape. Jie pradkami „nuo apačios“, nes, prieš pradėdant rinkti sudėtingą sistemą iš jos dalių ir tikrinti ją, tikslinga turėti patikrintas jos dalis.

Darbinio projektavimo metu naudojamosi abiejų metodų sinteze, tačiau didžioji dalis šio etapo darbų pradama „nuo apačios“, t.y. nuo sistemos elementų ir jų pagaminimo technologinių procesų darbinio projektų, nes, tik sudarius elementų darbinio projektus, galima projektuoti sistemos posistemius ir jų surinkimo (pagaminimo, suorganizavimo) technologinius procesus. Tai dar vienas darbinio ir techninio projektavimo skirtumas, nes, kuriant techninį projektą, pradama „nuo viršaus“: pradžioje stengiamasi įsivaizduoti (surasti, išrasti) visos sistemos veikimo principą ir susidaryti jos koncepcinį modelį, iš ko po to išplaukia detalesnis sistemos įvaizdis, nusakantis, iš kokių dalių ta sistema galėtų būti sudaryta ir kaip tos dalys turėtų būti sujungtos. Darbinio projektavimo metu tie darbai atliekami atvirkštine tvarka: pradama nuo detalių darbinio projektavimo ir pagaminimo ir baigiama visos sistemos darbinio projekto sudarymu ir jos surinkimu bei išbandymu.

„Nuo apačios“ pradėdamas ir projekto diegimas bei įdiegtos sistemos testavimas, nes tik pagaminus ir patikrinus smulkesnes naujos sistemos dalis, gaminamos (diegiamos) ir tikrinamos stambesnės sistemos dalys ir, galų gale, visa sistema.

23. Sudėtingos sistemos kūrimo palaipsniškumo principas, teigiantis, jog sudėtingos sistemos kūrimas – daugiaetapis procesas, kuriame, pereinant nuo etapo prie etapo, palaipsniui išryškėja vis daugiau ir vis smulkesnių projekto detalių ir ryšių tarp jų. Tų detalių ir ryšių pradiniuose projektavimo etapuose numatyti dar neįmanoma, todėl tuose etapuose to ir nesiekama. **Kiekvienas sistemos kūrimo etapas, pasiekdamas sisteminėje metodologijoje numatytus dalinius rezultatus, sudaro sąlygas naujame etape kelti tame etape numatomus naujus uždavinius, spręsti juos ir dar labiau priartėti prie galutinio tikslo – naujos sistemos sukūrimo.** Tokiu būdu, žingsnis po žingsnio, etapas po etapo, pradėdant nuo pačių bendriausių ir pradžioje net nelabai aiškių euristicinių idėjų, jas konkretizuojant, detalizuojant, papildant smulkesnėmis ir konkretesnėmis euristinėmis idėjomis (atradiniais bei išradiniais), palaipsniui pereinama prie sistemos techninio projekto, po to – prie jos darbinio projekto ir to projekto realizavimo. Tą palaipsnišką judėjimą nuo projekto idėjos jos realizavimo link stambiais blokais vaizduoja 6 pav. (žiūr. 21 psl.). Dėl sudėtingos sistemos projektavimo proceso iteratyvumo tas judėjimas nėra toks paprastas ir tiesiaiegis, kaip galėtų pasirodyti iš čia ką tik pateikto aprašymo: palaipsniui judant pirmyn (6 pav. blokų numerių didėjimo link), ne kartą tenka grįžti atgal, pakoreguoti ankstesnių blokų rezultatus ir, remiantis pakoreguotais rezultatais, vėl judėti pirmyn. Dėl tokių daugkartinių grįžimų atgal atskiri etapai kartais taip persipina, jog tampa ne taip paprasta atsakyti į klausimą, kurio etapo darbai dabartiniu metu daromi. Pavyzdžiui, jei projektuojamoji ar tobulinamoji sudėtinga sistema priklauso organizacijų tipui, tai gana dažnai dėl jau minėtojo iteratyvumo ir iš jo išplaukiančio etapų persipynimo beveik išnyksta riba tarp projektavimo ir diegimo stadijų. Juk tokiais atvejais neretai atsitinka taip, kad tuo metu, kai vienos mūsų kuriamos sistemos sudėtinės dalys dar tik projektuojamos, kitos jau diegiamos ar net eksploatuojamos. O ir **pats projektavimas, jei jame aktyviai dalyvauja organizacijos personalas, yra tuo pačiu metu ir naujos sistemos diegimo dalis, nes organizacijos personalas, projektuodamas naują sistemą ar jos patobulinimus, keičia pats save (savo mąstyseną, savo įgūdžius) ir tuo pačiu kuria pačią svarbiausiąją naujos organizacinės sistemos dalį – „žmogiškąją“ sistemos „žmogus –technika“ sudedamąją dalį.**

Be to, projektuojant ir diegiant sudėtingos sistemos (didelės įmonės, atominės elektrinės, ligoninės ir pan.) projektą eilėmis, išlošiama ir ekonomiškai: kai kurios

sistemos dalys pradeda funkcionuoti ir duoti naudą anksčiau, negu bus baigtas visos sistemos projektavimas ir pilnas jos įdiegimas.

24. Iš ką tik suformuluotojo sudėtingos sistemos kūrimo palaipsniškumo principo išplaukia **sudėtingos organizacijos ir jos valdymo sistemos tobulinimo projektavimo ir diegimo procesų „lygiagretumo“ principas**, teigiantis, jog, **tobulinant sudėtingą organizaciją bei jos valdymo sistemą, netikslinga laukti, kol visi projektavimo darbai bus pilnai užbaigti, ir tik po to pradėti viso projekto įgyvendinimą (diegimą).** Kai kuriuos projektinius sprendimus galima pradėti realizuoti žymiai anksčiau ir tai duoda **trigubą naudą**: 1) projektuotojai gauna papildomą informaciją apie kuriamo projekto kokybę, jo stipriąsias ir silpnąsias vietas, o tai labai padeda likusios projekto dalies projektavime bei viso projekto koregavime; 2) tie projektiniai sprendimai, kuriuos jau galima įgyvendinti, nelaukiant viso projekto pasirodymo, įgyvendinami anksčiau ir anksčiau pradeda duoti naudą (nors, galbūt, ir mažesnę, negu ta, kurią duos, įdiegus visą projektą, tačiau – vis dėlto naudą); 3) organizacijos personalas palengva įpranta dirbti naujomis sąlygomis, ir todėl viso projekto įdiegimas vėliau tampa lengvesnis, mažiau panašus į „revoliuciją“ su visais jos stresais.

Taikant šį principą, pavojinga persistengti: kol dar nesudarytas bendras visos sistemos techninis projektas, apsprendžiantis, kokius sisteminius reikalavimus turės tenkinti kiekviena projekto dalis, atskirų projekto dalių darbinio projektavimo ir realizavimo forsavimas gali privesti prie to, kad vėliau, diegiant pilną projektą, didelę dalį anksčiau įdiegtų projekto dalių reiks kardinaliai keisti, o tai – ir papildomos išlaidos ir psichologinis diskomfortas.

Projektavimo ir diegimo procesų „lygiagretumo“ principas gali būti taikomas ir didelėms sudėtingoms techninėms sistemoms (pavyzdžiui, atominėms elektrinėms ir pan.), kurių projektavimą ir diegimą galima ir tikslinga organizuoti „eilėmis“ (pirmoje „eilėje“ – pirmasis blokas, po to – antrasis ir t.t.).

Kalbant apie projektavimo ir diegimo procesų „lygiagretumą“, nereikia pamiršti, jog tai – ypatingas lygiagretumas, kadangi diegimas, vykdamas lygiagrečiai su projektavimo procesu, vis dėlto atsilieka nuo jo, nes diegiama tai, kas jau suprojektuota, o to diegimo metu kitų pilno projektų dalių projektavimo procesas vėl pasislenka pirmyn.

25. **Organizacijos personalo dalyvavimo organizacijos bei jos valdymo sistemos tobulinimo projektavime ir diegime tikslingumo principas**, teigiantis, jog **į organizacijos bei jos valdymo sistemos tobulinimo projektavimo procesą tikslinga maksimaliai įtraukti organizacijos personalą, pirmoje eilėje – organizacijos ir jos padalinių vadovus, o po to – ir tuos eilinius darbuotojus, kurių funkcijos bei jų vykdymo technologija keisis, diegiant projekte numatomus patobulinimus.**

Personalo įtraukimo į projektavimo procesą tikslingumą sąlygoja trys pagrindinės priežastys.

Pirmosios priežasties esmę sudaro tai, jog net geriausi projektai gali likti popieriuje, jei jie nebus suprasti ir palaikyti tų žmonių, kurie tuos projektus turės įgyvendinti, tų žmonių, kuriems teks projekte numatytais patobulinimais naudotis. Organizacija tuo ir skiriasi nuo techninės sistemos, kad pagrindinę organizacijos dalį sudaro žmonės – organizacijos personalas, ir todėl, diegiant organizacijoje kokią nors naują, būtina paruošti ne tik techninę, bet ir „žmogiškąją“ tos naujovės dalį, t.y. pakeisti žmonių galvose senus įsitikinimus ir senas veiklos programas į naujus įsitikinimus ir naujas veiklos programas, pakeisti senus įgūdžius į naujus. Tai labai nelengvas darbas, nes, jį vykdant, nepakanka, kad kokie nors aukštos kvalifikacijos specialistai, gerai žinantys organizacijos darbuotojų darbą, suprojektuotų naujoviškas to darbo vykdymo technologijas ir parašytų detales, darbuotojams gerai suprantamas instrukcijas, kuriomis vadovaudamiesi šie pradėtų dirbti naujoviškai. Nepakanka todėl, kad, kaip jau buvo minėta trečiame šios metodinės priemonės skyrelyje, aprašant aktyviųjų elementų savybes (žiūr. 7÷8 psl.), aktyviųjų elementų veiksmų tiesioginis valdymas yra neefektyvus. Daug efektyviau valdyti aktyviųjų elementų veiksmus netiesiogiai – per jų tikslų, jų interesų, jų įsitikinimų valdymą. Reikia sudaryti

tokias sąlygas, kad aktyvieji elementai patys norėtų daryti tai, ko reikia organizacijai, ir daryti ne blogiau, kaip tai numatoma normatyvinėje technologinėje dokumentacijoje (TNOD). **Jeigu mes norime, kad organizacijos veikloje įvyktų kokie nors pasikeitimai, mes turime pasiekti, kad jie įvyktų organizacijos darbuotojų galvose.** Žinoma, galima siekti tokių pasikeitimų, verčiant tuos darbuotojus studijuoti kažkieno parašytus projektų toms bei instrukcijas, tačiau toks metodas labai neefektyvus ir gali iššaukti visai nepageidautiną atvirkštinį psichologinį efektą – darbuotojų priešišumą diegiamoms naujovėms. Žymiai efektyvesnis ne pasyvusis, o aktyvusis kelias: reikia stengtis įtraukti organizacijos darbuotojus į naujovių inicijavimo, jų projektavimo ir diegimo procesą. To proceso metu darbuotojai geriau supras naujovių esmę, jų reikalingumą bei naudingumą ir jaus save tų naujovių kūrimo dalyviais, laikys jas „savomis“. Tuomet jie ne tik nesipriešins (pasyviai ar aktyviai) naujovių diegimui, o padės tai daryti, kartais net patys generuodami naujas idėjas ir inicijuodami jų įgyvendinimą. **Be organizacijos darbuotojų tokios veiklos, be jų aktyvumo ir suinteresuotumo pažadinimo net ir geriausiai sudaryti projektai liks neįgyvendinti,** o tų projektų tomiai – iki galo neperskaityti ir neįsisavinti. Gaila, kad šią elementarią tiesą, kurią gyvenimo praktika yra jau ne vieną kartą patvirtinusi, nepakankamai įsisavino kai kurie net labai svarbių organizacijų vadovai bei atsakingi darbuotojai. Pavyzdžiui, Lietuvos Vyriausybė ir Seimas, matyt, iki šiol dar neįvertino, kokią didelę reikšmę visuomenės valdymo ir pilietinės visuomenės kūrimo procese turi kuo didesnis visuomenės įtraukimas į valstybės valdymą, į valstybėje iškilančių problemų sprendimą. Neužtenka sukurti ir paskelbti gerus įstatymus (nors tai ir nelengva, nes dėl įvairių priežasčių kai kurie įstatymai bent jau pirmose jų redakcijose nebūna pakankamai geri) bei nutarimus. **Reikia dar kad visuomenė norėtų ir sugebėtų juos vykdyti,** o tam visų pirma reikia, kad visuomenė juos žinotų, suprastų ir priimtų kaip visuomenei naudingus, vertus tų pastangų, kurios reikalingos tų įstatymų bei nutarimų įgyvendinimui. Ir nėra geresnio būdo visuomenės pilietiškumo ugdymui, kaip visuomenės įtraukimas į valstybei svarbių problemų sprendimą, į ruošiamų įstatymų bei nutarimų projektų parengimą bei svarstymą, į jų įgyvendinimo kontrolę. Tai suprato net sovietinė valdžia, kuri, imituodama liaudies dalyvavimą valstybės valdyme, net varu varydavo organizacijų darbuotojus į susirinkimus, skirtus tų ar kitų nutarimų projektų svarstymui. **Atmetant sovietinį palikimą, kartu su daugybe tikrai atmestinių dalykų buvo atmesta ir visuomenės dalyvavimo valstybės valdyme idėja, kuri, bendrai paėmus, nėra bloga ir išreiškia vieną iš pagrindinių demokratijos bruožų.** Daugelyje demokratinių valstybių (JAV, Švedijoje, Šveicarijoje ir t.t.) egzistuoja įstatymai, įteisinantys visuomeninį auditą (visuomeninę kontrolę) ir visuomenės dalyvavimą valstybei svarbių problemų sprendimo projektų svarstyme. Pas mus gi visuomenės dalyvavimo valstybės valdyme idėja buvo išjuokta, sutapatinus ją su Lenino aforizmu, kad „kiekviena virėja galės valdyti valstybę“, ir atmesta kaip „sovietinis reliktas“. **Kartu su tuo pradėjo reikštis ir viešumo mažėjimo tendencija: apie Vyriausybės priimamus sprendimus ir nutarimus visuomenė dabar sužino (jei iš viso sužino) tik po to, kai tie sprendimai ir nutarimai jau priimti. Šiuo atžvilgiu pas mus viešumo dabar net mažiau, negu M. S. Gorbačiovo pertvarkos laikais.** Todėl nenuostabu, kad dabar Lietuvos visuomenė tapo tokia abejinga ir nepakankamai pagarbiai žiūri į įstatymus bei Vyriausybės nutarimus, nelabai stropiai juos vykdo, o kartais ir visai nevykdo, nelaikydama jų „savais“ ir reikalingais visuomenei.

Yra tik du būdai valdyti visuomenę: 1) jėga priversti ją daryti tai, ko valdžiai reikia; 2) įtikinti visuomenę, jog įstatymai ir nutarimai, kuriuos priima valstybės valdymo institucijos, yra reikalingi ir naudingi pačiai visuomenei. Pirmajam būdai įgyvendinti reikia galingų represinių struktūrų, kurias pačias reikia valdyti, be to pirmasis būdas yra pavojingas tuo, kad gali privesti prie demokratijos panaikinimo ir diktatūros įvedimo, todėl tą būdą tikslinga naudoti tik siauruose rėmuose – tik kovoje su nusikalstamomis tendencijomis ir nusikaltėliais. Antrasis būdas yra žymiai universalesnis ir efektyvesnis, tačiau jo įgyvendinimui reikalinga, kad valstybės valdymo institucijų priimami įstatymai bei nutarimai būtų tikrai reikalingi ir naudingi visuomenei ir kad visuomenė tai

suprastų ir norėtų tuos įstatymus bei nutarimus vykdyti. Tai pasiekti galima, tik įtraukiant visuomenę į įstatymų bei nutarimų projektų kūrimo procesus ir į įstatymų bei nutarimų vykdymo kontrolės procesus. Visa tai, kas čia buvo pasakyta valstybės atžvilgiu, galioja ir bet kokios organizacijos atžvilgiu, o iš to ir išplaukia mūsų nagrinėjamojo principo reikalingumo pirmoji (ir pagrindinė) priežastis.

Antrąją priežastį sudaro tai, kad organizacijos darbuotojai organizacijos veiklos specifiką žino geriau, negu bet kurie iš šalies pakviesti specialistai – organizacijos veiklos ir jos valdymo sistemos tobulinimo projektuotojai. **Projektavimo specialistų ir organizacijos darbuotojų simbiozė, tobulinant organizacijos veiklą ir valdymą, yra labai naudinga abiem šioms grupėms ir projekto kokybei, nes tos grupės viena kitą papildo.** Be to, organizacijos darbuotojai, dalyvaudami projektavime, geriau supranta tas naujoves, kurios bus diegiamos, ir todėl palengvėja patobulintos sistemos diegimo procesas. Iš kitos pusės, specialistų projektuotojų bendradarbiavimas su organizacijos darbuotojais priartina tuos specialistus prie gyvenimo tikrovės, daro jų sprendimus praktiškesniais.

Trečioji priežastis glūdi tame, kad sudėtingos organizacijos veiklos ir jos valdymo tobulinimas – labai sudėtingas ir didelis darbas, kuriam įvykdyti reikia labai daug laiko. Organizacijos darbuotojų įtraukimas į šį darbą leidžia žymiai padidinti jo tempus, o taip pat sumažinti reikalingų iteracijų skaičių, o tai leidžia žymiai sutrumpinti viso projektavimo ir diegimo trukmę bei sumažinti išlaidas.

Organizacijos darbuotojų dalyvavimo projektavimo procesuose principo aprašymui čia skirta gana daug vietos todėl, kad šių eilučių autoriui labai norėjosi įtikinti skaitytoją šio principo reikalavimų ir iš jo išplaukiančių rekomendacijų laikymosi būtinumu, nes iki šiol didelė organizacijų vadovų ir darbuotojų dalis ne tik to nesupranta, bet yra iš anksto psichologiškai nusiteikusi prieš šį principą. Todėl šio principo reikalavimai praktikoje įgyvendinami dar gana retai, nors praktika rodo, jog šio principo taikymas duoda stebėtinai gerus rezultatus.

26. Sudėtingų organizuotų sistemų matematinio modeliavimo principas, teigiantis, jog sudėtingų organizuotų sistemų matematiniam modeliavimui jau nebepakanka analitinių klasikinės matematikos metodų, susiformavusių tiriant ir modeliuojant paprastas sistemas, o taip pat nepakanka ir tikimybių teorijos bei matematinės statistikos metodų, susiformavusių tiriant atsitiktinius procesus ir dideles neorganizuotų objektų aibes, nes sudėtingose organizuotose sistemose jų elementai nesielia nepriklausomai ir visai atsitiktinai. Nepakanka ir paprastos šių metodų sintezės, nes tokia sintezė, nors ir atveria daug naujų matematikos taikymo galimybių, neišsprendžia didelės sudėtingai organizuotos sistemos neapžvelgiamumo problemos. Todėl **sudėtingų organizuotų sistemų matematiniam aprašymui, jų matematiniam modeliavimui ir matematiniam tyrinėjimui reikia kurti specialius metodus, sudėtingesnius, negu klasikinės matematikos, tikimybių teorijos ir matematinės statistikos metodai.** Tokių metodų kūrimo galimybė atsiranda, pasitelkiant kompiuterinę techniką ir informacines technologijas, kuriant programinę įrangą, leidžiančią matematikų ir kompiuterių kolektyvams dirbti dialogo režime. Vienas tokių metodų yra sudėtingų sistemų funkcionavimo (sudėtingų procesų) **imitacinis modeliavimas, matematiniai eksperimentai su imitaciniais modeliais ir tų eksperimentų metu gautos informacijos matematinis apdorojimas matematinės statistikos ir klasikinės matematikos metodais.** Naudojantis imitacinio modeliavimo metodu, galima prognozuoti įvairių poveikių sudėtingoms sistemoms pasekmes ir išsirinkti iš kelių sudėtingos sistemos valdymo variantų geriausiąjį*. Imitacinį modeliavimą galima naudoti ir įvairių hipotezių adekvatumo tikrovei tikrinime ir sudėtingų projektų testavime. Matematiniai eksperimentai su

* žiūr. antrąją šios metodinės priemonės priedą „Sisteminės analizės ir sistemų modeliavimo taikymas valdymo tobulinime“, kuriame aprašyta valdymo sprendimų pasekmių modeliavimo ir geriausių sprendimo variantų išrinkimo metodika, tinkanti ir projektinių sprendimų testavimui bei optimizavimui.

imitaciniais modeliais neretai gali pakeisti brangius ir kartais net rizikingus natūrinius eksperimentus su realiomis sistemomis. Apibendrinant galima pasakyti, jog **imitacinis modeliavimas ir matematiniai eksperimentai su imitaciniais modeliais žymiai išplečia matematinių metodų taikymo sudėtingų sistemų tyrime bei projektavime galimybes ir yra naujas reikšmingas ir turintis dideles perspektyvas kokybinis žingsnis matematikos taikyme**. Eksperimentai su matematiniais modeliais, naudojantis kompiuterine technika, yra naujas kokybinis žingsnis ir pačioje matematikoje: iki šiol eksperimentuodavo fizikai, chemikai, biologai ir pan., o matematika buvo laikoma grynai deduciniu mokslu, kur viskas turėjo būti įrodyta, remiantis pradinėmis aksiomomis ir griežta logika, nesinaudojant jokiais eksperimentais. Eksperimento rezultatai matematikoje galėjo būti naudojami, tik įrodant kokios nors formulės ar teoremos neuniversalumą, t.y. tik paneigiant kokį nors teiginį, bet ne įrodant jo teisingumą. Todėl buvo toks mokslas kaip, pavyzdžiui, empirinė fizika, tačiau nebuvo ir negalėjo būti empirinės matematikos. Dabar, atrodo, atsiranda prielaidos ir tokiam mokslui susiformuoti, nes galima eksperimentuoti ne tik su realių sistemų deskriptyviniais ir normatyviniais modeliais, bet ir su grynai matematiniais modeliais – matematinėmis abstrakcijomis, su virtualių sistemų matematiniais modeliais.

Išvardindamas ir trumpai charakterizuodamas šioje metodinėje priemonėje kai kuriuos bendrasisteminius principus, jos autorius vadovavosi nuostata, jog šių principų žinojimas palengvins SIS kūrėjams sisteminio mąstymo ir sisteminės projektavimo metodologijos įsisavinimą. Žinoma, buvo galima parekomenduoti skaitytojams, kad jie perskaitytų pirmąją iš šios priemonės septynioliktajame puslapyje išvardintų knygutėlių, kurioje tie principai aprašyti žymiai išsamiau, tačiau ta knygutė parašyta rusų kalba ir Lietuvoje ją gauti nėra labai paprasta. Suprasdamas, kad perskaityti šią knygutę ryšis tik labai nedidelė dalis šios metodinės priemonės skaitytojų (jei iš viso kas nors ryšis), šių eilučių autorius ir nusprendė dalį metodinės priemonės puslapių paskirti bendrasisteminiams principams, nors, savaime suprantama, labai norėtų, kad perskaičiusių minėtąją knygutę atsirastų kuo daugiau, nes su jais būtų žymiai lengviau dirbti, kuriant SIS, o vėliau – ir VADIS. Juk puiku, kai kartu dirbantys supranta vienas kitą „iš pusės žodžio“!

Bendrasisteminiai principai sudaro tik dalį principų, kuriais naudojasi sisteminė organizacijų ir jų valdymo sistemų tobulinimo ir projektavimo metodologija. Yra gana daug kitų principų. Tai **objekto paruošimo automatizavimui, kibernetinių, normatyvinių organizacinių struktūrų projektavimo ir diegimo, informacinių sistemų projektavimo ir diegimo, organizacijos ir jos automatizuoto valdymo sistemų matematinės (tame tarpe ir programinės) įrangos projektavimo ir diegimo, automatizuotu sistemų techninės įrangos projektavimo ir diegimo principų grupės**. Apie 70 principų, priklausančių šioms grupėms, yra aprašyta antrojoje iš septynioliktajame puslapyje paminėtų knygutėlių. Pastaroji knygutė parašyta lietuvių kalba ir ją gauti Lietuvoje nėra sunku. Todėl nėra reikalo kartoti jos tekstą šioje metodinėje priemonėje, tuo labiau, kad tai užimtų labai daug vietos, nes knygutės apimtis – 232 puslapiai. **Šią knygutę privalėtų perskaityti visi SIS projektuotojai ir SIS kūrimo koordinacinės grupės nariai, nes tai padėtų jiems išvengti daugybės tipinių klaidų ir efektyviau organizuoti darbą, sukurti žymiai efektyvesnę sistemą**. Juk toje knygutėje aprašyti principai yra daugelio projektuotojų daugiametės patirties apibendrinimas. Ši patirtis buvo sukaupta, darant daugybę klaidų ir retkarčiais pasiekiant sėkmingų rezultatų, ir ji galėjo atsirasti tik praėjus ilgą dažnų nusivylimų ir gana retų laimėjimų kelią, siekiant užsibrėžtų tikslų. Todėl tos patirties žinojimas gali žymiai sutrumpinti SIS kūrimo trukmę ir padėti SIS kūrėjams išvengti didelės dalies nusivylimų, neišvengiamų kiekvieno kūrėjo kelyje. Norėtusi atkreipti SIS kūrėjų dėmesį į tai, jog **antrojoje knygutėje aprašytieji principai, būdami žymiai siauresni ir labiau specializuoti, negu bendrasisteminiai, yra žymiai praktiškesni ir konkretesni, negu pastarieji, ir todėl jų taikymas tuoj pat duoda apčiuopiamą efektą**. O tai gana svarbu ilgame sudėtingos sistemos kūrimo kelyje, nes dauguma žmonių yra nekantrūs ir nori kuo greičiau matyti padaryto darbo rezultatus. Bendrasisteminį principų paskirtis –formuoti projektuotojų sisteminį mąstymą („sisteminę

filosofiją“), padėti projektuotojams įsisavinti bendrasisteminius sisteminės metodologijos pagrindus. Tai labai svarbu, tačiau tai reikalauja gana daug laiko ir rezultatų (nors ir gana svarių) duoda ne taip jau greit. **Principai, aprašytieji ką tik minėtoje antrojoje knygutėje, yra daugiau instrumentiniai negu fundamentiniai**, t.y. skirti konkrečioms nurodymams, ką ir kaip daryti ir ko nedaryti. Todėl tų principų taikymo praktinė nauda yra labiau akivaizdi ir greičiau pajuntama. Manau, jog SIS kūrėjai, suradę laiko ir kantrybės perskaityti tą (antrąją) knygutę, nenusivils ja. O tos knygutės 3.4 skyrelį (109–150 psl.), skirtą organizacijos struktūros ir joje vykdomų darbų technologijos tobulinimui bei SNOD, TNOD ir PNOD projektavimui, manau, būtų naudinga perskaityti ir visiems kitiems Statistikos departamento padalinių vadovams, o taip pat darbuotojams, norintiems sudaryti kokybiškas pareigines instrukcijas (pareigybės pasus). Norėdamas šiek tiek padėti tiems darbuotojams, sudariau „Pareigybės paso (PP) projektavimo metodinius nurodymus“, kuriuos pridėdu prie šios metodinės priemonės kaip jos pirmąjį priedą, nors, žinoma, tais nurodymais galima naudotis ir atskirai – be šios metodinės priemonės.

Žinoma, visuomet atsiras skeptikų, kurie pasakys: „O kas iš to, jei aš, sugaišęs gana daug laiko, įsisavinsiu visus tuos principus ir metodologijas? Juk jie mano darbo už mane vis tiek nepadarys“. Ir tokie skeptikai bus teisūs. Jei Jūs susiruošėte į alpinistinį žygį, norėdami įkopti į aukštą viršukalnę, kurios apylinkėse anksčiau niekad nesate buvęs, tai nepriklausomai nuo to, ar Jūs turite ar neturite tų apylinkių žemėlapius, žinote ar nežinote tą maršrutą, kuriuo kopsite į tą miglotuose toliuose spindinčią sunkiai pasiekiamą ledinę viršūnę, visą kelią iki pačios viršūnės teks praeiti savomis kojomis, nešant sunkią kuprinę ant savų pečių. Žemėlapių turėjimas ir maršruto žinojimas automatiškai neužkels Jūsų ant to kalno, nepalengvins kuprinės svorio. Tačiau žemėlapių turėjimas ir maršruto žinojimas gali žymiai padidinti tikimybę, jog Jūs sėkmingai – gyvas ir sveikas – pasieksite užsibrėžtą tikslą, lyginant su tuo, jei Jūs leistumėtės į šį rizikingą žygį be žemėlapių ir be gerai išnagrinėto maršruto. Tai žino kiekvienas nors šiek tiek patyręs alpinistas ir todėl, prieš pradėdamas žygį, rimtai ir kruopščiai ruošiasi jam. O juk tie žemėlapiai ir maršrutai – tai juk irgi patirties rezultatai, už kuriuos daugelis ankstesnių keliautojų sumokėjo gana nemažą (o kai kas, galbūt, ir aukščiausią) kainą. SIS kūrimas – tai irgi kopimas į aukštą kalną maršrutu, kuriuo daugelis iš jos kūrėjų dar nebuvo ejęs. O visi čia aprašomi principai ir metodiniai nurodymai – tai ankstesnių keliautojų patirties apibendrinimas ir iš jo išplaukiantys geranoriški patarimai, padedantys greičiau ir lengviau pasiekti tikslą, išvengti kelyje tykančių pavojų. Žinoma, tie patarimai neužkels Jūsų į kalną automatiškai ir negali garantuoti, kad Jūs, padaręs neatsakingą žingsnį, nenugarmėsite į kokią nors prarają, tačiau jie gali žymiai sumažinti tokios nelaimingos baigties tikimybę.

Kartais klausiama, kodėl tų principų tiek daug, kodėl negalima sumažinti jų skaičiaus, kad būtų lengviau juose orientuotis ir jais naudotis. Juk klasikinės mechanikos pagrindą sudaro tik keletas Niutono dėsnių, klasikinė elektrodinamika remiasi tik keliomis Maksvelo formulėmis, klasikinė termodinamika turi tik keletą principų. Kodėl to nepadaryta sistemoje?

Žinoma, būtų gerai, jei visą sisteminę metodologiją būtų galima logiškai išvesti iš kelių labai fundamentalių principų, kaip tai daroma klasikiniuose moksluose. Ir tokių fundamentalių principų reikia ieškoti, suvedant mažiau universalius principus į universalesnius. Tačiau **sukurti sisteminės metodologijos principų sistemą, susidedančią tik iš labai nedidelio skaičiaus principų, niekad nepavyks, nes sisteminė metodologija turi reikalą su sudėtingomis ir supersudėtingomis sistemomis, kurių savybės ir funkcionavimo dėsningumus išvesti iš kelių fundamentalių principų, kaip tai daroma klasikiniuose moksluose, nagrinėjančiuose paprastas sistemas, iš principo neįmanoma. Sudėtinga sistema todėl ir vadinama sudėtinga, kad jos funkcionavimo dėsningumų mes negalime išvesti iš keleto paprastų principų.** Jeigu tai pavyksta, tai tokia sistema mums tampa paprasta ir mes ją išbraukiame iš sudėtingų sistemų sąrašo. Tačiau tas sąrašas niekad nelieka tuščias, o atvirkščiai – vis labiau pasipildo. **Ir susidurdami su naujomis sudėtingomis sistemomis, bandydami jas modeliuoti ir projektuoti, mes gana greitai**

pastebime, jog tam reikalingų principų mes turime ne tik ne per daug, bet per mažai, kad reikia ieškoti naujų principų, papildančių jau turimą jų sąrašą. Ir tas sąrašas niekad nebus pakankamai pilnas, nes supantis mus pasaulis yra neišsemiamas ir visi jo egzistavimo ir funkcionavimo dėsningumai, matyt, negali būti išvesti iš baigtinio skaičiaus principų.

Todėl, iš vienos pusės, nagrinėjant sudėtingą realųjį pasaulį, jo funkcionavimą aprašančių, jo modeliavimui bei naujų jo dalių projektavimui padedančių principų sąrašą reikia plėsti, atrandant vis naujus dėsningumus, iš kitos pusės, jau turimus principus reikia sistematizuoti, bandyti juos apibendrinti, suvesti į universalesnius ir tokiu būdu paprastinti turimą principų sistemą, daryti ją labiau apžvelgiamą. Abi šios tendencijos yra naudingos, neneigia viena kitos, o viena kitą papildo, tuo pačiu dar kartą patvirtindamos sisteminio mąstymo teisingumą.

6. TAIKOMOJI SISTEMOTYRA IR JOS SUDĖTINĖS DALYS. SISTEMOTECHNIKA.

Kaip jau buvo minėta šios metodinės priemonės trečiajame skyrelyje, šiuo metu egzistuoja trys sistemų teorijos ir sisteminės metodologijos taikymų praktikoje kryptys:

- 1) **sistemotechnika**, apimanti sudėtingų techninių sistemų projektavimą, jų bandomųjų egzempliorių gamybą, išbandymą ir įvertinimą, bei tos metodologijos taikymą orgprojektavime;
- 2) **orgprojektavimas**, apimantis sudėtingų organizacijų ir jų valdymo sistemų konstravimą, diegimą, išbandymą, vertinimą ir tobulinimą;
- 3) **operacijų tyrimas**, apimantis sudėtingų procesų, vykstančių organizacijose ir kitose sudėtingose sistemose, optimizavimą ir jų atlikimo optimalių technologijų projektavimą bei diegimą.

Istoriškai taip susiklostė, kad trečioji kryptis pradėjo audringai vystytis šiek tiek anksčiau už pirmąsias dvi (dar Antrojo pasaulinio karo metu bei tuoj po jo) ir tapo savarankišku taikomuoju mokslu ir praktine veikla dar tada, kada sistemotyros mokslas dar nebuvo pakankamai susiformavęs. Operacijų tyrimo specialistai pradėjo burtis į mokslines bendrijas ir leisti specialius mokslinius žurnalus taip pat šiek tiek anksčiau, negu sistemotyros specialistai. Ir aukštosiose mokyklose operacijų tyrimo kursai buvo pradėti dėstyti taip pat anksčiau, negu sistemotyros kursai. Dėl šių priežasčių ir dabar operacijų tyrimas egzistuoja lyg ir atskirai nuo sistemotyros, nors kai kurie mokslininkai (pvz. R.L.Akofas, N.N.Moisiejevas) linkę teigti, jog sistemotyra yra operacijų tyrimo vystymosi ir jo taikymų sferos plėtimosi rezultatas, nauja operacijų tyrimo modifikacija, gavusi platesnį sistemų tyrimo pavadinimą.

Jei nagrinėtume tik taikomąją ir metodologinę sistemotyros dalį, tai turėtume pripažinti, jog R.L.Akofas (R.L.Ackoff), N.N.Moisiejevas (N.N.Moiseev) ir kiti jų bendraminčiai yra teisūs, nes ir sistemotechnika, ir orgprojektavimas, ir operacijų tyrimas remiasi ta pačia sistetine metodologija (kurios pagrindiniai bruožai ir principai aprašyti šios metodinės priemonės penktajame skyrelyje); ir operacijų tyrime, ir sistemotechniniame bei organizacijų sisteminame projektavime dalyvauja kompleksiniai įvairių specialistų kolektyvai, vadovaujami dženeralistų; ir operacijų tyrime, ir sistemotechnikoje, ir orgprojektavime plačiai taikomi matematinio modeliavimo, naudojantis kompiuterine technika, metodai; ir operacijų tyrime, ir sistemotechnikoje, ir orgprojektavime ieškoma sprendimų, kurie būtų optimalūs arba pakankamai artimi jiems; ir operacijų tyrime, ir sistemotechnikoje, ir orgprojektavime didelis dėmesys skiriamas „žmogiškajam faktoriui“; kuriant sudėtingas organizacijas ir jų valdymo sistemas, tenka naudotis ir orgprojektavimo, ir sistemotechnikos, ir operacijų tyrimo pasiekimais, tenka bendradarbiauti ir orgprojektavimo, ir sistemotechnikos, ir operacijų tyrimo specialistams, net nebeskiriant, kuris iš jų kuriai grupei priklauso. Galima sakyti, kad tokiais atvejais tie specialistai užsiima beveik tuo pačiu ir

skiriasi tik jų dėmesio bei veiklos akcentai: **sistemotechnikos ir orgprojektavimo specialistų dėmesio centre yra sistemos struktūros ir jos sudėties konstravimas, optimizavimas ir diegimas, operacijų tyrimo specialistų dėmesio centre – sistemoje vykstančių procesų, sistemos funkcionavimo technologijų optimizavimas, projektavimas ir diegimas.** Kuriant sistemą, ir viena, ir kita yra reikalinga ir beveik neatsiejama, nes šie aspektai (struktūrinis – objektinis ir funkcinis – technologinis) vienas kitą papildo: iš vienos pusės negalima suprojektuoti sistemos struktūros ir jos elementų, neatsižvelgiant į tai, kaip sistema privalės funkcionuoti; iš kitos pusės, neįmanoma suprojektuoti technologinių procesų ir optimizuoti juos, neatsižvelgiant į sistemos struktūrą ir sudėtį, į tų procesų vykdytojų bei tuose procesuose dalyvaujančių objektų savybes, t.y. kartu neprojektuojant sistemos struktūros ir sudėties. Todėl, vadovaujantis sisteminiu mąstymu, galima teigti, jog orgprojektavimas, sistemotechnika ir operacijų tyrimas yra neatsiejamos sudėtingų sistemų sisteminio projektavimo dalys, sudarančios vieningą visumą.

Tačiau **tapatinti visą sistemotyra vien tik su operacijų tyrimu**, kaip tai bando daryti minėtieji mokslininkai, vis dėlto nederėtų, nes, pavyzdžiui, **sistemų teorijos tyrimų sfera ir to mokslo turinys yra žymiai platesni ir abstraktesni, negu operacijų tyrimo sfera ir turinys.** Be to, ir visos trys taikomosios sistemotyros sudėtinės dalys (sistemotechnika, orgprojektavimas ir operacijų tyrimas), nors ir labai panašios bei beveik neatsiejamos, nėra tapačios.

Operacijų tyrimą tikslinga būtų laikyti sudėtine taikomosios sistemotyros dalimi, kaip tai padaryta šio skyrelio pradžioje, tačiau, kadangi operacijų tyrimas, sistemotyrų formuojantis, jau egzistavo kaip savarankiškas mokslas ir praktinė veikla, tai istoriškai taip susiklostė, jog tradicinei taikomajai sistemotyrų dabar priskiriama tik sistemotechnika ir orgprojektavimas. Ryšius gi tarp sistemotyros ir operacijų tyrimo bei kitų giminingų sistemotyrų disciplinų atspindi mokslo, technikos ir praktinės veiklos kryptis, charakterizuojama bendra sistetine metodologija ir pavadinta **sisteminio sąjūdžio** vardu. Sisteminiam sąjūdžiui priklauso ne tik sistemotyra ir operacijų tyrimas, bet ir **kibernetika siaurąja prasme** (mokslas apie bendruosius valdymo „mechanizmus“ bei jų funkcionavimo techninėse sistemose, gyvojoje gamtoje ir visuomenėje dėsningumus), **informatika arba kibernetika plačiąja prasme** (mokslas apie informacinius procesus, informacijos apdorojimą, saugojimą bei panaudojimą, apie informacinių procesų algoritmus bei technologijas, apie informacinę techniką bei jos panaudojimą), **ergonomika ir inžinerinė psichologija** (mokslai, nagrinėjantys žmogaus fiziologines ir psichologines savybes tuo tikslu, kad į jas būtų atsižvelgta, projektuojant techninius įrengimus ir kuriant žmogaus techninę aplinką, idant būtų sukurta palankiausia – komfortinė – efektyviai žmonių veiklai techninė aplinka), **grupinė dinamika** (mokslas, nagrinėjantis tai, kaip keičiasi bei turėtų keistis aktyviųjų elementų organizavimo formos, t.y. organizacijų struktūros, jų centralizavimo bei decentralizavimo lygis, keičiantis organizacijų tikslams ir funkcijoms), **bionika** (mokslas apie biologinių ir techninių „mechanizmų“ bei jų funkcionavimo analogijas ir apie tų analogijų panaudojimą naujos technikos kūrime bei gyvosios gamtos reiškinių nagrinėjime), **ekologija** (mokslas apie žmogaus ir jo techninės veiklos poveikį gamtai, o taip pat praktinė veikla, nukreipta į gamtos ir paties žmogaus apsaugojimą nuo kenksmingos gamtai jo veiklos ir tos veiklos pasekmių) ir eilė kitų naujų mokslų, atsirandančių, sąveikaujant bei jungiantis seniau susikūrusiems ir tapusiems jau „klasikiniais“, mokslams. **Sisteminiam sąjūdžiui priklausančių mokslų atstovai gana dažnai įeina į kompleksinių kolektyvų, skirtų sudėtingų sistemų tyrimui bei projektavimui, sudėtį kartu su klasikiniiais inžinieriais, fizikais, chemikais, matematikais, ekonomistais, sociologais, medikais bei psichologais, didindami projektų ir tyrimų visapusiškumą.**

Istorinės tiesos dėlei reikia pažymėti, jog ir paties **sisteminio sąjūdžio formavimuisi pradžią davė ne šiuolaikinė sistemotyra, o kibernetika ir kompiuterinės technikos atsiradimas, jų sukeltas „kibernetinis bumas“ moksle ir technikoje, kurio pasekoje**

pradėjo intensyviai formuotis eilė naujų mokslo ir technikos krypčių, tarp jų – ir sistemotyra.

Šiame skyrelyje bus trumpai apibūdinta pirmoji taikomosios sistemotyros sudėtinė dalis – sistemotechnika, nes žinios apie sudėtingų techninių sistemų projektavimą bus naudingos SIS kūrėjams, kadangi į SIS sudėtį įeina gana sudėtinga techninė sistema – SIS techninė įranga, be to **daug iš to, kas būdinga sudėtingų techninių sistemų projektavimui, būdinga ir organizacijų bei jų valdymo sistemų projektavimui**. Kitos dvi taikomosios sistemotyros vystymosi kryptys SIS kūrėjams yra dar aktualesnės ir bus apibūdintos kituose šios metodinės priemonės skyreliuose.

Sistemotechnika, kaip taikomasis sudėtingų techninių sistemų projektavimo mokslas ir praktinė veikla, pradėjo formuotis apie XX amžiaus vidurį: šiek tiek vėliau, negu operacijų tyrimas, tačiau anksčiau, negu orgprojektavimas. **Sistemotechnikos formavimąsi skatino dvi grupės faktorių: iš vienos pusės, reikiamybė projektuoti sudėtingas technines sistemas karinėje technikoje bei pramonėje, atominėje energetikoje, kosminėje technikoje ir t.t.; iš kitos pusės, atsiradusios naujos galimybės dėl kompiuterinės technikos atsiradimo, skaičiavimo, matematikos, matematinio modeliavimo ir optimizavimo metodų vystymosi.** Žmonės, susidūrę su sudėtingų techninių sistemų projektavimu, kuriose jungiančiųjų elementų lyginamasis svoris ir reikšmė yra palyginami ar net didesni negu jungiamųjų elementų lyginamasis svoris ir reikšmė, įsitikino, jog tokių sistemų projektavimą tikslinga vykdyti kaip dviejų vienas kitą papildančių projektavimo procesų, besiskiriančių savo aspektais, sintezę. Vieną iš šių procesų sudaro sistemos funkcijų ir jos struktūros projektavimas, vadinamas struktūriniu projektavimu, arba makroprojektavimu, antrąjį – sistemos sudėties, jos fizinių komponentų projektavimas, vadinamas „klasikiniu projektavimu“, arba mikroprojektavimu.

Makroprojektavimo metu vadovaujamosi teleologine metodologija: žinodami tikslus, kurių siekia užsakovai, ir projektuojamos techninės sistemos paskirtį, makroprojektuotojai, **naudodamiesi funkciniu projektavimo metodu, projektuoja funkcinę būsimos sistemos struktūrą, tos sistemos veikimo principinę schemą.** Mikroprojektavimo metu vadovaujamosi terminaline metodologija: naudojantis objektiniu metodu, projektuojamos sistemos sudėtinės dalys, kurios, sąveikaudamos kuriamoje techninėje sistemoje, realizuotų makroprojektavimo metu suprojektuotą sistemos funkcionavimą. Mikroprojektavimo rezultatas – techninės sistemos detalių išoriniai modeliai, jos posistemių (mazgų ir pan.) ir visos sistemos vidinis modelis, o taip pat sistemos dalių ir visos sistemos pagaminimo proceso aprašymas, kitaip tariant, visa techninės sistemos pagaminimui reikalinga konstruktorinė ir technologinė dokumentacija. **Abu šie procesai – makroprojektavimo ir mikroprojektavimo – vyksta beveik lygiagrečiai** (makroprojektavimas prasižada šiek tiek anksčiau, o mikroprojektuotojai įsijungia į bendrą darbą šiek tiek vėliau, kai atsiranda jų darbo pradžia reikalingas makroprojektuotojų įdirbis) **ir vienas kitą papildo**, vienas kitam padeda detalizuoti projektinius sprendimus, keldami vienas kitam klausimus ir užduotis. Taip **sistemotechniniame projektavime realizuojami teleologinės ir terminalinės metodologijų bei funkcinio ir objektinio analizės ir projektavimo metodų sintezės principų** (žiūr. 32-33 psl.) **reikalavimai ir rekomendacijos, o taip pat sudėtingos sistemos kūrimo procesų palaipsniškumo ir lygiagretumo principų reikalavimai ir rekomendacijos** (žiūr. 34-35 psl.).

Šių dviejų vienas kitą papildančių projektavimo procesų skirtumai pasirodė tokie reikšmingi, jog gana greitai buvo įsitikinta, kad tokiame projektavime **klasikinių specialistų – klasikinių inžinierių – jau nebepakanka ir reikalingi dar ir kito tipo projektuotojai – sisteminiai analitikai** bei iš jų tarpo išskirti viso projekto ir projektavimo grupių vadovai – **dženeralistai**. Taip pat buvo įsitikinta, jog **nebepakanka ir klasikinės projektavimo metodologijos** (kurios blokinę schemą vaizduoja 4 pav.) **ir reikalinga žymiai sudėtingesnė metodologija, kuri vėliau buvo pavadinta sistemine** (tos metodologijos aprašymui buvo

skirtas penktasis šios metodinės priemonės skyrelis). Pasirodė taip pat, jog, **pereinant nuo klasikinio prie sisteminio projektavimo, tenka papildyti ir projektavimo priemonių, tame skaičiuje projektavimui reikalingų matematinių ir techninių mokslų žinių bei metodų, arsenalą**, nes sudėtingos techninės sistemos pasirodė esančios daugiamatės, kuriose vieno kintamųjų bei jų grupių pokyčiai priklauso nuo kitų kintamųjų bei jų grupių pokyčių (ir tos priklausomybės dažnai yra netiesinės, pasireiškiančios per daugybę grįžtamųjų ryšių, kurių kilpose gali egzistuoti kitų grįžtamųjų ryšių kilpos); tokių sistemų funkcionavimas ir sąveikavimas su aplinka pasirodė ne tik sudėtingas, bet ir dėl tam funkcionavimui ir sąveikavimui būdingų atsitiktinumų nepilnai apibrėžtas, su kai kuriais „rungtyniavimo“ tarp atskirų elementų ir paprocesių reiškiniais. Todėl sudėtingų techninių sistemų projektuotojams teko papildyti savo bazinį išsilavinimą tikimybių teorijos bei matematinės statistikos, masinio aptarnavimo sistemų teorijos, informacijos teorijos, automatų ir algoritmų teorijos, automatinio reguliavimo ir optimizavimo, techninės kibernetikos ir ergonomikos bei inžinerinės psichologijos žiniomis, susipažinti su ekologiniais reikalavimais, išmokti naudotis kompiuterine technika bei jos programine įranga, o kai kuriems iš jų – ir tokią įrangą kurti. **Pereinant nuo klasikinio prie sistemos techninio projektavimo, keičiasi ir samprata apie projektavimo proceso organizavimą, atsiranda išorinio ir vidinio projektavimo loginių etapų ir projektavimo chronologinių fazių sąvokos.**

Išorinio projektavimo metu nagrinėjama projektuojamos sistemos aplinka ir formuluojami tie reikalavimai, kuriuos turi tenkinti ta sistema, sudaroma projektavimo techninė užduotis bei išorinis projektuojamos sistemos modelis.

Išskiriami šie **išorinio projektavimo loginiai etapai**:

- 1) projektuotojams keliamo **uždavinio aplinkos nagrinėjimas**, kurio metu projektuotojai susipažįsta su tomis problemomis, kurias turi spręsti projektuojamoji techninė sistema, o taip pat su galimais tų problemų sprendimo būdais bei su tomis priemonėmis, kurios bus reikalingos, projektuojant sistemą; **pagrindinis šio etapo tikslas – paruošti projektuotojus – sistemos technikus – pageidaujamos techninės sistemos projektavimui, supažindinant juos su uždavinio aplinkos specifiškai**;
- 2) būsimų vartotojų bei projekto užsakovų **požiūrių į būsimąją sistemą, kurią reikia suprojektuoti, išsiaiškinimas**, tų požiūrių derinimas, jų hierarchijos nustatymas, projektavimo tikslų patikslinimas ir jų raštiškas formulavimas; šio etapo metu projektuotojai – **sistemos technikai, bendraudami su projekto užsakovais ir būsimais projektuojamos sistemos vartotojais, stengiasi kuo pilniau ir tiksliau išsiaiškinti jų pageidavimus, suderinti juos ir dokumentaliai užfiksuoti**;
- 3) tų **apribojimų**, į kuriuos turi atsižvelgti projektuotojai, priimdami projektinius sprendimus, **išsiaiškinimas**; šio etapo metu projektuotojai stengiasi nustatyti ir aprašyti tuos apribojimus, kurie išplaukia iš sistemos sukūrimui išskirtų resursų ir laiko, iš projektuotojų ir būsimos techninės sistemos gamintojų kvalifikacijos bei jų turimos techninės bazės, esamo mokslo ir technikos lygio, medžiagų ir pusfabrikačių, iš kurių bus gaminama sistema, kokybės, iš juridinių, sanitarinių bei ekologinių reikalavimų, iš dabartiniu metu egzistuojančių tradicijų, socialinės situacijos, estetinių pažiūrų, mados tendencijų ir pan.;
- 4) būsimos sistemos **funkcionavimo efektyvumo optimalumo rodiklių ir kriterijų**, o taip pat ir **projektinių kriterijų nustatymas**; šio etapo metu turi būti pagrįstai parinkti bendrieji (globaliniai) ir daliniai (lokalieji) efektyvumo rodikliai ir kriterijai, o taip pat projektiniai kriterijai, nustatantys kuriamos sistemos ribines charakteristikas, kurių viršijimas jau nebeduoda reikšmingo papildomo išlošio; iš globaliųjų efektyvumo rodiklių turi būti išrinktas (ar sukonstruotas) **globalusis optimalumo rodiklis**, o iš lokaliųjų efektyvumo rodiklių – **lokalieji optimalumo rodikliai** atskiriems sistemos posistemiams;

- 5) projektuojamos sistemos posistemių ir visos sistemos **matematinų modelių sudarymas ir jų analizė**; šio etapo metu bandoma matematiškai išreikšti ir išanalizuoti priklausomybes tarp projektuojamos sistemos parametrų ir efektyvumo rodiklių, tarp tos sistemos ir jos posistemių įėjimų ir išėjimų, apskaičiuoti sistemos sukūrimui reikalingas išlaidas bei įvertinti tą efektą, kurį galės duoti sukurtoji sistema;
- 6) techniškai ir ekonomiškai pagrįstos **projektavimo techninės užduoties sudarymas**, jos aptarimas ir tvirtinimas.

Išorinio projektavimo metu **pagrindinę darbų dalį sudaro duomenų rinkimas ir jų apdorojimas**. Duomenys renkami iš įvairių dokumentų bei kitų literatūrinių šaltinių, jie gaunami pokalbių su užsakovais ir būsimais vartotojais metu, projektuotojams dalyvaujant tuose darbuose, kuriuos vykdydys projektuojamoji sistema, modeliuojant jos funkcionavimą, atliekant (reikalui esant) kai kuriuos natūrinius eksperimentus, įvairius matavimus, skaičiavimus ir pan. **Pagrindinės matematinės išorinio projektavimo priemonės yra tikimybių teorijos ir matematinės statistikos metodai bei matematinis modeliavimas.**

Vidinio projektavimo metu surandami konstrukciniai sprendimai, iš kurių išplaukia vidinė sistemos sandara ir jos vidinis funkcionavimas, suprojektuojami vidiniai ryšiai, jungiantys sistemos įėjimus ir išėjimus.

Išskiriami trys **vidinio projektavimo loginiai etapai**:

- 1) **„vienetinių siūlų“ projektavimas**, kurio metu **projektuojami atskirų išorinių poveikių bei jų pokyčių iššaukiami vidiniai procesai**, pradedant nuo vieno kurio nors įėjimo ir baigiant kuriuo nors išėjimu, kai kiti įėjimai turi normalias (arba artimas normalioms) reikšmes ir išlieka pastovūs;
- 2) **„pilno apkrovimo“ projektavimas**, kurio metu **projektuojamas sistemos funkcionavimas, esant pilnam jos įėjimų apkrovimui ir veikiant visiems jos išėjimams**, t.y. sąveikaujant visiems tiems atskiriems paprocesiams, kurie buvo suprojektuoti „vienetinių siūlų“ etape;
- 3) **„paprocesų rungtyniavimo“ projektavimas**, kurio metu **nagrinėjamos** tokios kartais susidarančios **konfliktinės situacijos**, kuomet kai kurie du ar daugiau paprocesių, vykstančių pagal jau suprojektuotus „vienetinio siūlo“ algoritmus, pilno apkrovimo metu susiduria, „konfliktuodami“ vienas su kitu (pavyzdžiui, kai vieno ir to paties įrengimo prireikia dviems paprocesiams vienu ir tuo pačiu metu, ir tampa neaišku, kuriam įrengimui atiduoti pirmenybę, arba kai koks nors paprocesis dėl savo vėlavimo atsilieka nuo bendro sistemos funkcionavimo ritmo ir sukuria „konfliktines“ situacijas“), ir **projektuojami tokių „konfliktų“ sprendimo algoritmai**.

Pirmajame vidinio projektavimo loginiame etape projektuojamas sudėtingas procesas dekomponuojamas į „vienetinio siūlo“ paprocesius ir sudaromi tų paprocesių projektai, antrajame – vyksta „vienetinio siūlo“ paprocesių sintezė į vieną bendrą procesą ir sudaromas to proceso projektas, trečiajame – analizuojamas suprojektuotos sistemos funkcionavimas ir derinamas atskirų posistemių sąveikavimas, projektuojant paprocesių „rungtynes“ ir „konfliktinių“ situacijų sprendimą.

Pirmojo etapo rezultatai yra trijų rūšių blokinės schemas: 1) **funkcinės blokinės schemas**, vaizduojančios „vienetinio siūlo“ paprocesių sudėtį ir struktūrą vykdomų funkcijų (veiksmų) lygyje – tų paprocesių grafiniai modeliai; 2) projektuojamos sistemos funkcionavime dalyvaujančių **„įrengimų“** (t.y. techninių įrengimų, o kartais ir tuos įrenginius aptarnaujančių bei su jais sąveikaujančių žmonių) **sąveikavimo blokinės schemas**, vaizduojančios „įrengimų“, dalyvaujančių „vienetinio siūlo“ paprocesiuose, grandines, jungiančias sistemos įėjimus su jos išėjimais; 3) **komponenčių blokinės schemas**, vaizduojančios kiekvieno įrengimo bei jų blokų sudėtinės dalis (komponentes) ir jų sąveikavimą įrengimų bei jų blokų funkcionavimo metu. Tokios schemas sudaromos, pavyzdžiui, projektuojant didelės įmonės technologines linijas.

Antrojo etapo rezultatai grafiškai vaizduojami trimis schemomis: 1) „geografinė“ **blokinė schema**, vaizduojančia „vienetinio siūlo“ technologinių linijų „geografinį išsidėstymą“; 2) **aparaturės fizinio išdėstymo schema**, vaizduojančia „geografinę“ aparaturės išdėstymą ir jos fizinius ryšius projektuojamoje sistemoje; 3) **jungtinė funkcinė schema**, vaizduojančia projektuojamoje sistemoje vykstančius procesus jos posistemų bei blokų vykdomų funkcijų lygyje; ši schema gaunama, jungiant „vienetinio siūlo“ funkcinės blokines schemas į vieningą visumą.

Trečiojo etapo rezultatai: 1) optimalios (ar bent jau patenkinamos) **sprendimų „konfliktingose“ situacijose priėmimo strategijos ir jų realizavimo algoritmai**; 2) **suderinta (pakoreguota) jungtinė funkcinė schema**, kurioje vaizduojamos ne tik pagrindinės projektuojamos sistemos blokų („įrengimų“) funkcijos, bet ir pagalbinės funkcijos, reikalingos pagrindinių paprocesų suderinimui, „konfliktų“ sprendimui, disfunkcijų neutralizavimui.

Vidinio projektavimo loginiuose etapuose kaip pagrindiniai **matematiniai instrumentai** naudojami: sisteminės **matematinės logikos** priemonės, reikalingos formalizuotam technologinių procesų aprašymui bei tų procesų algoritmizavimui; **masinio aptarnavimo procesų ir sistemų teorija**; **grupinė dinamika**; **matematinis** (tame skaičiuje ir imitacinis) **modeliavimas** bei **eksperimentai su matematiniais modeliais**; **lošimų teorija**.

Tais atvejais, kai projektinių sprendimų išbandymo metu paaiškėja, jog matematinis modeliavimas ir eksperimentai su matematiniais modeliais nepajėgūs atsakyti į kai kuriuos projektuotojų klausimus bei išsklaidyti kai kurias projektuotojų abejones, imamas **natūrinių eksperimentų**, tačiau tokių atvejų nebūna taip jau daug. Žymiai daugiau natūrinių eksperimentų vykdoma darbinio projektavimo metu, ypač po to, kai pagaminamas sistemos bandomasis egzempliorius.

Vidinio projektavimo loginiais etapais užsibaigia techninės sistemos **makroprojektavimas** ir gaunamas tos sistemos **techninis projektas**, aprašantis projektuojamos sistemos funkcinę struktūrą ir jos veikimo principus.

Lygiagrečiai, tačiau šiek tiek atsilikdamas nuo makroprojektavimo proceso, vyksta **mikroprojektavimas** – techninės sistemos sudėtinių dalių konstravimas ir jų pagaminimo technologinių procesų projektavimas, kitaip tariant, techninės sistemos **darbinio projekto kūrimas**. Žiūrint į 6 pav., galėjo susidaryti įspūdis, jog darbinis projektavimas gali prasidėti, tik visiškai užbaigus sistemos techninį projektą, tačiau tikrovėje taip nėra: sistemos sudėtinės dalys pradedamos konstruoti tuoj pat, kai techninio projektavimo metu paaiškėja, jog jos sistemoje bus tikrai reikalingos, ir išaiškėja jų vieta ir vaidmuo (jų funkcijos) sistemoje. Neretai sistemos sudėtinių dalių darbinio projektavimo metu išaiškėja ir kai kurie nauji svarbūs momentai, į kuriuos būtina atsižvelgti jau sistemos techninio projektavimo metu. Tuos grįžtamuosius ryšius vaizduoja iteraciniai procesai IP, vykstantys tarp 6 pav. blokų, priklausančių sistemos techniniam ir darbiniam projektavimui. Iš to išplaukia ne tik abiejų šių procesų lygiagretumo galimybė, bet ir tikslingumas. Be to, toks lygiagretumas leidžia sumažinti bendrąją projektavimo trukmę.

Pats mikroprojektavimas taip pat gali būti suskaidytas į keletą lygiagrečiai vykstančių procesų, kaip tai (projektavimo trukmės mažinimo sumetimais) dažniausiai ir daroma, tuo labiau, kad dėl techninės sistemos sudėtinių dalių specifikos jas konstruoja skirtingos specialistų grupės, turinčios skirtingą pasiruošimą ir skirtingą patirtį.

Dažniausiai išskiriamos tokios **mikroprojektuotojų grupės**:

- 1) **įėjimo ir išėjimo įrengimų (periferinės įrangos) projektuotojai**, kurių tarpe išskiriami smulkesni pogrupiai, užsiimančys **medžiagininiais, energetiniais ir informaciniais** įėjimais bei išėjimais, nes kiekviename iš šių pogrupių yra sava specifika, reikalinga sava specializacija;
- 2) **medžiagų, energijos ir informacijos perdavimo ir apdorojimo įrengimų bei iš tų įrengimų sudarytų technologinių linijų projektuotojai**, kurie pagal savo specializaciją taip pat skirstomi į pogrupius, sudarant vieną ar kelias hierarchines

struktūras, pavaldžias dženeralistams;

- 3) **informacinių, energetinių ir medžiaginių ryšių sistemos ir jos aparatūros projektuotojai**, kurie taip pat skirstomi į pogrupius pagal jų specializacijas;
- 4) **refleksinio valdymo (automatinio reguliavimo) aparatūros ir refleksinio valdymo posistemių ir sistemos projektuotojai**;
- 5) **loginio (t.y. programinio) valdymo aparatūros ir loginio valdymo posistemių ir sistemos projektuotojai**;

Mikroprojektuotojai savo darbe pastoviai bendradarbiauja su makroprojektuotojais, naudodamiesi makroprojektavimo rezultatais ir teikdami makroprojektuotojams informaciją apie mikroprojektavimo metu projektuojamų bei parenkamų (perkamų) įrengimų specifines savybes. Gana dažnai organizuojamos **kompleksinės grupės**, sudarytos iš mikroprojektuotojų ir makroprojektuotojų, kurioms vadovauja makroprojektuotojai – sisteminiai analitikai bei dženeralistai.

Mikroprojektuotojams, dalyvaujantiems sistemotechniniame projektavime, šalia savo specialybės inžinerinių žinių bei įgūdžių reikia ir kai kurių papildomų „sisteminių žinių“. Pavyzdžiui, periferinės įrangos projektuotojams reikia žinoti inžinerinės psichologijos reikalavimus, refleksinio valdymo įrangos projektuotojams – automatinio reguliavimo teoriją, loginio valdymo įrangos projektuotojams – kompiuterinę techniką ir jos programinę įrangą, automatų analizės ir sintezės teoriją ir technologiją.

Organizuojant sistemotechninį sudėtingos techninės sistemos kūrimo procesą, išskiriami ne tik loginiai etapai, bet ir keletas **chronologinių to proceso fazių**, koreliuojančių su loginiais etapais, tačiau nesutampančių su jais, nes skiriasi jų **išskyrimo principai**: loginiai etapai išskiriami dekomponuojant sudėtingos techninės sistemos kūrimo procesą į atskirus darbus **pagal jų paskirtį ir specifiką**, o chronologinės fazės išskiriamos grupuojant atskirus darbus **pagal jų atlikimą laiko tėkmėje**. Chronologinių fazių, dalinančių sudėtingos sistemos projektavimo ir diegimo laiką į specifinius intervalus, metu gali vykti ir vyksta ne tik makroprojektavimo, bet ir mikroprojektavimo atitinkami loginiai etapai.

Pirmoji fazė – darbų pradžios fazė. Jos metu, atliekant išorinį makroprojektavimą, formuluojama projektavimo techninė užduotis, o vykdant vidinį makroprojektavimą, formuluojamos naujos sistemos sukūrimo problemų sprendimo galimų variantų idėjos – galimi principiniai projektiniai sprendimai. Makroprojektuotojų grupės dydis pradinėje fazėje, nors ir priklauso nuo projektuojamos sistemos dydžio ir sudėtingumo, neturi būti didelis: pakanka kelių, turinčių sistemotechninio projektavimo patirtį, sisteminių analitikų.

Pirmosios fazės trukmė labai priklauso nuo projektuojamos sistemos naujumo, jos sudėtingumo ir dydžio, todėl negali būti griežtai reglamentuojama: pradinė fazė tęsiasi tol, kol nebūna kokybiškai įvykdytos visos užduotys, įtrauktos į tos fazės **užduočių standartinį sąrašą**, ir nebūna gauti šioje fazėje **numatyti rezultatai**.^{*} Fazės pabaigoje projektuotojai pateikia **ataskaitą raštu**, kurią po to **aptaria** su užsakovais ir būsimų vartotojų atstovais, reikalui esant, papildomai vykdo **nestandartines užduotis**, papildo ataskaitą ir pateikia užsakovui tvirtinimui. Tik patvirtinus ataskaitą, fazė laikoma užbaigta, ir tuomet pereinama prie antros fazės. **Ataskaitoje turi būti tokie skyriai**: 1) problemų, kurias reikia spręsti, formuluotės (pirmoji tų formuluočių redakcija, kuri vėliau bus tikslinama ir papildoma); 2) siūlomi tų problemų sprendimo variantai ir trumpa lyginamoji jų analizė; 3) siūlymai, kiek ir kokių specialistų reikėtų, vykdant numatomus projektavimo darbus, kokiuose etapuose bei chronologinėse fazėse tie specialistai bus reikalingi; 4) sistemos sukūrimo proceso trukmės ir tam reikalingų lėšų (išlaidų) artutinis įvertinimas; 5) efekto, kurį galės duoti projektuojamos sistemos sukūrimas, artutinis įvertinimas.

Antroji fazė – darbų organizavimo fazė. Jos uždaviniai: 1) pilno darbų (projektavimo ir projekto realizavimo) plano sudarymas; 2) probleminių ir darbo grupių komplektavimas (taip vadinamos R sistemos, skirtos projektuojamos sistemos S sukūrimui,

^{*} Užduočių standartinių sąrašų pavyzdžiai pateikti šios metodinės priemonės vienuoliktajame skyrelyje, skirtame CASE metodo aprašymui. Ten pateikti ir laukiamų rezultatų sąrašai.

organizavimas**); 3) išorinio ir vidinio projektavimo tęsimas, analizuojant galimus projekto variantus ir ieškant tinkamiausio.

Išorinio projektavimo metu tikslinamos spęstinių problemų formuluotės, sudaromi reikalingi matematiniai modeliai ir planuojami eksperimentai, skirti tų modelių patikrinimui. Vidinio projektavimo metu pradedamas „vienetinio siūlo“ projektų kūrimas, jų jungimas į „pilno apkrovimo“ projektą, o taip pat galimų „konfliktinių“ situacijų analizė bei „konfliktų“ sprendimo procesų algoritmizavimas. Išorinio ir vidinio makroprojektavimo eigoje, lyginant išorinio projektavimo metu suformuluotus reikalavimus su vidinio projektavimo metu gautais rezultatais, ieškoma tinkamiausių projektinių sprendimų. Antroji fazė (kaip ir pirmoji) užbaigiama ataskaitos sudarymu, jos aptarimu, koregavimu ir tvirtinimu. Kartu tvirtinami darbų planai tolimesnėms fazėms (tame tarpe ir detalus trečiosios chronologinės fazės darbų planas).

Trečioji fazė – detalaus techninio (funkcinio) projekto sudarymo fazė. Jos metu baigiami išorinio ir vidinio makroprojektavimo loginiai etapai, laboratoriniais (stendiniais) eksperimentais išbandomos pagrindinės techninio projekto idėjos ir pradedamas sistemos sudėtinių dalių mikroprojektavimas. Ataskaitos aptarimo ir tvirtinimo metu aptariamas ir tvirtinamas kuriamos sistemos techninis projektas bei darbinio projekto sudarymo detalus planas.

Ketvirtoji fazė – darbinio projekto sudarymo fazė. Jos metu baigiamas sistemos sudėtinių dalių (įėjimų, išėjimų, vidinių įrenginių ir jų ryšių, refleksinio ir programinio (loginio) valdymo įrengimų) mikroprojektavimas ir, naudojantis makroprojektavimo rezultatais, sudaromas darbinis visos sistemos projektas. To projekto dalys ir visas projektas testuojamas, naudojantis matematiniais instrumentais bei įvairiais eksperimentais. Sudaromas bandomojo egzemplioriaus (laboratorinio ar pramoninio, o kartais ir bandomosios serijos) pagaminimo planas, kuris tvirtinamas kartu su projektuotojų ataskaita už šioje fazėje padarytus darbus.

Penktoji fazė – bandomojo egzemplioriaus pagaminimo fazė. Jos metu, kaip rodo fazės pavadinimas, naudojantis ketvirtoje fazėje sudarytu sistemos darbinio projektu, pagaminamas bandomasis suprojektuotos sistemos egzempliorius. Jei bandomasis egzempliorius pagaminamas kaip laboratorinis, tai jis gaminamas viename egzemplioriuje, o jei bandomasis egzempliorius gaminamas kaip pramoninis, naudojantis suprojektuota gamybos proceso technologine dokumentacija, tai dažnai gaminamas ne vienas egzempliorius, o bandomoji serija. Gaminant pramoninį bandomąjį egzempliorių (ar jų seriją), tikrinamas ir tobulinamas technologinis projektas ir jam priklausanti technologinė dokumentacija. Ši fazė užsibaigia bandomojo egzemplioriaus eksploatavimo ir išbandymo instrukcijų bei išbandymų plano sudarymu, kuris tvirtinamas kartu su ataskaita už penktojoje fazėje padarytus darbus.

Šeštoji fazė – bandomojo egzemplioriaus išbandymo, jo suregulavimo, įvertinimo ir projektinės dokumentacijos koregavimo fazė. Jos metu gali būti atliekami įvairūs bandomojo egzemplioriaus išbandymai: laboratoriniai, pusiau pramoniniai išbandymai pas vartotoją, pramoniniai išbandymai ir t.t. Išbandymų metu įvertinamas suprojektuotos techninės sistemos atitikimas projektavimo techninės užduoties reikalavimams, jos efektyvumas, patikimumas, patogumas vartotojams ir pan. Išbandymų metu nustatomos sukurtos techninės sistemos silpnosios vietos ir privalumai, reikalui esant, bandomasis egzempliorius patobulinamas ir sureguliuojamas, o po to pristatomas specialiai priėmimo komisijai, sudarytai iš užsakovo ir projektuotojų bei gamintojų atstovų, vadovaujamai užsakovo, kuris gali pakviesti į tą komisiją ir būsimųjų vartotojų atstovus. Priėmimo komisija, pravedusi kontrolinius išbandymus, arba priima bandomąjį egzempliorių kartu su visa projektine ir eksploatacine dokumentacija, arba nusprendžia, kad bandomasis egzempliorius ir sistemos projektas nepakankamai tobuli, sustatydama atitinkamą aktą apie rastus trūkumus ir reikalingus patobulinimus ir paskirdama naują priėmimo datą, iki kurios turi būti pašalinti

** Žiūrėti dvilyktąjį šios metodinės priemonės skyrelį „SIS kūrimo darbų organizavimas“.

nurodyti trūkumai bei padaryti reikalingi patobulinimai.

Jei, sudarant projektavimo sutartį, buvo numatyta naujai suprojektuotos techninės sistemos **autorinė priežiūra**, tai atsiranda dar ir **septintoji chronologinė fazė** – tam tikras (dažnai vadinamas **garantiniu**) laikotarpis, kurio metu projektuotojai stebi suprojektuotus ir pagamintus techninės sistemos egzempliorius, tiria jų kokybę ir šalina pastebėtus trūkumus bei įveda reikalingus patobulinimus.

Kadangi sudėtingos sistemos sukūrimo procesas yra gana ilgas bei sudėtingas ir jame dalyvauja įvairios projektuotojų bei gamybininkų grupės, tai vadovavimas tokiam procesui – taip pat sudėtingas darbas. Tam darbui palengvinti naudojamosi tokia organizavimo priemonė kaip **tinkliniai grafikai**. Jie naudojami kaip planavimo ir kontrolės bei operatyvaus valdymo instrumentas.

Kadangi sudėtingų techninių sistemų projektavimo metu tenka atlikti gana daug ir gana sudėtingų skaičiavimų, o taip pat įvairių eksperimentų su matematiniais modeliais, tai sistemotechninis projektavimas neišsivaizduojamas be kompiuterinės technikos panaudojimo. Pradžioje kompiuterinė technika ir jos programinė įranga buvo taikomi epizodiškai, tik paskaičiavimams ir modeliavimui, tačiau paskutiniaisiais dešimtmečiais, atsiradus personaliniams kompiuteriams ir projektuotojo dialogo su kompiuteriu režimui, vis plačiau pradedamos kurti ir naudoti **automatizuoto projektavimo sistemos (APS)**, automatizuojančios ne tik sudėtingus skaičiavimus bei modeliavimą, bet ir projektavimo metu gaunamos informacijos kaupimą ir panaudojimą, įvairios projekcinės (tame skaičiuje ir grafinės) dokumentacijos gaminimą, o taip pat ir viso projektavimo proceso valdymą. Kiek vėliau automatizuoto projektavimo sistemos buvo pradėtos taikyti ne tik techninių sistemų, bet ir organizacijų bei jų valdymo sistemų projektavime. Su viena iš tokių, kuri buvo sukurta informacinių sistemų organizacijose projektavimo automatizavimui ir pavadinta CASE („Computer – Aided Systems Engineering“), mes susipažinsime vienuoliktajame šios metodinės priemonės skyrelyje.

Skyrelio apie sistemotechniką pabaigoje tiems skaitytojams, kurie norėtų sužinoti apie sistemotechniką žymiai daugiau, norėtusi parekomenduoti jau minėtąją Gudo ir Makolo knygą „Sistemotechnika“. Joje skaitytojas ras ne tik žymiai detalesnį visų projektavimo etapų, fazių, užduočių, jų vykdymo metodų bei rezultatų ir ataskaitų aprašymą, bet ir daug konkrečių pavyzdžių, vaizdžiai ir tiksliai iliustruojančių sistemotechninio projektavimo ypatumus. Be to, šioje knygoje skaitytojas ras ir gana vykusiai įpintus į pagrindinį tekstą skyrius, kuriuose glaustai, bet labai suprantamai išdėstytos tos matematinės žinios, kurios reikalingos sudėtingų techninių sistemų projektuotojams, kartu parodant matematinio aparato praktinį panaudojimą projektavimo metu išskylančių problemų sprendime. Knygoje, turinčioje daugiau kaip tris šimtus puslapių ir susidedančioje iš šešių dalių, 31 skyriaus ir 152 paragrafų, skaitytojas ras skyrius bei paragrafus, skirtus tikimybių teorijai ir matematinei statistikai, masinio aptarnavimo teorijai, matematiniam modeliavimui, eksperimentų planavimui, grupinei dinamikai, lošimų teorijai, automatinio reguliavimo teorijai, inžinerinei psichologijai, tiesiniam programavimui, informacijos teorijai, sisteminei logikai ir, aišku, kompiuterinei technikai. **Inžinierius, perskaitęs šiuos matematikos kurso skyrius, išdėstytus jam suprantama kalba ir įpintus į inžinerinį kontekstą, tikrai pajus matematikos ir jos metodų reikšmę ir naudą praktiniame savo darbe ir taps bent jau psichologiškai pasiruošęs naudotis matematiniu aparatu ir gilinti savo matematines žinias.** Gudo ir Makolo knygoje skaitytojas taip pat ras ir metodų, naudojamų kompleksinių kolektyvų veikloje ir skatinančių jų darbo efektyvumą, aprašymą (pavyzdžiui, tokių kaip „smegenų atakos“, „atakuojančios programos“ ir pan. metodai). Ir visa tai, apie ką čia buvo rašyta, išdėstyta labai įdomiai, sklandžiai, lengvai skaitoma ir suprantama kalba, o pati knyga sukonstruota gana racionaliai: matyt, jos autoriai, projektuodami knygos turinį ir struktūrą, naudojosi sistemine projektavimo metodologija. Svarbu pažymėti dar vieną vertingą knygos savybę: nors knyga išleista gana seniai, jos tekstas, išskyrus ketvirtąją dalį, kurioje aprašoma skaičiavimo technika, nepaseno. Knygoje išdėstyti sistemotechnikos ir sisteminės

metodologijos pagrindai, besiremiantys turtinga mokslinių tyrimų ir sudėtingų sistemų projektavimo patirtimi, tapo savo rūšies „klasika“, kurios fundamentalumą patvirtino praėję po jos išleidimo keli dešimtmečiai. **Ši knyga naudinga ne tik sudėtingų techninių sistemų, bet ir organizacijų bei jų valdymo sistemų projektuotojams, nes (kaip tai išplaukia iš izomorfizmo principo) sudėtingų techninių sistemų ir organizacijų projektavimo procesai turi gana daug bendrų bruožų. Organizacijų projektuotojams tikrai verta pasinaudoti gana turtinga techninių sistemų projektavimo patirtimi bei neblogai nušlifuota inžinerine projektavimo kultūra, racionalių projektavimo darbų organizavimu, suskirstant visą naujos sistemos kūrimo procesą į chronologines fazes ir loginius etapus, turinčius aiškiai apibrėžtas užduotis ir siekiančius konkrečių rezultatų, už kuriuos kiekvieno etapo gale atsiskaitoma, patikslinant likusių etapų planus. Šia patirtimi ir sistemotechnine projektavimo technologija plačiai naudojasi visos automatizuotos orgprojektavimo sistemos, tame tarpe ir mūsų jau minėtoji CASE, ku nesunku įsitikinti, susipažinus su CASE metodu.**

Žinoma, projektuojant organizacines sistemas, nepakanka vien sistemotechninės technologijos, susiformavusios projektuojant ir kuriant sudėtingas technines sistemas, nes, kuriant ir tobulinant organizacijas, tenka projektuoti ir organizuoti dar ir „žmogiškąją“ organizacinės sistemos dalį, sudarytą iš aktyviųjų elementų, kurių savybės žymiai skiriasi nuo techninių elementų savybių. Neužtenka tik gerai suprojektuoti aktyviųjų elementų veiklą, reikia juos dar ir išmokyti dirbti ne blogiau, kaip numatyta projekte, ir, kas svarbiausia, reikia pasiekti, kad jie to norėtų ir aktyviai siektų, patys tobulintų savo ir savo organizacijos veiklą, prisitaikydami prie besikeičiančios aplinkos. Tai, palyginti, nesunkiai galima pasiekti tik įtraukiant aktyviuosius elementus į organizacijos tobulinimo veiklą. **Organizacijos aktyviųjų elementų dalyvavimas orgprojektavime ir sudarytų projektų diegime nėra tik pagalba specialistų projektuotojų grupei, o viena iš būtiniausiųjų orgprojektavimo sudėtinių dalių – organizacijos „žmogiškosios“ dalies projektavimas, organizavimas ir tobulinimas.**

Iš čia išplaukia pagrindinis techninių ir organizacinių sistemų kūrimo bei tobulinimo metodologijų skirtumas, kurio kai kurie vadovai bei specialistai (ypatingai tie, kurie atėjo į orgprojektavimą iš techninių sistemų projektavimo srities) iki šiol dar pakankamai nesuvokė. Šio skirtumo esmė yra ta, kad techninės sistemos atžvilgiu **žmonės** – techninės sistemos projekto užsakovai, būsimi tos sistemos vartotojai, tą sistemą aptarnaujantis personalas ir patys techninės sistemos projektuotojai – **nėra techninės sistemos sudėtinė dalis**, kai tuo tarpu organizacijos, kurios veikla yra tobulinama, kompiuterizuojant kai kuriuos procesus, atžvilgiu jos **personalas yra organizacijos dalis ir net pati svarbiausioji.**

Todėl **technines sistemas projektuoja ir kuria nebūtinai tie (ir dažniausiai ne tie žmonės), kurie tomis sistemomis naudosis**, o būtent tie, kurie specializavosi tokių sistemų projektavime ir kūrime ir todėl sugeba daryti šį darbą geriau, negu kiti. Visų kitų žmonių dalyvavimas techninių sistemų kūrimo procese **apsiriboja užsakovo ir vertintojo vaidmeniu: užsakovas projektavimo pradžioje išsako savo reikalavimus bei pageidavimus, o vertintojas – būsimasis techninės sistemos vartotojas – sistemos kūrimo pabaigoje sprendžia, ar jam reikia (ar nereikia) sukurtos sistemos.**

Kuriant ir tobulinant organizacines sistemas, taip pat reikalingos specialios projektuotojų (dženeralistų ir specialistų) grupės, mokančios daryti specialų savo darbą geriau, negu kiti, tačiau vien tik jų jau nebepakanka. Nebepakanka bendravimo su užsakovais ir vartotojais vien tik sistemos kūrimo pradžioje ir pabaigoje, nes, tobulinant organizacinę sistemą, joje turi pasikeisti ne tik techninė bei technologinė, bet ir „žmogiškoji“ jos dalis: turi pasikeisti žmonių veiklos „programos“, „užrašytos“ jų galvose. Tas „programas“ neįmanoma staigiai pakeisti, projektuotojams ar organizacijos vadovui įsakius: „Nuo šiandien galvokite taip, o ne kitaip.“ Tas „programas“ bei jų motyvaciją, skatinančią žmones elgtis taip, o ne kitaip, gali pakeisti tik patys tų galvų savininkai, o ir jie – dažniausiai neiškart, nes žmonių psichika gana inertiška. Todėl ir

reikalingas tų žmonių, kurie sudaro aktyviają organizacijos dalį, pastovus aktyvus dalyvavimas organizacijos tobulinime. Kiekvienas organizacijos darbuotojas, jei jo pareigybė laikoma reikalinga patobulintoje organizacijos struktūroje, **privalo suprojektuoti savo pareigybės veiklą** taip, kad ji būtų to darbuotojo nuomone kuo racionalesnė, o **sisteminė projektavimo grupė turi visus tokius organizacijos skyrių ir jų darbuotojų pateiktus projektus, bendraudama su jų autoriais, suderinti ir nuspręsti, ką organizacijos veikloje ir kaip būtų tikslinga kompiuterizuoti**, pavesdama ADV ir jų tinklo, o taip pat duomenų banko projektavimą bei diegimą specialioms darbo grupėms.

Iš pagrindinio skirtumo tarp techninių sistemų organizacijų nesuvokimo išplaukia ir gana paplitusi klaidinga automatizuotų organizacinių sistemų (AVS, AIS, SIS ir t.t.) koncepcija, esą AVS, AIS, SIS ir t.t. yra kažkas tokio, ką reikia sukurti ir prijungti prie veikiančios „organizacijos, kaip kokią tą organizaciją tobulinantį ir ją papildantį priedą ar apendiksą, kuris po to turi funkcionuoti lygiagrečiai su senąja organizacija. Ši klaidinga koncepcija jau suspėjo atnešti daug žalos, apie kurią plačiai rašoma septynioliktajame šios priemonės puslapyje paminėtoje antrojoje knygelėje, ir tokią koncepciją reikia kuo greičiau išgyvendinti. **AVS, AIS, SIS ir t.t. yra ne priedas, lipdomas prie organizacijos, o jos nauja būseną, į kurią reikia pervesti veikiančią organizaciją.**

Turėdami omenyje šią koncepciją, pereisime prie organizacinio projektavimo metodologijos, kuria reikėtų vadovautis, kuriant SIS, aprašymo.

7. FUNKCINIS ORGANIZACIJOS STRUKTŪROS PROJEKTAVIMO METODAS IR JO TAIKYMAS

Organizacinės sistemos projektavimas pradedamas nuo jos misijos (paskirties) ir tikslų nustatymo ir iš jų išplaukiančios sistemos struktūros projektavimo.

Žinant organizacinės sistemos struktūrą, organizacijos padalinių ir jų darbuotojų funkcijas, galima pradėti projektuoti tų funkcijų vykdymo technologiją, t.y. pradėti projektuoti technologinę normatyvinę organizacinę dokumentaciją (TNOD), tuo pačiu metu sprendžiant, kurią dalį žmonių veiklos tikslinga automatizuoti. TNOD projektavimo metu atsiranda daug naujos informacijos, reikalaujančios ir leidžiančios patobulinti organizacinės sistemos struktūrą ir ją aprašančią struktūrinę normatyvinę organizacinę dokumentaciją (SNOD). Todėl galima sakyti, jog organizacinės sistemos kūrimas pradedamas nuo jos struktūros ir SNOD projektavimo, tačiau SNOD projektavimas užbaigiamas tik tada, kai suprojektuojama ir įdiegiama TNOD, pritaikyta naujoms, organizacinės sistemos projekte, numatytoms sąlygoms. Kartais kartu tenka projektuoti ir PNOD. Taip atsitinka, pavyzdžiui, kuriant įmonių, skirtų masinei konvejerinei sudėtingos technikos (pavyzdžiui, automobilių) gamybai, projektus.

Projektuojant organizacinės sistemos struktūrą ir sudėtį bei jų patobulinimus, projektavimo pagrindą ir jį atitinkančią **projektavimo metodologiją galima pasirinkti dvejopai:**

- 1) pagrindiniais projektavimo elementais pasirinkti **veiklos funkcijas** ir projektuoti tų funkcijų ryšių hierarchinę schemą – organizacijos **funkcinę struktūrą**, iš kurios, grupuojant organizacijos tikslų dekompozicijos metu išskirtas elementarias funkcijas (operacijas) pagal jų funkcinį, objektinį, informacinį bei algoritminį giminingumą ir darbo imlumą į paketus, skirtus vienam vykdytojui (organizacijos darbuotojui ar jų pastoviai grupei, žemesnio ar aukštesnio lygio padaliniui), suprojektuoti organizacinės sistemos **vykdomąją struktūrą**, vaizduojančią ne tik pačias funkcijas ir jų ryšius, bet ir tas funkcijas vykdančius subjektus bei pavaldumo ryšius tarp jų (t.y. sudaryti **funkcijų hierarchijos diagramas – FHD – grafinį vykdomosios funkcinės struktūros atvaizdą**);
- 2) pagrindiniais projektavimo elementais pasirinkti **esybes** (konkrečius bei abstrakčius veiklos objektus ir veiklos subjektus), nustatyti (suprojektuoti) ryšius, santykius, atsirandančius tarp esybių ir jų atributų (savybių) veiklos procese, ir

tokiu būdu gauti taip vadinamąją **objektinę organizacijos struktūrą** (t.y. **organizacijos sudėties elementų ir jų ryšių schemą**), grafiškai vaizduojamą **esybių ryšių diagramomis (ERD)**.

Pirmasis iš čia aprašytų projektavimo metodų vadinamas **funkciniu**, antrasis – **objektiniu**.

Kai projektuojamoji ar tobulinamoji organizacinė sistema yra nesudėtinga, tai suprojektuoti pakankamai tobulą jos struktūrą galima naudojantis bet kuriuo iš šių metodų.

Tačiau, kai organizacija sudėtinga, kai neįmanoma susidaryti vieno žmogaus galvoje pakankamai detalų ir pilną tos organizacijos ir jos funkcionavimo koncepcinį modelį, nė vienas iš aukščiau paminėtų metodų negali garantuoti, jog, naudojantis juo, bus gautas pakankamai tobulas organizacijos projektas, nes negalima garantuoti, kad visos reikalingos veiklos funkcijos, visi reikalingi veiklos objektai ir subjektai, visi jų ryšiai, visa reikalinga informacija bus įtraukta į projektą ir suderinta, kad projektuotojai nieko, turinčio reikšmę organizacinės sistemos veiklos efektyvumui, nepraleis, o taip pat nesuprojektuos nieko bereikalingo (bereikalingų funkcijų, nereikalingų veiklos subjektų ir objektų, bereikalingų informacinių ir kitų ryšių).

Kadangi funkcinio ir objektinio metodų plusai ir trūkumai nesutampa, tai abu metodai gali vienas kitą papildyti, tuo pačiu padidindami plusų ir sumažindami minusų skaičių. Todėl sisteminė metodologija rekomenduoja abiejų šių metodų ir projektų, gaunamų, naudojantis tais metodais, sintezę, suformuluodama atitinkamą principą (žiūr. dvidešimtąjį principą šios metodinės priemonės 33 psl.).

Šiame – septintajame – skyrelyje bus aprašytas funkcinis metodas, aštuntajame – objektinis metodas, o devintajame – abiejų šių metodų sintezė.

Funkcinis projektavimo metodas gali būti **normatyvinis** ir **deskriptyvinis**.

Normatyvinio funkcinio metodo esmė sudaro organizacijos normatyvinių tikslų dekompozicija į tikslų medžius iki elementarių funkcijų (operacijų), kurios gaunamos žemutiniuose tų medžių lygiuose.

Deskriptyvinio funkcinio metodo esmė – veikiančios (analogiškos projektuojamai, tačiau jau veikiančios) organizacijos veiklos tyrimas, vykdomų organizacijoje elementarių ir sudėtingų funkcijų nustatymas bei aprašymas ir jų sujungimas į hierarchines funkcijų ryšių schemas, vaizduojančias organizacijos deskriptyvinę funkcinę struktūrą.

Kai projektuojama nauja nesudėtinga organizacinė sistema, neturinti veikiančių analogų, tai vienintelis galimas funkcinio projektavimo metodas yra normatyvinis.

Kai tobulinama nesudėtingos veikiančios organizacijos struktūra arba kai tobulinama bet kokios organizacinės sistemos elementų veiklą, nedarant radikalių tos sistemos struktūros pakeitimų, tai pakanka deskriptyvinio funkcinio metodo.

Kai projektuojamoji ar tobulinamoji organizacinė sistema yra sudėtinga ir kai tobulinimo metu reikia perprojektuoti sistemos struktūrą, tai nė vieno iš dviejų minėtų funkcinio projektavimo metodų jau nebepakanka ir, vadovaujantis sistetine projektavimo metodologija (žiūr. dvidešimt pirmąjį principą šios metodinės priemonės 33 psl.), tenka atlikti šių metodų sintezę – **normatyvinės funkcinės struktūros**, gautos, naudojantis normatyviniu funkciniu metodu, ir **deskriptyvinės funkcinės struktūros**, gautos, naudojantis deskriptyviniu funkciniu metodu, **palyginimą ir derinimą**, koreguojant normatyvinį ir deskriptyvinį funkcinį struktūrų modelius.

Tokį normatyvinio ir deskriptyvinio modelių palyginimą įvykdyti yra lengviau, kai normatyvinis funkcinis modelis prieš tai yra papildytas valdymo funkcijomis, gautomis **transformuojant normatyvinę funkcinę organizacijos struktūrą į normatyvinę vykdomąją funkcinę struktūrą**.

Normatyvinę vykdomąją organizacinės sistemos struktūrą projektuojama, **grupuojant ir jungiant normatyvinėje funkcinėje struktūroje esančias funkcijas į paketus**, skirtus vienam vykdytojui, ir tokiu būdu konstruojant **vykdomosios struktūros elementus ir posistemius – veiklos subjektus**.

Funkcijų grupavimo į paketus reikiamybė atsiranda todėl, kad elementarių funkcijų darbo imlumas dažniausiai yra toks, jog vieną operaciją nepajėgi pilnai (t.y. visą darbo laiką) apkrauti vieno elementaraus veiklos subjekto (vieno pareigūno be kompiuterio ar su juo). Norėdami suformuoti pareigūnų pakankamai pilną „**funkcinį apkrovimą**“ ir suprojektuoti **pareigybių pasus** (pareigines instrukcijas), elementarias funkcijas turime sugrupuoti į jau minėtus paketus, kiekviename iš kurių visų patekusių į vieną paketą funkcijų darbo imlumų suma neturi viršyti pareigūno darbo laiko fondo. Ji turi būti net šiek tiek mažesnė, pavyzdžiui, sudarydama apie 85 % darbo laiko fondo, nes pareigūnui reikia palikti laiko ir atsitiktiniams darbams, kurių visada atsiranda, vykdant savo pastovias funkcijas. Darbo laiko fondo dalis, kurią reikia skirti atsitiktiniams darbams, labai priklauso nuo pareigybės funkcijų specifikos: ji gali būti maža toms pareigybėms, kurių funkcijos priklauso **rutininių reguliarių griežtai reglamentuotų darbų** grupei, ir žymiai didesnė toms pareigybėms, kurių funkcijos didžiaja dalimi priklauso **kūrybinių darbų** grupei. Vertinant tą laiko dalį, kurią reikia palikti atsitiktiniams darbams, netikslinga persistengti, nes mažindami darbo laiko dalį, skiriamą pastovių funkcijų vykdymui, tampame priversti didinti pareigūnų skaičių, o kartu su tuo – darbo užmokesčio išlaidas ir organizacijos valdymo sudėtingumą, artėdami prie „antrojo informacinio barjero“ (žiūr. 29÷31 psl.). Be to, palikdami pareigūnams per daug laisvo laiko darbo metu, galime pakenkti kai kurių pareigūnų, o kartais – ir visos organizacijos organizuotumui (žiūr. 24÷26 psl.), nes ne visi pareigūnai pajėgūs racionaliai panaudoti atsitiktinai (ar net pastoviai) atsirandantį laisvą laiką ir, neturėdami ką veikti, ne tik patys nieko naudingo nedarys, bet pradės trukdyti kitiems, net to nenorėdami.

Kai pareigūnų apkrovimą ir atsitiktiniams darbams skiriamą laiko dalį sunku įvertinti matuojant ir skaičiuojant bei naudojantis specialiais ekspertinio vertinimo metodais, o tai atsitinka, kai projektuojami nauji darbai, tai gali būti organizuoti specialūs tyrimai, tačiau tai reikalauja lėšų ir laiko. Todėl tokiais atvejais dažniausiai apsiribojama apytiksliais ekspertiniais vertinimais, kurie po to tikslinami sistemos funkcionavimo eigoje.

Vykdomosios normatyvinės struktūros konstravimas pradedamas „nuo apačios“, nuo elementarių funkcijų, t.y. atvirkštine tvarka, lyginant su tuo, kaip buvo formuojama normatyvinė funkcinė struktūra, kuri buvo gauta, skaidant tikslus į potikslus, šiuos į dar smulkesnius potikslus, kol buvo gauti tokie potiksliai, kurių kiekvieno pasiekimui pakanka įvykdyti vieną elementarią funkciją.

Prieš pradedant elementarių funkcijų grupavimą į paketus, skirtus atskiroms pareigybėms, reikia gana detaliai išnagrinėti tas funkcijas, išsiaiškinant, kokių sudėtingesnių funkcijų sudėtinėmis dalimis yra nagrinėjamos elementarios funkcijos, kas bus veiklos objektais, vykdant tą ar kitą elementarią funkciją, kokių priemonių ir kokios informacijos reikės vykdytojui, kokiais algoritmais jis naudosis, kokia kiekvienos funkcijos įvykdymui reikalingo darbo trukmė, ar tikslinga tos funkcijos vykdymą pilnai ar bent dalinai automatizuoti.

Grupuojant elementarias funkcijas į paketus, **reikia stengtis, kad paketai jungtų kuo labiau giminingas funkcijas, nes tai padeda gauti racionalesnę vykdomąją struktūrą**: supaprastėja pareigūnų darbas ir informacinių srautų struktūra, sumažėja cirkuliuojančios informacijos apimtys, supaprastėja valdymo sistema.

Funkcijų giminingumas gali būti funkcinis, objektinis, informacinis ir algoritminis.

Kai kelios funkcijos, priklausydamos tai pačiai tikslų medžio šakai, yra vienos sudėtingesnės funkcijos dalys, tai tokį tas funkcijas jungiantį bendrumą vadina **funkciniu giminingumu**.

Kai kelios funkcijos turi tuos pačius, beveik tuos pačius bei teritoriniu atžvilgiu artimus veiklos objektus, o taip pat kai tų funkcijų vykdymui reikalingos tos pačios ar beveik tos pačios pagalbinės priemonės, tai tokį tas funkcijas jungiantį bendrumą vadina **objektiniu giminingumu**.

Kai kelių funkcijų vykdymui reikalinga ta pati ar beveik ta pati informacija, tai tokį tas

funkcijas jungiantį bendrumą vadina **informaciniu giminingumu**.

Kai kelios funkcijos turi vienodus ar beveik vienodus jų vykdymo algoritmus, tai tokių funkcijų jungiantį bendrumą vadina **algoritminiu giminingumu**.

Ne visada pavyksta pasiekti tai, kad funkcijos, patekusios į vieną paketą, būtų giminingos visais aspektais, tačiau bent vienu ar dviem aspektais jos turėtų būti giminingos ir charakterizuoti jų vykdytojo veiklos sritį ir specializaciją.

Paskirsčius elementarias funkcijas į paketus, gaunami **žemutinio ešelono pareigybių normatyviniai funkciniai modeliai**. Būna tik suteikti taip sukonstruotoms pareigybėms pavadinimus (identifikatorius, kodus) ir galima **pradėti normatyvini pareigybių pasų (PP) projektavimą**.

Kai projektuojamoji organizacinė sistema yra nauja ir neturi veikiančių analogų, tai normatyvinis PP projektavimo metodas yra vienintelis galimas. Tačiau tais atvejais, kai vyksta jau veikiančios sudėtingos organizacijos struktūros tobulinimas, ją perprojektuojant, arba kai projektuojama nauja sudėtinga organizacinė sistema, turinti veikiantį analogą, šalia normatyvinio PP projektavimo metodo stengiamasi naudoti dar ir deskriptyviniu metodu, nes normatyvinių ir deskripcinių projektavimo metodų sintezė, kaip teigia dvidešimt pirmasis principas (žiūr. 33-jį puslapį) leidžia sumažinti sudėtingos organizacijos projektavimo klaidų skaičių ir tokiu būdu pagerinti projekto kokybę. Deskripcinis projektavimo metodas bus aprašytas šiek tiek vėliau. Čia kol kas apsiribosiu tik pastaba apie pagrindinius šių PP projektavimo metodų skirtumus ir tų metodų sintezės tikslumą. Pirmąjį skirtumą sudaro tai, kad normatyvinis PP projektavimo metodas taikomas, transformuojant funkcinę organizacijos struktūrą į vykdomąją ir to transformavimo metu konstruojant pačias pareigybes kaip funkcijų paketų, gautų jas grupuojant, vykdymo subjektus, o deskriptyvinis PP projektavimo metodas taikomas tada, kai organizacija (ar jos analogas) jau funkcionuoja, t.y. tada, kada egzistuoja ne tik pačios pareigybės, bet ir jas užimantys pareigūnai. Antrojo skirtumo esmė ta, kad normatyvinį PP projektavimą vykdo organizacinių sistemų projektavimo specialistai, įsisavinę sisteminę orgprojektavimo metodologiją ir mokantys naudoti kompiuterinę techniką su specialia programine įranga, automatizuojančia kai kuriuos projektavimo darbus, o deskriptyvinį PP projektavimą toje organizacijoje, kurioje vyksta struktūros tobulinimas, privalo vykdyti patys pareigūnai, vadovaudamiesi savo patirtimi, savo darbo supratimu bei mokėjimu ir metodiniais nurodymais, panašiais į tuos, kurie pridėti prie šios metodinės priemonės kaip pirmasis jos priedas. Abu metodai vienas kitą papildo, nes, **naudojantis normatyviniu metodu, gaunami gana racionalūs, suderinti ir logiškai pagrįsti, tačiau gyvenimo praktika dar nepatikrinti PP projektai, o naudojantis deskriptyviniu metodu, gaunami, galbūt, ir ne tokie racionalūs, bet atitinkantys gyvenimo praktikos reikalavimus pareigybių pasai: lyginant abu PP variantus ir apmąstant pastebėtus skirtumus, galima sudaryti PP, kurie, būdami racionalesni už deskriptyvinius jų modelius, galėtų būti gana nesunkiai įdiegti organizacijos praktinėje veikloje**. Projektuojant PP, abu metodus – deskriptyvinį ir normatyvinį – tikslinga taikyti lygiagrečiai, organizuojant vieningą PP projektavimo procesą, kuriame bendrautų orgprojektavimo bei kompiuterizavimo specialistai ir organizacijos darbuotojai – atitinkamas pareigas užimantys pareigūnai.

Sudarius giminingų funkcijų paketus, skirtus atskiroms žemutiniojo ešelono pareigybėms ir pradėjus pareigybių pasų projektavimą, normatyvinis vykdomosios struktūros projektavimas tęsiamas, **jungiant į grupes giminingus funkcijų paketus** ir formuojant organizacijos žemutiniojo ešelono **padalinių normatyvinę sudėtį ir struktūrą**. Tas formavimas vykdomas, vadovaujantis **pavaldumo normatyvais**, rekomenduojančiais optimalius vienam vadovui pavaldžių pareigūnų skaičius, o taip pat jau minėtais giminingumo požymiais, jungiant į padalinio paketą tokius pareigybių paketus, kurie turi funkcinį, objektinį, informacinį ar algoritminį giminingumą. Optimalus pareigybių skaičius viename žemutinio ešelono padalinyje labai priklauso nuo pareigybių ir padalinio veiklos tipo bei jos sudėtingumo ir giminingumo laipsnio. Pavyzdžiui, jei visos į padalinį jungiamos pareigybės

turi aukštą algoritminio giminingumo laipsnį, tai tokiame padalinyje gali būti net kelios dešimtys pareigūnų, ypač, jei jungiamos į padalinį pareigybės šalia to turi dar ir objektinį ar informacinį giminingumą. Kai tenka jungti į žemutinio ešelono padalinį mažai giminingas pareigybes arba į aukštesnio ešelono padalinį mažai giminingus (ar visai negiminingus) žemesniojo ešelono padalinius, tai jungiamų vienetų skaičius paprastai būna lygus 5–11 ir vėlgi priklauso nuo padalinių veiklos ir jungiamųjų dalių giminingumo laipsnio.

Jungiant pareigybes į žemutiniojo ešelono padalinius, reikia gauti pirmojo tipo sistemas (žiūr. 5–6 psl.), turinčias savas sisteminės funkcijas ir savybes, o todėl jungiamos pareigybės turi tenkinti suderinamumo, organizavimosi bei organizuotumo ir emerđžentiškumo principų (žiūr. 24–25 psl.) reikalavimus. **Tuos reikalavimus patenkinti yra žymiai lengviau, jei jungiamos pareigybės turi funkcinį giminingumą, todėl tas giminingumo tipas ir minimas pirmuoju.** Didelę reikšmę padalinio „sistemiškumui“ turi ir objektinis giminingumas, kitaip tariant, veiklos objektų teritorinis artumas bei „kompaktiškumas“, jų priklausymas tokiai stambesnei visumai, kurią būtų galima laikyti sistema – savistoviu objektu. Todėl objektinis giminingumas ir minimas antruoju.

Jungdami pareigybes į žemutinio ešelono padalinius, normatyvinės vykdomosios struktūros projektuotojai formuoja padalinio sudėtį ir nustato ryšius tarp pareigybių, t.y. projektuoja padalinio struktūrą. Tuo pačiu metu formuojama ir nauja pareigybė – padalinio vadovo pareigybė, sudaromas normatyvinis tos pareigybės paso ir padalinio nuostatų projektas.

Sudarius žemutiniojo ešelono padalinių ir jų nuostatų normatyvinius projektus, analogišku būdu, jungiant funkcijų paketų paketus, konstruojami aukštesniojo ešelono padaliniai, sudaromi normatyviniai jų nuostatų ir jų vadovų PP projektai.

Tęsiant aukščiau aprašytąją funkcijų paketų jungimo procedūrą ir slenkant „nuo apačios į viršų“, gaunama **normatyvinė vykdomoji organizacijos struktūra**, kurioje kiekvienai funkcijai (elementariai ir sudėtingai) priskiriamas jos vykdytojas – normatyvinės struktūros elementas ar posistemis (pareigūnas ar organizacijos padalinys).

Normatyvinis vykdomosios struktūros projektavimo procesas, vykdomas „nuo apačios į viršų“, ne tik priskiria funkcinės struktūros funkcijoms jų vykdytojus, bet ir papildo normatyvinį funkcinės struktūros projektavimo procesą, vykusį „nuo viršaus į apačią“, nes vykdomosios struktūros projektuotojai papildo organizacijai reikalingų funkcijų sąrašą naujomis funkcijomis – valdymo funkcijomis, kurios galėjo nebūti gautos, dekomponuojant organizacijos tikslus į potikslus ir sudarant tikslų medžius, vaizduojančius normatyvinę funkcinę organizacijos struktūrą.

Elementarios valdymo funkcijos gaunamos dekomponuojant valdymo subjektų veiklą. **Valdymas siaurąja to žodžio prasme – tai valdymo subjekto poveikis į valdymo objektą** (žiūr. 1 pav., vaizduojantį elementarų valdymo kontūrą), siekiant, kad to objekto būsenos $\vec{Z}$ ir išėjimų $\vec{Y}$ vektorių koordinatės (žiūr. 9–10 psl.) įgytų arba išlaikytų valdymo subjektui ir organizacijai pageidaujamas reikšmes.

Valdymo subjekto poveikiai į valdymo objektą, priklausydami vieno objekto poveikių į kitą objektą grupei (žiūr. antrąjį šios mokymo priemonės skyrelį), **išskiriami į tos grupės specifinį pogrupį.** Taip daroma todėl, kad **valdymo subjekto poveikiai į valdymo objektą** valdymo procese, nors ir surišti su kai kuriomis energetinėmis ir medžiaginėmis sąnaudomis, **nėra nei energetiniai, nei medžiaginiai, o yra informaciniai:** valdymo proceso metu valdymo objekte bei jo veikloje atsirandantys pokyčiai vyksta ne valdymo subjekto, o paties valdymo objekto energetinių ir medžiaginių resursų sąskaita. Valdymo subjektas tik inicijuoja tuos pokyčius, o juos atlieka pats valdymo objektas. Todėl ne kiekvienas fizinis objektas gali būti valdymo subjektu bei valdymo objektu. **Tam, kad kokie nors fiziniai objektai galėtų būti valdymo subjektu ir valdymo objektu, jie privalo turėti specialius informacinio sąveikavimo mechanizmus – valdymo mechanizmus.** Projektuojant valdymo sistemas, t.y. projektuojant valdymo objektus ir valdymo subjektus bei jų sąveikavimą, **reikia suprojektuoti ir tuos mechanizmus bei jų funkcionavimą.**

Valdymo subjektas, siekdamas valdymo tikslų, negali apsiriboti ka tik minėtais poveikiais, t.y. **vien tik valdymu pačia siaurąja prasme („komandavimu“)**. Valdymo subjektas, visų pirma, turi tuos tikslus žinoti: arba nustatyti pats, arba sužinoti iš kitų subjektų. Todėl į **valdymo procesą platesniąja prasme** kaip pirmoji jo sudėtinė dalis – **pirmoji valdymo proceso fazė** – įeina **strateginių tikslų nustatymas – strateginio planavimo funkcijos vykdymas. Atroji valdymo proceso fazė – operatyvinio taktinio planavimo funkcijos vykdymas**, t.y. pageidautinų („planinių“) valdymo objekto funkcionavimo trajektorijų – fazinių ir išėjimų trajektorijų (žiūr. 10 psl.) – nustatymas, nes **reikia žinoti ne tik siekiamus tikslus, bet ir tikslų pasiekimo kelius**. Trečioji valdymo proceso fazė – **momentinių valdymo sprendimų priėmimas ir subjekto poveikių objektui formavimas (valdymas pačia siaurąja prasme)**. Jei nebūtų nevaldomų (tame skaičiuje ir atsitiktinių) išorinių ir vidinių poveikių į objektą, tai aukščiau išvardintų valdymo funkcijų pakaktų **valdymo poveikių ir valdymo proceso tikslų realizavimui – ketvirtajai valdymo proceso fazei**. Tačiau tikrovėje taip nėra, ir todėl tenka valdymo funkcijų sąrašą plėsti, papildant naujomis funkcijomis. Kadangi subjekto poveikio į objektą rezultatas priklauso ne tik nuo to poveikio, bet ir nuo subjektui nepavaldžios aplinkos (išorinės ir vidinės) poveikių, tai subjektas, neanalizuodamas objekto funkcionavimo, tų rezultatų pakankamai tiksliai žinoti negali. Todėl ir atsiranda **penktoji valdymo proceso fazė** – objekto funkcionavimo stebėjimai, jo būsenos $\bar{Z}$, išėjimų $\bar{Y}$ ir įėjimų $\bar{X}$ vektorių koordinačių matavimai, ką trumpai vadina objekto funkcionavimo **apskaita ir kontrole**. Nustačius faktines koordinačių reikšmes, jas svarbu palyginti su operatyvinio planavimo metu planuotomis („normatyvinėmis“), nustatyti faktinių reikšmių nuokrypius nuo „normatyvinių“, įvertinti juos (pageidautini ar nepageidautini, esminiai ar neesminiai, t.y. verti dėmesio ir signalizavimo kitiems subjektams ar neverti), nustatyti tų nukrypimų priežastis bei pasekmes. Visa tai ir sudaro **analizės funkcijos** valdymo procese esmę – **šeštosios valdymo proceso fazės esmę**. Jei pasirodo, kad minėtieji nukrypimai yra nepageidautini ir jų dydis artėja prie leistinos ribos ar net peržengė ją, reikia imtis skubių priemonių tokiems nuokrypiams pašalinti ar bent sumažinti, t.y. imtis **reguliavimo** funkcijos vykdymo. Reguliavimas tai **septintoji valdymo proceso fazė**. Norint išvengti nepageidautinų nuokrypių, atsirandančių dėl nevaldomų poveikių į objektą, būtų tikslinga iš anksto imtis **prevencinių priemonių**, tačiau tuomet reikia iš anksto žinoti, kokie nevaldomi poveikiai į valdymo objektą gali atsirasti ir kaip jie paveiks valdymo objektą. Taip atsiranda **aštuntoji valdymo proceso fazė – prognozavimas**, kurios metu bandoma įvertinti būsimas nevaldomųjų faktorių reikšmes bei jų kitimo tendencijas. Prognozuojant gali paaiškėti ne tik neigiamų, bet ir teigiamų aplinkos pokyčių galimybės, į kurias nebuvo atsižvelgta strateginio ir operatyvinio planavimo fazėse. Todėl gali atsirasti strateginio ir operatyvinio **planų koregavimo** reikiamybė. Tai būtų **devintoji valdymo proceso fazė**. Tokia reikiamybė gali paaiškėti ir apskaitos bei kontrolės metu. Kadangi sudėtingų organizacijų valdymo sistemos yra daugiaėseloninės ir žemesnio ešelono valdymo kontūrų subjektai tampa valdymo objektais aukštesnio ešelono atžvilgiu, tai atsiranda ir **atsiskaitomybės funkcija**. Ši funkcija turėtų būti būdinga visiems valdymo subjektams, tame tarpe ir aukščiausiojo lygio vadovams, nes **subjektas, esantis organizacijos valdymo sistemos valdymo kontūrų piramidės viršūnėje, privalo kartkartėmis atsiskaityti organizacijos personalui, o taip pat tos visuomenės bei valstybės, į kurių sudėtį įeina organizacija, atstovams. Ataskaitų parengimas ir atsiskaitymas – dešimtoji valdymo proceso fazė**. Kadangi atskirų valdymo subjektų tikslai bei iš jų išplaukiantys veiksmai gali būti nepilnai suderinti ir tai gali kenkti organizacijos funkcionavimui, tai atsiranda valdymo subjektų tikslų ir jų veiklos **koordinavimo funkcija** (apie koordinavimo principus žiūr. 24 psl., o plačiau – antrąją 17 psl. paminėtą knygutę). Koordinavimo funkcijos atlikimas – **vienuoliktoji valdymo proceso platesniąja prasme fazė**.

Čia išvardintas valdymo funkcijas kartais vadina **pirminėmis valdymo funkcijomis**. Dauguma iš jų – gana sudėtingos, ir jų vykdyme dalyvauja ne tik organizacijų bei jų padalinių vadovai, bet ir pagalbinis personalas, padedantis valdymo subjektams ruošti ir realizuoti

valdymo sprendimus. Dėl to atsiranda daug pagalbinių funkcijų, kartais vadinamų **antrinėmis valdymo funkcijomis**. Antrinių valdymo funkcijų klasei, pavyzdžiui, priklauso **ryšių (vertikalių ir horizontalių, realizuojamų reglamentuotame, signaliniame ir užklausimų režimuose) sistemos sukūrimo ir palaikymo, informacijos rinkimo, kaupimo, saugojimo ir apdorojimo funkcijos**, be kurių organizacijos valdymas būtų neįmanomas. **Didelių ir sudėtingų organizacijų vadovai, būdami nebepajėgūs aprėpti visą organizaciją ir nagrinėti tiesiogiai vykstančius joje realius procesus, dažnai būna priversti to atsisakyti ir nagrinėti ne pačią realią organizaciją tiesiogiai su joje vykstančiais realiais procesais, o tik informaciją apie tai, kas vyksta organizacijoje, t.y., apsiriboti informacinių modelių nagrinėjimu**. Todėl atsirado tokios antrinės valdymo funkcijos kaip **informacinių modelių sudarymo valdymo tikslams ir tokių modelių analizės, prieš priimant valdymo sprendimus, funkcijos**, o taip pat **eksperimentų su matematiniais modeliais funkcijos**, pakeičiančios natūrinius eksperimentus ir padedančios prognozuoti įvairių valdymo sprendimų variantų pasekmes. Modeliai, vartojami valdymo tikslams, gali būti ir sisteminiai, ir optimizaciniai, ir analiziniai ir imitaciniai, ir deskriptyviniai, ir normatyviniai (žiūr. šios metodinės priemonės ketvirtąjį skyrelį), priklausyti įvairioms **stratoms** – modeliuojamos sistemos bei proceso apimties ir aprašymo detalumo lygiams. Matematinų modelių ir metodų bei kompiuterinės technikos taikymas valdyme sudarė galimybes atsirasti ir **valdymo sprendimų automatizavimo funkcijoms**, padedančioms efektyviau vykdyti tokias pirmines valdymo funkcijas kaip strateginis ir operatyvinis planavimas, valdymas pačia siaurąja prasme, reguliavimas, planų koregavimas.

Pirminių valdymo funkcijų vykdymas, pasitelkiant antrines, leidžia organizacijai siekti pakankamai **efektyvaus ir stabilaus funkcionavimo tikslų**, sudarančių **pirmąją organizacijos valdymo tikslų grupę**. Dekomponuojant tuos tikslus gaunamos pirminės valdymo funkcijos bei jas aptarnaujančios antrinės funkcijos, todėl čia aprašytas valdymo funkcijų išskyrimas valdymo procese gali būti interpretuojamas kaip tikslų dekompozicijos, naudojamos normatyviniame funkcinės struktūros projektavime, pavyzdys. Organizacijos valdymo sistema, siekdama šios – pirmosios – grupės valdymo tikslų, sudaro sąlygas veikiančiai organizacijai stabiliai vykdyti savo **misiją** ir iš jos išplaukiančias **pagrindines (darbines) funkcijas**.

Tačiau **darbinių ir ką tik aprašytųjų valdymo funkcijų vykdymas** galimas tik tada, kai jau egzistuoja pati **organizacija ir jos valdymo sistema**. Todėl organizacijos vadovams tenka vykdyti ne tik jau aprašytas valdymo funkcijas, bet taip pat organizacijos ir jos valdymo sistemos **organizavimo (struktūrų formavimo, kadro parinkimo ir t.t.) funkcijas**. Be to, organizacijos ir jos padalinių **vadovai turi rūpintis ne tik efektyviu ir stabiliu organizacijos funkcionavimu, bet ir jos bei jos valdymo sistemos evoliucija**. Todėl atsiranda dar ir **antroji organizacijos valdymo tikslų grupė** ir iš jų išplaukiančios **organizacijos ir jos valdymo sistemos organizavimo bei jų veiklos tobulinimo funkcijos**, įeinančios į valdymo **plačiąją prasme** sudėtį.

Organizavimo bei organizacijų tobulinimo funkcijos nėra elementarios, ir jų vykdymas – sudėtingas procesas. Apie tai byloja ir ši metodinė priemonė, skirta, būtent, organizacijų kūrimo ir jų tobulinimo metodologijos aprašymui. Organizavimo ir organizacijų veiklos tobulinimo funkcijoms, o taip pat šių funkcijų sudėtinėms dalims, gaunamoms dekomponuojant organizavimo ir organizacijų tobulinimo tikslus į tikslų medžius, būdinga tai, kad dauguma iš jų yra **pilnai ar dalinai kūrybinės**. Todėl jų vykdymas negali būti taip detalai reglamentuotas, kad būtų tikslinga ir galima sudaryti maršrutines technologines ir operacines kortas (žiūr. 4 psl.). Tokios funkcijos, žinoma, yra išvardijamos struktūrinėje NOD ir jų vykdymas numatomas organizacijos bei jos vadovų planuose, tačiau vietoj MTK ir OK apsiribojama OI ir metodinių nurodymų bei metodinių priemonių, padedančių jų vykdytojams, sudarymu. Metodinės priemonės ir metodiniai nurodymai, priklausomai nuo to, kokiems darbuotojams jie skirti ir kokius darbus aprašo, gali skirtis savo detalumo lygiu ir aprašomos veiklos apimtimi, t.y. gali būti parašyti įvairiose stratose. Ši priemonė, kurios

eilutes dabar skaitote, priklauso išplėstų metodinių priemonių stratai, artimai vadovėlių lygiui, o jos pirmasis ir trečiasis priedai yra metodinių nurodymų pavyzdžiai. Tokia šios metodinės priemonės parašymo strata pasirinkta todėl, kad šios priemonės paskirtis – **ne tik aprašyti, ką ir kaip reikia daryti, bet ir paaiškinti, kodėl taip reikia daryti, supažindinti skaitytoją su sistetine terminologija bei su sisteminio mąstymo, sisteminio modeliavimo ir sisteminio projektavimo pagrindais, paruošti jį specialios literatūros ir specialios projektinės dokumentacijos skaitymui bei darbui kompleksiniuose kolektyvuose**. Skaitydami šią metodinę priemonę, SIS projektuotojai ir Koordinacinės tarybos nariai turėtų ne tik šiek tiek papildyti savo žinias tose naujose srityse, kuriose daugumai iš jų anksčiau nebuvo tekę dirbti, bet ir palengva priprasti prie sisteminio mąstymo, suprasti ir įsisavinti jį.

Organizavimo, organizacijų bei jų valdymo sistemų tobulinimo ir aukščiau aprašytos pirminės valdymo „platesniąja prasme“ funkcijos sudaro funkcijų, vadinamų **vadovavimo funkcijomis**, klasę, įeinančių į valdymo pačia plačiąja prasme sudėtį. Vadovavimo funkcijų vykdymas vadinamas **vadovavimu**, o mokslas apie vadovavimą – vadybos mokslu arba tiesiog **vadyba**. Kalbant apie valdymo procesą pačia plačiąja prasme ir valdymo funkcijas, derėtų skirti **valdymo pačia siaurąja prasme** („komandavimo“, „rikiavimo“, „vairavimo“), apimančio valdymo „platesniąja prasme“ trečiąją ir ketvirtąją fazes, **valdymo platesniąja prasme**, arba tiesiog valdymo (angliškai vadinamo „control“, o rusiškai – „upravljenje“), ir **valdymo pačia plačiąja prasme, arba vadovavimo** (angliškai – „management“, „government“, rusiškai „rukovodstvo“), **sąvokas**, turint omenyje, kad vadovavimas (vadyba) yra platesnė sąvoka, negu jau veikiančios valdymo sistemos valdymas („control“), nors žodis „valdymas“ neretai vartojamas ir pačia plačiąja prasme, aprėpiančia ir sąvoką „vadovavimas“ (pavyzdžiui, „valstybės valdymas“ – „government“, „valdymo sistema“, „valdymo subjektas“, „valdymo objektas“ ir pan.). Šioje metodinėje priemonėje sąvoką „valdymo funkcijos“ vartosime pačia plačiąja prasme, jungiančia pirmines ir antrines valdymo funkcijas, o taip pat organizavimo ir organizacijos bei jos valdymo sistemos tobulinimo funkcijas.

Organizacijų valdymo sistemose išskiriami du valdymo (pačia plačiąja prasme) tipai: **tiesioginis vadovavimas** (rusiškai „lineinoje rukovodstvo“) ir **funkcinis valdymas** (rusiškai – „funkcionaljnoje upravljenje“). Tiesioginiam vadovavimui būdingas aktyviųjų valdymo objektų (žmonių) personalinis pavaldumas valdymo subjektui visais darbo organizacijoje klausimais ir visapusiška valdymo subjekto atsakomybė už jo vadovaujamo padalinio ir visų į tą padalinį įeinančių valdymo objektų darbą. Funkciniam valdymui būdingas **ne visapusiškas personalinis, o tik funkcinis pavaldumas**, t.y. pavaldumas tik tam tikrų funkcijų vykdyme, tik tam tikru aspektu. Funkciniam valdymui būdinga taip pat tai, kad valdymo subjektas yra atsakingas už valdymo objekto veiklą ne visais aspektais, o tik tam tikrų funkcijų vykdymo aspektu. Pavyzdžiui, gamybinėje įmonėje jos direktorius, cechų viršininkai ir tų cechų barų meistrai sudaro **triešeloninę tiesioginio vadovavimo sistemą**, vadovaujančią **pagrindiniam procesui** – gamybai, o **tą procesą ir tiesioginio vadovavimo sistemą aptarnauja įvairūs funkciniai padaliniai** (kadrų skyrius, buhalterija, vyr. technologo skyrius, materialinio bei techninio tiekimo skyrius ir t.t.), kurių viduje veikia nuosavos tiesioginio vadovavimo sistemos, tačiau **tų padalinių santykiams su visos įmonės tiesioginio vadovavimo sistemos valdymo objektais būdingas funkcinis valdymas**. Funkcinis valdymas yra būdingas ir Statistikos departamento santykiams su kitomis organizacijomis, teikiančiomis statistinėms tarnyboms statistinę informaciją, tačiau pats Statistikos departamentas ir jo vadovas tiesiogiai pavaldus Vyriausybei. Tiesioginio vadovavimo ir funkcinio valdymo funkcijas bei santykius nesunku rasti ir pačiame Statistikos departamente. Į tokių valdymo funkcijų tipų egzistavimą būtina atsižvelgti ir SIS projektuotojams, nagrinėjantiems statistinėse tarnybose vykdomas funkcijas ir projektuojantiems naują – tobulesnę ir pritaikytą SIS funkcionavimo sąlygoms – funkcinę statistinės sistemos struktūrą.

Paskirstant vadovavimo funkcijas jų vykdytojams organizacijos vykdomosios struktūros normatyvinio projektavimo metu, reikia atsižvelgti ir į tai, jog **pirmosios grupės**

tikslų – organizacijos efektyvaus ir stabilaus funkcionavimo užtikrinimo tikslų – siekimas ir iš jų išplaukiančių pirminių valdymo funkcijų vykdymas yra organizacijos valdymo sistemos žemutiniojo ešelono pagrindinė paskirtis ir prerogatyva, o antrosios grupės tikslų – organizavimo ir organizacijos bei jos valdymo sistemos tobulinimo tikslų – siekimas ir iš jų išplaukiančių vadovavimo funkcijų vykdymas yra organizacijos vadovų, jos valdymo sistemos aukšutinio ešelono pagrindinė paskirtis ir prerogatyva. Šio labai svarbaus valdymo funkcijų paskirstymo principo reikalavimų gana dažnai nesilaikoma arba laikomasi nepakankamai: didelė dalis aukšutinio ešelono vadovų turi žalingą įprotį vykdyti ne savo, o žemesnių ešelonų funkcijas vietoj to, kad griežtai pareikalautų iš savo pavaldinių kokybiško savo funkcijų vykdymo, o pastariesiems nesugebant tai daryti, keistų juos sugebančiais. Tokie funkcijų ir atsakomybės paskirstymo principo pažeidimai daro **dvigubą žalą**: dėl aukštesnio ešelono vadovų kišimosi į žemesnio ešelono kasdieninę veiklą, perimant žemesniojo ešelono darbuotojų funkcijas ir atsakomybę, **sumažėja žemesniojo ešelono darbuotojų atsakingumas ir iniciatyvumas**, jų pasitikėjimas savo jėgomis, o patiems aukštesniojo lygio vadovams dažnai **nebelieka pakankamai laiko nuosavų funkcijų vykdymui** – strateginiam planavimui, organizacijos bei jos veiklos ir valdymo sistemos tobulinimui, dėl to **organizacija pradeda atsilikti nuo gyvenimo reikalavimų** ir po kurio laiko susiduria su sunkiai išsprendžiamomis problemomis. **Aukšutinio ešelono vadovai turi sukurti tokia organizacijos valdymo sistemą, kad organizacija, kardinaliai nesikeičiant išorinėms sąlygoms, galėtų stabiliai ir efektyviai funkcionuoti ir be tu vadovų kišimosi.** Tie vadovai turi stebėti ir analizuoti organizacijos funkcionavimą lyg ir „iš šalies“, **įsikišdami į jo eigą tik organizacijos ir jos veiklos tobulinimo, prisitaikymo prie pasikeitusios aplinkos tikslais, o be to gana retais „avariniais“ atvejais, kada reikia keisti kai kuriuos žemesniojo ešelono vadovus** dėl to, kad jie nesugeba pakankamai gerai vykdyti savo funkcijų ir spręsti savo lygio problemų.

Kadangi pirmosios ir antrosios grupės tikslai, iš vienos pusės, papildo vieni kitus, sudarydami geresnes sąlygas vieni kitų pasiekimui besikeičiančioje aplinkoje, iš kitos pusės šiek tiek prieštarauja vieni kitiems, nes organizacijos ir jos veiklos tobulinimas neįmanomas be pasikeitimų organizacijoje, o bet kokie pasikeitimai trikdo stabilų funkcionavimą ir gali kuriam laikui sumažinti jo efektyvumą, tai tenka taip organizuoti organizacijos veiklą, kad būtų išlaikoma pusiausvyra tarp stabilaus funkcionavimo ir evoliucijos procesų, kad kreivė, vaizduojanti organizacijos veiklos efektyvumo kitimą laike, kaskart kiltų į viršų, o jos bangavimų amplitudė nebūtų pavojingai didelė. Tokios pusiausvyros palaikymas, vykdant aukštesniojo ešelono užduotis, yra vienas pagrindinių valdymo sistemos viduriniųjų ešelonų uždavinių. Dėl dviejų grupių valdymo tikslų egzistavimo bei dėl tobulinimo ir stabilaus funkcionavimo procesų vieni kitiems prieštaravimo ir jų derinimo būtinybės didelė dalis sudėtingų organizacijų valdymo sistemų yra, būtent, **triešloninės**: žemutinis ešelonas valdo pagrindinius (darbinius) organizacijoje vykstančius procesus, siekdamas, kad tie procesai vyktų stabiliai ir efektyviai, aukštutinis ešelonas vadovauja organizacijos tobulinimui, įskaitant ir jos plėtimą bei prisitaikymą prie išorinių sąlygų, o viduriniojo ešelono vadovai, vykdydami aukšutiniojo ešelono užduotis, stengiasi sudaryti ir palaikyti sąlygas, reikalingas žemutiniojo ešelono efektyviai veiklai ir tikslų pasiekimui. Tik labai didelėse organizacijose viduriniųjų valdymo ešelonų gali būti ne vienas, o keli. Mažose organizacijose valdymo sistema būna ir dviešloninė, o paprastose – net vienaėšloninė. Dviešloninėse valdymo sistemose aukštutinis ešelonas vykdo ir viduriniojo ešelono funkcijas, o vienaėšloninėse – visų trijų valdymo sistemos ešelonų funkcijas.

Valdymo sistemos normatyvinės struktūros projektavimu baigiamas visos organizacijos vykdomosios struktūros normatyvinis projektavimas. Jo metu nustatomos ir aprašomos visos valdymui reikalingos funkcijos, o taip pat ir tos, kurių galėjo nebūti normatyvinėje funkcinėje struktūroje, gautoje dekomponuojant pagrindinius organizacijos tikslus. **Tokios funkcijos įjungiamos į normatyvinę funkcinę struktūrą, papildant organizacijos tikslų sąrašą jos valdymo tikslais.** Taip gaunama pilna normatyvinė

vykdomoji funkcinė struktūra.

Normatyvinis funkcinis projektavimas atitinka šeštajame skyrelyje aprašytą sistemotechninį makroprojektavimą. Jis nesibaigia normatyvinės vykdomosios funkcinės struktūros sudarymu, nes **reikalinga suprojektuoti dar ir normatyvinę funkcijų vykdymo technologiją ir ją aprašančią TNOD** (o kartais – ir PNOD). Tas darbas pradedamas jau projektuojant normatyvinę vykdomąją funkcinę struktūrą, nes, jungiant funkcijas į paketus, reikia įvertinti ir funkcijų giminingumą ir jų įvykdymui reikalingo darbo trukmę, o tai galima sužinoti tik projektuojant atskirų funkcijų vykdymo procesus. Tačiau atskirų elementarių funkcijų aprašymu ir jų OK bei OI sudarymu dar nesibaigia normatyvinis TNOD projektavimas, nes reikia dar suprojektuoti sudėtingų funkcijų vykdymo technologinius procesus ir juos aprašančias MTK, suderinti tuos technologinius procesus erdvėje ir laike. Tokio makroprojektavimo ir jo loginių etapų (šeštajame skyrelyje jau minėtų „vienetinio siūlo“, „pilno apkrovimo“ ir „rungtinių projektavimo“), o taip pat sudėtingų funkcijų vykdymo technologinių procesų optimizavimo aprašymas išeina už šio skyrelio temos – organizacijos funkcinės struktūros normatyvinio projektavimo – ribų, ir todėl tam aprašymui skiriamas kitas – dešimtas – šios metodinės priemonės skyrelis.

Kaip jau buvo minėta šio skyrelio pradžioje, **normatyvinis funkcinis organizacijos struktūros projektavimo metodas tais atvejais, kai organizacija sudėtinga, negali užtikrinti, kad į suprojektuotą normatyvinę vykdomąją funkcinę struktūrą būtinai pateks visos reikalingos funkcijos ir nepateks nė vienos bereikalingos.** Todėl, kai egzistuoja projektuojamos sistemos analogas arba kai projektuojamas veikiančios organizacijos struktūros patobulintas variantas, **normatyvinį funkcinį projektavimą tikslinga papildyti veikiančios organizacijos vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymu.**

Organizacijos vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymas pradedamas nuo organizacijoje egzistuojančios struktūrinės ir technologinės NOD nagrinėjimo ir projektuotojų pokalbių (interviu) su organizacijos darbuotojais, išsiaiškinant tuos klausimus, kurie kilo projektuotojams, benagrinėjant SNOD, TNOD ir PNOD bei sprendžiant, kokia turėtų būti sistemos struktūra.

Naujos funkcinės struktūros normatyvinis projektavimas vyksta lygiagrečiai su deskriptyvinio veikiančios struktūros modelio sudarymu bei jo tobulinimu, nes deskriptyvinio modelio turėjimas padeda gauti pilnesnį ir realistiškesnį normatyvinės funkcinės struktūros modelį, o normatyvinis projektavimas padeda pastebėti deskriptyvinio modelio trūkumus ir to modelio patobulinimo galimybes.

Kartais bandoma sudaryti deskriptyvinį vykdomosios funkcinės struktūros modelį, nenagrinėjant SNOD, TNOD ir PNOD, o apsiribojant tik klausimų organizacijos darbuotojams pateikimu ir projektuotojų interviu su organizacijos personalu organizavimu. Toks kelias yra galimas, tačiau, juo einant, susiduriama su sunkumais, kurių nebūtų (ar jie būtų žymiai mažesni), jei projektuotojai, prieš imdami interviu, susipažintų su SNOD, TNOD bei PNOD ir turėtų prieš akis tą dokumentaciją interviu metu.

Pirmasis sunkumas – tai **susikalbėjimo vienu su kitais, vieni kitų supratimo sunkumas.** Gana dažnai organizacijos darbuotojai nepilnai (o kartais ir visai neteisingai) supranta, ko projektuotojai iš jų nori, o projektuotojai nepilnai (o kartais ir visai neteisingai) supranta organizacijos personalo atsakymus. Tas tarpusavio nesupratimas – nepakankamo vienas kitų veiklos ir jos specifikos žinojimo pasekmė: projektuotojai nepajėgūs būti specialistais visose organizacijos darbuotojų veiklos srityse, o darbuotojai nežino projektuotojų darbo ir jame išskylančių problemų specifikos. Todėl praeina nemažai laiko, kol projektuotojai ir organizacijos darbuotojai pradeda vienas kitą pakankamai suprasti. Jei projektuotojai, prieš pradėdami pokalbius su darbuotojais, būtų išstudijavę atitinkamą SNOD, TNOD ir PNOD, tai jiems susikalbėti su darbuotojais būtų buvę žymiai lengviau, o ir darbuotojams projektuotojų klausimai būtų buvę suprantamesni, nes būtų buvę konkretesni, labiau surišti su darbuotojų darbu.

Antrasis sunkumas atsiranda todėl, kad „greitosiomis“ suformuluoti atsakymai į interviu metu pateiktus ir ne visada pilnai suprastus klausimus, ne visada yra pakankamai tikslūs ir pakankamai pilni. Todėl neretai projektuotojai suprojektuoja ne visai tokią sistemą, kokios norėtų jos vartotojai – organizacijos darbuotojai. Tai paaiškėja tada, kada sistema pradedama eksploatuoti. Tuomet iškyla nemažai konfliktų tarp projektuotojų ir vartotojų, nes sistemos perprojektavimas ir perdarymas nemažai kainuoja. Tada prasideda beprasmiškas kaltųjų dėl projekte padarytų klaidų ieškojimas su iš anksto „užprogramuotu“ rezultatu, nes **kaltais beveik visuomet lieka projektuotojai, kadangi net tada, kada projektuotojus suklaidino vartotojai savo nepakankamai apgalvotais atsakymais, to įrodyti praktiškai neįmanoma.** Panašių sunkumų ir klaidų žymiai sumažėtų ir abiejų pusių atsakingumas žymiai padidėtų, jei **projektuotojai iš anksto apgalvotų savo klausimus, iš anksto pateiktų juos raštiškai, prašydami raštiškų atsakymų iš darbuotojų, ir tik gavę ir išnagrinėję tuos atsakymus, eitų į interviu su darbuotojais,** kad išsiaiškintų kai kurias tų atsakymų vietas bei kilusius naujus klausimus. Tačiau toks projektuotojų ir organizacijos personalo bendravimo būdas reikalauja daug rašliavos ir daug laiko, o tai nėra pageidautina. Rašliavos apimtį ir reikalingą deskriptyvinio modelio sudarymui laiką galima žymiai sumažinti, kai organizacijoje veikia pakankamai tobulai sudaryta SNOD, TNOD ir PNOD, ir kai projektuotojai gerai ją išstudijuoja dar prieš klausimynų sudarymą ir interviu ėmimą.

Trečiasis sunkumas išplaukia iš organizacijos struktūros projektavimo darbų apimties ir sudėtingumo. Naivu būtų tikėtis, kad keli projektuotojai (o didelė projektuotojų grupė brangiai kainuotų ir dėl antrojo informacinio barjero efekto dirbtų nepakankamai efektyviai) per trumpą laiką sugebės įsigilinti į kiekvieno didelės organizacijos darbuotojo darbus, pakankamai pilnai suvokti jų specifiką ir algoritmus, aprašyti juos, nuspręsti, kuriuos iš tų darbų tikslinga kompiuterizuoti, ir sudaryti tokį visos organizacijos funkcionavimo naujomis sąlygomis projektą, kurio įdiegimo efektas būtų tikrai vertas tokio didelio ir sudėtingo darbo. **Projektuotojų darbo apimtį ir trukmę galima būtų žymiai sumažinti, ne tik nepabloginant projektų kokybės, bet ir žymiai pagerinant ją, jei projektuotojai jau pačioje savo darbo pradžioje remtųsi organizacijoje veikiančia pakankamai kokybiška SNOD (organizacijos įstatais, jos struktūriniu schema, padalinių nuostatais ir jų struktūros schemomis, darbuotojų pareiginėmis instrukcijomis ar jų pareigybių pasais) bei TNOD ir PNOD (MTK, OK ar OI, organizacijos, jos padalinių ir darbuotojų darbo planais ir tvarkaraščiais), bendradarbiautų su organizacijos personalu, bendromis jėgomis tobulindami organizacijos struktūrą ir veiklą bei tą veiklą aprašančią ir reglamentuojančią NOD.**

Jau šių trijų argumentų pilnai pakanka, kad būtų įrodytas SNOD, TNOD ir PNOD panaudojimo, kuriant organizacijos struktūros deskriptyvinį modelį ir projektuojant naują – tobulesnę – funkcinę organizacijos struktūrą, tikslingumas, o taip pat nuomonės, kad, projektuojant SIS, nėra reikalo nagrinėti ir tobulinti SNOD, TNOD ir PNOD, klaidingumas.

Bet ir nugalėjus paplitusį projektuotojų, organizacijų vadovų ir jų darbuotojų tarpe psichologinį barjerą dėl SNOD, TNOD ir PNOD panaudojimo tikslingumo, kuriant organizacijos veiklos kompiuterizavimo projektus, dar lieka kiti sunkumai, kurie iškyla, bandant pasinaudoti SNOD, TNOD ir PNOD organizacijos struktūros ir jos veiklos deskriptyvinio modelio sudaryme.

Tų sunkumų esmę sudaro tai, kad kai kurių organizacijų turima SNOD yra labai netobula, o technologinės NOD daugumoje organizacijų, išskyrus gamybinės, kuriose jau seniai naudojama gamybinių procesų TNOD, iš viso nėra. Pavyzdžiui, pareigūnų pareiginės instrukcijos (PI) neretai yra mažai informatyvios ir neatitinka savo paskirties būti darbuotojų veiklos reglamentavimo priemone, **neleidžiančia darbuotojams dirbti blogiau, negu buvo dirbama iki šiol, bet skatinančia juos tobulinti tą veiklą, fiksuojant atsiradusią teigiamą patirtį atitinkamoje TNOD, ir tokiu būdu tampančia darbuotojų patirties kaupimo ir jos**

perdavimo mažiau patyrusiems bei savo įpėdiniams priemone. Kai kuriose valstybinėse organizacijose pareigūnų PI parašytos tik „dėl formos“, „dėl paukščiuko“, matyt, tik todėl, kad kažkokia „aukščiau stovinti instancija“ reikalauja tokių dokumentų sudarymo. Savaiame aišku, kad **tokių PI rašymas ir skaitymas yra tik tuščias laiko gaišinimas, ir tokių PI turėjimas yra net blogiau, negu jokių PI neturėjimas.** Tačiau, jei organizacijų vadovai, suprasdami organizacijos veiklos organizavimo ir jos racionalizavimo reikšmę, skiria pakankamai dėmesio tos veiklos efektyvumo palaikymui ir didinimui, darbuotojų patirties kaupimui ir jos perdavimui ateinantiems į organizaciją naujiems darbuotojams, tai tokiose organizacijose visi pareigūnai, pradedant nuo pačių aukščiausiųjų vadovų, turi kokybiškas PI (arba pareigybių pasus) ir visą kitą SNOD, TNOD ir PNOD. Tokia dokumentacija labai naudinga, automatizuojant organizacijos darbuotojų veiklą, nes žymiai palengvina ir pagreitina projektuotojų darbą.

Statistikos departamento pareigūnų PI nėra visiškai neinformatyvios, nes jos, nors ir nepakankamai vienareikšmiškai ir pilnai, vis dėlto apibrėžia funkcijų paskirstymą pareigūnams ir jų pavaldumo bei atsiskaitomybės ryšius. Tačiau šios PI dar netenkina visų reikalavimų kaip pareigūnų veiklos reglamentavimo priemonė ir net nepretenduoja būti darbuotojų patirties kaupimo ir jos perdavimo priemone. Matyt, tų PI sudarytojai net tokio tikslo sau nekėlė. Tai rodo, kad Statistikos departamente dar neskiriama dėmesio darbuotojų veiklos organizavimui ir tobulinimui, jos efektyvumo didinimui. Tai sunkiai derinasi (ir, galima sakyti, tiesiog disonuoja) su pasiryžimu kurti tokią brangią ir sudėtingą sistemą kaip SIS, kadangi **SIS – Lietuvos statistinės sistemos veiklos tobulinimo priemonė ir ta priemonė ne tik techninė, bet ir organizacinė,** nes negalima įdiegti sudėtingos kompiuterinės ir informacinės sistemos, nekeičiant organizacijos struktūros ir darbuotojų veiklos. Todėl manau, kad, pasakius „A“, teks pasakyti ir „B“: pradėjus kurti SIS, teks užsiimti ir žmonių veiklos organizavimu, jos tobulinimu. Atsižvelgdamas į tai, ir parašiau jau minėtus metodinius nurodymus, skirtus PI projektavimui, paverčiant pareigybines instrukcijas į pareigybių pasus (PP). Projektuojant PP, o taip pat ir SIS, gali labai padėti grafiškas funkcijų vykdymo procesų vaizdavimas – grafinė TNOD ir PNOD išraiška. Tokios grafinės TNOD ir PNOD pavyzdžių, aprašančių kai kuriuos Statistikos departamente vykstančius procesus, skaitytojas gali rasti trečiajame šios priemonės priede.

Naudojantis minėtais metodiniais nurodymais bei šios priemonės, priedais, galima sukurti pakankamai tobulus PP, o taip pat žymiai patobulinti visą kitą SNOD, TNOD ir PNOD. Tai, iš vienos pusės, pagerintų organizacijos veiklą dar iki kompiuterinės sistemos įdiegimo, iš kitos pusės, žymiai palengvintų kompiuterinės sistemos projektavimą ir įdiegimą. **Todėl tose organizacijose, kuriose nėra pakankamai kokybiškos SNOD, TNOD ir PNOD, kompiuterizuotos sistemos projektavimą ir tokį to projektavimo etapą, kaip vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymą ir to modelio tobulinimą, būtų tikslinga vykdyti kartu su SNOD, TNOD ir PNOD projektavimu, pradedant SNOD, TNOD ir PNOD projektavimą net šiek tiek anksčiau, negu naujos – kompiuterizuotos – sistemos struktūros projektavimas.**

Žinoma, tokį darbą negalima padaryti pakankamai greitai ir kokybiškai, neįtraukiant į jį visų organizacijos darbuotojų, nekeliant jų kvalifikacijos, neskatinant jų iniciatyvos. Čia vėl tenka susidurti su psichologiniu barjeru: **kai kurie vadovai, vis dar nesuprasdami, kad svarbiausioji kiekvienos organizacijos dalis, nulemianti organizacijos veiklos efektyvumą, yra jos žmonės, nelinkę skirti daug pastangų jų veiklos organizavimui, jų kvalifikacijos kėlimui. Pasitaiko ir tokių vadovų, kurie taip nepasitiki savo pavaldiniais, taip mažai vertina jų sugebėjimus, kad net nedrįsta iš jų pareikalauti gilesnio savo funkcijų suvokimo ir jų vykdymo tobulinimo, nesiryžta investuoti lėšas į darbuotojų kvalifikacijos kėlimą, kuris tampa būtinu, tobulinant organizacijos veiklą, didinant jos efektyvumą.** Tokiais atvejais reikėtų keisti arba kai kuriuos nieko nesugebančius darbuotojus, arba jų vadovus, nesugebančius ar nenorinčius organizuoti organizacijos veiklos tobulinimo procesų, arba ir vienus, ir kitus.

O jei vadovai supranta organizacijos veiklos efektyvumo didinimo, kompiuterizuojant tą veiklą, būtinumą, tai anksčiau ar vėliau jie bus priversti suprasti ir darbuotojų veiklos bei tą veiklą aprašančios dokumentacijos projektavimo ir tobulinimo bei darbuotojų kvalifikacijos kėlimo būtinumą.

Tai labai vaizdžiai rodo SIS kūrimo pavyzdys.

Negalima kompiuterizuoti darbuotojų veiklos, nežinant darbuotojų funkcijų bei jų vykdymo algoritmų. Todėl tų funkcijų paskirstymo ir jų vykdymo procesų aprašymai – SNOD, TNOD ir PNOD – reikalingi jau SIS projektavimo metu, ir jei jų nebūtų, jie turėtų būti sudaryti.

Todėl čia ir siūloma organizacijos normatyvinio projektavimo ir deskriptyvinio modeliavimo bei abiejų šių procesų sintezės metu vykdyti **normatyvinį SNOD, TNOD ir PNOD projektavimą, deskriptyvinį darbuotojų veiklos modelių**, aprašančių organizacijoje esantį funkcijų paskirstymą ir tų funkcijų vykdymo procesus, **kūrimą**, o taip pat **normatyvinio projektavimo ir deskriptyvinio modeliavimo rezultatų sintezę**, gaunant patobulintą ir pritaikytą projektuojamos kompiuterizuotos sistemos funkcionavimo sąlygomis SNOD, TNOD ir PNOD.

Ši dokumentacija, parengta SIS projektavimo metu ir būdama sudėtine SIS projekto dalimi, reikalinga ir SIS diegimo metu, nes negalima įdiegti sudėtingą kompiuterinę sistemą, atitinkamai neorganizavus žmonių, kurie naudosis SIS technine, informacine ir programine įranga, veiklos. Visa ta sudėtinga ir brangi įranga nieko neduos, jei žmonės nemokės ir nenorės ja naudotis, jei nebus suprojektuoti ir suorganizuoti žmonių ir kompiuterių bendravimo procesai, besijungiantys į visos organizacijos funkcionavimo procesą.

Jei pačioje SIS projektavimo pradžioje statistinės sistemos darbuotojai sukurtų pakankamai detalius deskriptyvinius savo veiklos modelius, sudarydami SNOD, TNOD ir PNOD, tai ir SIS projektuotojai, naudodamiesi šia dokumentacija, galėtų žymiai produktyviau organizuoti savo interviu su statistinės sistemos darbuotojais ir žymiai greičiau sukurtų SIS projektą, pakankamai tenkinantį statistinės sistemos poreikius. SIS projektavimo ir diegimo metu vyktų ir tos SNOD, TNOD ir PNOD tobulinimas, pritaikant ją SIS funkcionavimo sąlygoms. Kitaip tariant, kartu su informacinės, programinės ir techninės įrangos kūrimo procesu vyktų ir SIS „organizacinės įrangos“ kūrimo procesas, SIS „žmogiškosios“ sudėtinės dalies pasiruošimas darbui SIS funkcionavimo sąlygomis, statistinės sistemos darbuotojų kvalifikacijos kėlimas.

Jeigu kas nors galvoja, kad SIS sukurti ir efektyviai eksploatuoti galima ir nekeliant statistinės sistemos darbuotojų kvalifikacijos, tai labai apsirinka, nes SIS kuriama ne tik tam, kad palengvintų ir pagreitintų dabar dirbamą statistinėse tarnybose darbą, bet ir tam, kad suteiktų statistinės sistemos darbuotojams galimybę gauti žymiai daugiau naudingos informacijos iš surinktų statistinių duomenų, apdorojant juos ne tik aprašomosios, bet ir matematinės statistikos metodais, panaudojant juos matematiniam modeliavimui, prognozavimui, priimamų sprendimų galimų pasekmių vertinimui (žiūr. antrąją šios metodinės priemonės priedą). **Tokios galimybės atsiranda, įdiegus SIS techninę, programinę ir informacinę įrangą, tačiau tos galimybės galės tapti realybe ir būti pilnai išnaudotos tik tada, kai statistinės sistemos darbuotojai supras matematinės statistikos, matematinio modeliavimo ir prognozavimo metodų esmę bei vertę ir išmoks tais metodais naudotis kompiuterizuotos informacinės sistemos funkcionavimo sąlygomis.** O tai galima pasiekti, tik pastoviai keliant statistinės sistemos darbuotojų kvalifikaciją, negailint tam laiko ir lėšų. **Išlaidos tikslingam darbuotojų kvalifikacijos kėlimui, didinančiam jų darbo efektyvumą, atsiperka greičiau, negu kurios nors kitos išlaidos.**

Todėl atrodo nelabai normalu, kad Statistikos departamento struktūroje iki šiol dar nėra tarnybos, kuri būtų pajėgi atsakyti už mokslinį statistinių tyrimų lygį bei to lygio kėlimą, siekiant, kad Lietuvos statistinė sistema neatsiliktų nuo mokslo ir technikos progreso, nors jau yra tokios tarnybos užuomazga. Valstybinė statistinė sistema nors ir nėra grynai mokslinė

organizacija, bet yra gana artima jai, nes statistiniai tyrimai, jei norima, kad jų rezultatai turėtų objektyvią vertę, negali nesiremti statistikos (aprašomosios ir matematinės) mokslu, negali būti vykdomi, nesivadovaujant mokslškai pagrįstomis metodikomis, kurios turėtų būti nuolat tobulinamos, atsižvelgiant į naujus mokslo ir technikos pasiekimus. Todėl būtų tikslinga, sustiprinus ir išplėtus statistinių ekonominių tyrimų ir sisteminės analizės skyrių, kuris dabar dar nepajėgus vykdyti visas Statistikos departamentui reikalingas mokslinio aptarnavimo ir mokslinio vadovavimo funkcijas, sukurti SD tarnybą, atsakingą už visų valstybinėje statistinėje sistemoje vykdomų statistinių tyrimų mokslinį lygį, naujausių mokslo ir technikos pasiekimų diegimą ir taikymą, statistinės sistemos darbuotojų mokslinės kvalifikacijos kėlimą. Toje tarnyboje galėtų būti matematinės statistikos metodų taikymo, sisteminės statistinės analizės, sisteminio statistinio modeliavimo bei prognozavimo, statistinių tyrimų ir eksperimentų planavimo, statistinės sistemos darbuotojų mokslinės kvalifikacijos kėlimo ir kiti valstybinės statistikos sistemai reikalingi moksliniai padaliniai.

Aptarę SNOD, TNOD ir PNOD projektavimo bei panaudojimo ir organizacijos darbuotojų kvalifikacijos kėlimo tikslingumą ir būtinumą, projektuojant ir diegiant kompiuterizuotą organizacijos veiklos tobulinimo ir jos efektyvumo didinimo sistemą, ties kuo specialiai buvo apsistota kiek ilgiau todėl, kad SNOD, TNOD bei PNOD panaudojimo ir darbuotojų kvalifikacijos kėlimo klausimais vis dar gana dažnai pasitaiko klaidingų nuomonių, trukdančių racionaliam projektavimo ir diegimo darbų organizavimui, grįšime prie vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymo technologijos aprašymo.

Kaip jau buvo minėta, makroprojektuotojai – sisteminiai analitikai, kurdami organizacijos vykdomosios funkcinės struktūros deskriptyvinį modelį, pradeda nuo organizacijoje esančios SNOD analizės. Susipažinę su SNOD, o reikalui esant – ir su atitinkama TNOD bei PNOD, makroprojektuotojai sudaro jiems kilusių klausimų sąrašus – **klausimynus**. Tuos klausimynus makroprojektuotojai pateikia organizacijos padalinių vadovams, susitardami su jais dėl **interview** terminų ir dėl atsakymų raštu į pateiktus klausimus paruošimo iki tų terminų. Interview su padalinių vadovais bei darbuotojais metu makroprojektuotojai, naudodamiesi tais iš anksto paruoštais atsakymais į savo anksčiau pateiktus klausimus, o taip pat egzistuojančia SNOD, TNOD bei PNOD ir vykdomosios funkcinės struktūros normatyvinio projektavimo, vykstančio lygiagrečiai su deskriptyvinio modelio kūrimu, rezultatais, stengiasi gauti ir užfiksuoti **interview protokoluose** visą informaciją, reikalingą vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymui ir jo patobulinimui, atsižvelgiant į organizacijos veiklos kompiuterizavimą, o taip pat išsiaiškinti visus klausimus, iškilusius bestudijuojant SNOD, TNOD ir PNOD, bei papildomus klausimus, iškilančius pačių interview metu.

Tais atvejais, kai organizacijoje egzistuojanti SNOD (organizacijos bei jos padalinių struktūrą vaizduojančios schemas, organizacijos įstatai ir jos padalinių nuostatai bei pareigybių pasai) yra pakankamai pilna ir tenkina tokiai NOD keliamus reikalavimus, organizacijos vykdomosios funkcinės struktūros deskriptyvinio modelio sudarymas susideda tik iš informacijos, sukauptos struktūrinėje NOD, įvedimą į kompiuterį, paruošiant deskriptyvinį modelį palyginimui su normatyviniu, nes tuomet **pati tokia SNOD jau yra organizacijos vykdomosios funkcinės struktūros deskriptyvinis modelis** (žinoma, jei ta pakankamai pilna SNOD egzistuoja ne tik popieriuje, t.y. jei organizacijos darbuotojai vykdo visas toje SNOD numatytas funkcijas).

Tais atvejais, kai organizacijoje dar nesudaryta SNOD, taip pat, kai egzistuojanti SNOD nėra pakankamai tobula ir neatspindi visų organizacijoje vykdomų funkcijų bei jų paskirstymo, makroprojektuotojams kartu su organizacijos vadovais ir darbuotojais tenka užsiimti ne tik kompiuterizuotos sistemos, bet ir SNOD kūrimu bei tobulinimu. SNOD kūrimas bei tobulinimas pradedamas nuo to, kad makroprojektuotojai, remdamiesi organizacijos vykdomosios funkcinės struktūros normatyvinio projektavimo rezultatais, paruošia organizacijos darbuotojams ir jų vadovams klausimynus, skirtus išsiaiškinimui, kokios funkcijos vykdomos organizacijoje ir kaip jos paskirstytos tarp organizacijos padalinių

ir jų darbuotojų. Pateikdami tuos klausimynus organizacijos padalinių vadovams, makroprojektuotojai susitaria su jais dėl interviu terminų. Interviu metu makroprojektuotojai aptaria su organizacijos darbuotojais gautus iš jų atsakymus ir išsiaiškina kai kuriuos papildomus klausimus (pavyzdžiui: ar vykdomos organizacijoje tos normatyvinio modelio funkcijos, kurių nepamirėjo organizacijos darbuotojai, atsakydami į klausimynų klausimus ir aprašydami savo veiklą, o jei nevykdomos, tai kodėl; ar reikalingos tos darbuotojų paminėtos funkcijos, kurių nėra normatyviniame modelyje; kokių funkcijų vykdymą būtų tikslinga kompiuterizuoti). Po to makroprojektuotojai, apdoroję sukaupią informaciją, sudaro ir pateikia organizacijos darbuotojams savo pareigybių pasų projektavimo (esamų pareiginių instrukcijų tobulinimo) užduotis bei metodinius nurodymus, kaip aprašyti savo vykdomas funkcijas ir kaip sudaryti savo pareigybių pasus. Makroprojektuotojams, formuluojantiems pareigybių pasų projektavimo užduotis ir sudarinėjantiems tam projektavimui metodinius nurodymus, rekomenduotina pasinaudoti informacija iš šios metodinės priemonės pirmojo ir trečiojo priedų. Organizacijos ir jos padalinių vadovai papildomai gauna užduotis sudaryti organizacijos bei jos padalinių struktūrines schemas ir organizacijos įstatų bei padalinių nuostatų projektus. Makroprojektuotojai neapsiriboja užduočių SNOD projektavimui sudarymu bei išdalinimu organizacijos darbuotojams, bet ir patys dalyvauja tame projektavime, projektuodami organizacijos vykdomąją funkcinę struktūrą, pritaikytą kompiuterizuotos informacinės sistemos funkcionavimo sąlygoms.

Pareigybių pasų ir kitos SNOD projektavimo metu taip pat gali būti organizuojami interviu (ir net kelios jų serijos), tačiau tie interviu rengiami ne būtinai tik makroprojektuotojų, bet ir pačių organizacijos darbuotojų iniciatyva. Tokių interviu metu ne tik makroprojektuotojai aiškinasi jiems iškilusius klausimus apie organizacijoje vykdomas funkcijas, bet ir organizacijos darbuotojai gali gauti atsakymus į savo klausimus, kaip projektuoti SNOD ir TNOD. Todėl tokie interviu tampa projektuotojų kolektyvų, sudarytų iš makroprojektuotojų (sisteminių analitikų) ir organizacijos darbuotojų **pasitarimais** (o kartais – ir diskusijomis), t.y. **kompleksinių kolektyvų bendro darbo forma** su jai būdingais darbo metodais („proto atakomis“, „atakuojančiomis programomis“, „kolektyvinio proto sistemų“ taikymu ir pan.).

Tokių kompleksinių projektavimo grupių darbo rezultate po kelių iteracijų atsiranda ne tik pakankamai tobuli pareigybių pasų projektai, bet ir visa kita SNOD, o taip pat ir TNOD (rutininių darbų MTK ir OK bei OI, kūrybinių darbų aprašymai ir jų vykdymo metodiniai nurodymai bei OI). Tuo pačiu metu suprojektuojama ir organizacijos vykdomoji funkcinė struktūra bei PNOD.

Pasitaiko atvejų, kai organizacijų ir kai kurių jų padalinių vadovai, nesuprasdami darbo racionalaus organizavimo bei reglamentavimo ir darbo patirties kaupimo reikšmės, galvoja: „Visa ta SNOD, TNOD ir PNOD yra tik mokslininkų išmėslas, neduodantis organizacijai jokios naudos, ir, jei kompiuterizuotos sistemos projektuotojams, projektuojant tos sistemos funkcinę struktūrą ir funkcijų vykdymo procesus, tokios dokumentacijos reikia, tai tegul patys sau tokią dokumentaciją ir kuria, netrukdydami organizacijos personalui dirbti savo tiesioginį darbą“. Tokia nuomonė, nors ir būdama gana plačiai paplitusi, neteisinga, nes remiasi net keliom klaidingom prielaidom. **Pirmoji prielaida, teigianti, jog žmonių veiklos organizuoti ir reglamentuoti nereikia, nes jie sugeba susiorganizuoti patys, būdama teisinga nedideliems kolektyvams, yra klaidinga didelėms organizacijoms, kadangi dideli kolektyvai, jei jų veiklos specialiai neorganizuoti, tampa spontaniškai veikiančia minia. Antroji prielaida, teigianti, kad funkcijų paskirstymo darbuotojams ir jų vykdymo algoritmų fiksavimas normatyviniuose organizaciniuose dokumentuose neturi prasmės, nes sąžiningi darbuotojai ir be jokių dokumentaliai fiksuotų reglamentų sugeba gerai vykdyti savo darbą, būdama teisinga tais atvejais, kai darbuotojo veikla nesudėtinga ir jis ta veikla užsiima pakankamai ilgai, yra klaidinga tais atvejais, kai veikla yra sudėtinga, o tuo labiau tais atvejais, kai seną patyrusį darbuotoją dėl jo ligos ar dėl kitų priežasčių reikia pakeisti nauju darbuotoju. Trečioji prielaida, teigianti, kad darbo**

patirties fiksavimas dokumentuose yra bereikalingas biurokratizmas, nes atima darbuotojo laiką aprašymui to, ką jis ir be aprašymo gerai prisimena ir pritaiko savo darbe, būdama beveik teisinga vieno darbuotojo atžvilgiu, tampa klaidinga, kai mes pradėdame spręsti sukauptos patirties perdavimo kolegoms ir ateinančioms kartoms problemas. Dar labiau klaidinga yra prielaida, teigianti, kad galima sukurti kompiuterizuotos informacinės sistemos informacinius bankus ir programinę įrangą, gerai nežinant organizacijos darbuotojų veiklos ir jos specifikos. Taip pat yra klaidinga ir naivu tikėtis, jog keli makroprojektuotojai ir programuotojai per trumpą laiką, naudodamiesi vien tik interviu su organizacijos darbuotojais, patiems organizacijos darbuotojams nedalyvaujant projektavimo darbe, sugebės tiek detalčiai išstudijuoti ir tiek giliai suprasti bei tiek pilnai įsiminti visų didelės organizacijos darbuotojų darbą ir jų ryšius, kad galėtų kompetentingai spręsti, ką ir kaip tų darbuotojų veikloje tikslinga keisti, ką kompiuterizuoti, ir sugebėtų sudaryti racionalius tos veiklos algoritmus ir patogias vartotojams programas bei jiems reikalingus informacinius bankus. Pokalbiai išlieka tik pokalbiais net ir tuomet, kai juos pavadiname interviu: jie būna gana produktyvūs, kai visi pokalbio dalyviai iš anksto žino pokalbio problematiką ir tam iš anksto pasiruošia, kai jau turimi dokumentaliai užfiksuoti galimi problemų sprendimo variantai, su kuriais iš anksto susipažinę visi pokalbio dalyviai, ir belieka tik ką išsiaiškinti bei patikrinti, suderinti pokalbio dalyvių nuomones ir išsirinkti geriausius variantus; tačiau pokalbiai mažai ką duoda, kai vienas pokalbio dalyvis nelabai žino, ko nori iš jo kitas, kai abu pokalbio dalyviai nelabai supranta vienas kitą, nes kai kuriems žodžiams priskiria skirtingas reikšmes ir negali surasti bendrų tikslų. **Todėl turėtų praeiti neleistinai daug laiko, kol makroprojektuotojai, naudodamiesi vien tik interviu, vieni, be organizacijos darbuotojų pagalbos suprojektuotų organizacijos funkcinę struktūrą bei ją aprašančią SNOD, organizacijos darbuotojų veiklos algoritmus bei juos aprašančią TNOD ir PNOD, o taip pat sukurtų kompiuterizuotos informacinės sistemos techninę, informacinę ir programinę įrangas. Po to turėtų praeiti dar nemažai laiko, kol kompiuterizuotos sistemos projektuotojai išmokytų organizacijos darbuotojus ir jų vadovus tomis įrangomis efektyviai naudotis ir įtikintų juos, kad naujoviškai dirbti yra patogiau ir efektyviau, negu taip, kaip buvo dirbama iki šiol. Kadangi dauguma žmonių yra nekantrūs ir gana greit pavargsta ir nusivilia, tai minėtieji vadovai, dėl savo klaidingų įsitikinimų įstūmę kompiuterizuotos sistemos projektuotojus į tokį neperspektyvų kelią, nesulaukdami iš jų greitų ir apčiuopiamų rezultatų, matyt, apkaltintų pastaruosius nekompetentingumu ar net nesąžiningumu, o pačią kompiuterizuotos informacinės sistemos sukūrimo idėją – nerealumu ar nesavalaikiškumu, nepelnytai diskredituodami ją.**

Viso to galima išvengti ir pakankamai greitai sukurti patogią kompiuterizuotą informacinę sistemą, tos sistemos projektuotojams ir organizacijos darbuotojams kuriant normatyvinį organizacijos naujos būsenos projektą ir derinant jį su iš deskriptyvinio organizacijos modelio išplaukiančiais reikalavimais, o organizacijos darbuotojams su jos ir jos padalinių vadovais priešakyje kuriant ir tobulinant SNOD, TNOD ir PNOD, derinant tą NOD su rekomendacijomis bei reikalavimais, išplaukiančiais iš normatyvinio organizacijos projekto.

Kompiuterizuotos informacinės sistemos projektavimo ir organizacijos pervedimo į naują būseną procesas tampa dar trumpesnis, kai jau pačioje projektavimo pradžioje organizacijoje egzistuoja pakankamai pilna ir kokybiškai sudaryta struktūrinė ir technologinė NOD, kurią kompiuterinės informacinės sistemos projektavimo metu bendromis projektuotojų ir organizacijos darbuotojų jėgomis belieka tik modernizuoti, pritaikant kompiuterinės sistemos funkcionavimo sąlygoms. **Todėl SNOD projektavimą organizacijos darbuotojų jėgomis galima laikyti organizacijos vykdomosios funkcinės struktūros deskriptyvinio modelio kūrimo pagrindine sudėtine dalimi.** Sudarius SNOD, makroprojektuotojams belieka, kaip jau buvo minėta, tik suvesti į kompiuterį tolimesniam projektavimui reikalingą deskriptyvinę informaciją, tuo užbaigiant sistemos struktūros

deskriptyvinio modelio sudarymą.

Kaip jau buvo minėta šio skyrelio pradžioje, nė vienas iš funkcinės struktūros projektavimo metodų – nei normatyvinis, nei deskriptyvinis – negali garantuoti, kad, projektuojant bei tobulinant sudėtingos organizacijos struktūrą, nebus padaryta kokių nors klaidų. Naudodamiesi normatyviniu metodu, gauname gana racionalios, logiškai pagrįstos ir darnios, tačiau gyvenimo praktikos dar nepatikrintos vykdomosios funkcinės struktūros projektą. Naudodamiesi deskriptyviniu metodu, gauname, galbūt, ne tokią racionalią, tačiau vis dėlto sugebančią funkcionuoti realiomis sąlygomis vykdomąją struktūrą. **Norėdami suprojektuoti pakankamai tobulą naujos organizacijos ar patobulinti jau veikiančios organizacijos funkcinę struktūrą, tuo pačiu „neatitrūkdami nuo žemės“ ir „nepriskaldydami malkų“, galime pasinaudoti normatyvinio ir deskriptyvinio projektavimo metodų bei normatyvinio ir deskriptyvinio modelių sinteze. Lygindami abu modelius ir apmąstydami pastebėtų skirtumų priežastis bei pasekmes, mes galime sudaryti projektą tokios vykdomosios struktūros, kuri, būdama racialesnė už esančiąją, tenkintų realaus gyvenimo reikalavimus, t.y. galėtų būti gana nesunkiai realizuota praktikoje.**

Organizacijos vykdomosios funkcinės struktūros normatyvinio projekto ir jos deskriptyvinio modelio sintezę, kurios tikslas – gauti pilnesnį ir racialesnį normatyvinį tos struktūros projektą, galima pradėti dar normatyvinio projektavimo ir deskriptyvinio modelio kūrimo metu, nelaukiant, kol bus pilnai užbaigti normatyvinio projektavimo ir deskriptyvinio modeliavimo darbai. Tai ne tik taupo bendrą projektavimo laiką, bet ir padeda kokybiškiau vykdyti normatyvinį projektavimą ir deskriptyvinį modeliavimą, nors, žinoma, galima įsivaizduoti normatyvinį projektavimą ir deskriptyvinį modeliavimą kaip du atskirai vykstančius procesus, kurių rezultatai sintezuojami, tik tiems procesams pasibaigus.

Funkcinės struktūros normatyvinio ir deskriptyvinio modelių sintezė pradedama nuo tų modelių palyginimo.

Kai kokia nors funkcija yra normatyviniame modelyje, bet jos nėra deskriptyviniame funkcinės struktūros modelyje, tai reikia išsiaiškinti, kokio tikslo bei potikslio pasiekimui reikalinga ta funkcija ir kodėl organizacijoje dabar ta funkcija nevykdoma. Besiaiškinant galima gauti vieną iš dviejų išvadų: 1) arba pasirodo, kad nagrinėjamoji funkcija tikrai reikalinga ir todėl tikslinga patobulinti veikiančią funkcinę struktūrą, įvedant tos funkcijos vykdymą, 2) arba pasirodo, kad, beprojektuojant normatyvinį funkcinės struktūros modelį, buvo „persitengta“ ir į projektą buvo įtraukta potikslų ir funkcijų daugiau, negu reikia, o todėl būtina tobulinti normatyvinį modelį, eliminuojant bereikalingus potikslus bei funkcijas. Pirmuoju atveju nagrinėjamoji funkcija įtraukiama į sintezuojamą patikslintą funkcinės struktūros projektą, antruoju – neįtraukiama.

Analogiškai elgiamasi ir tais atvejais, kai kokia nors funkcija randama deskriptyviniame modelyje, bet nerandama normatyviniame. Kiekvienu tokiu atveju išsiaiškinama, ar ta funkcija tikrai reikalinga ir todėl reikia tobulinti normatyvinį funkcinės struktūros modelį, papildant jį ir įtraukiant tą funkciją į sintezuojamą patikslintą projektą, ar ta funkcija nereikalinga ir todėl ją reikia eliminuoti iš deskriptyvinio modelio, supaprastinant veikiančią sistemą ir neįvedant į sintezuojamą patikslintą funkcinės struktūros modelį bereikalingų funkcijų.

Normatyvinio ir deskriptyvinio funkcinės struktūros modelių lyginimas nėra toks paprastas darbas, kaip galėtų iš pirmo žvilgsnio atrodyti, nes ne taip jau paprasta nustatyti normatyvinio ir deskriptyvinio modelių funkcijų tapatumą bei skirtumą. Tik išnagrinėjus funkcijų vykdymo algoritmus, galima nuspręsti, sutampa ar nesutampa funkcijos, turinčios vienodus ar analogiškus pavadinimus, ar nėra sutampančių funkcijų, kurios normatyviniame ir deskriptyviniame modeliuose vadinamos skirtingai. Todėl normatyvinio ir deskriptyvinio funkcinės struktūros modelių lyginimą ir derinimą tikslinga pradėti „nuo apačios“, t.y. nuo elementarių funkcijų. Jas palyginus ir suderinus, jau galima pereiti ir prie sudėtingesnių funkcijų lyginimo bei derinimo, lyginant bei derinant tų funkcijų sudėtinės dalis.

Belyginant normatyvinį ir deskriptyvinį modelius ir benagrinėjant funkcijų vykdymo algoritmus, neretai išaiškėja kai kurie esminiai tu modelių trūkumai ir todėl tenka grįžti (dažnai net ne vieną kartą) prie deskriptyvinio ir normatyvinio modelių sudarymo, papildomos informacijos rinkimo ir tų modelių tobulinimo. Tokie grįžimai yra būdingi sudėtingų sistemų struktūrų projektavimui, apie ką jau buvo kalbėta penktajame šios metodinės priemonės skyrelyje, aprašant iteracinius procesus modeliavime ir projektavime bei formuluojant iteratyvumo principą (žiūr. 29 psl.).

Normatyvinio ir deskriptyvinio vykdomosios funkcinės struktūros modelių lyginimo ir derinimo loginio etapo pabaigoje gaunamas abiejų tų modelių sintezės rezultatas – **patikslintas ir suderintas sudėtingos organizacinės sistemos vykdomosios funkcinės struktūros projektas.**

Net po tokio sudėtingo proceso, kaip normatyvinio ir deskriptyvinio modelių sudarymas ir tų modelių lyginimas bei derinimas, **dar negalima garantuoti, jog patikslintas ir suderintas sudėtingos organizacinės sistemos vykdomosios funkcinės struktūros projektas yra pakankamai tobulas ir neturi klaidų. Galimos klaidos bus dar „gaudomos“, lyginant funkcinės struktūros projektą su objekciniu sudėtingos organizacijos sudėties ir struktūros projektu.** Tai bus aprašyta devintame šios priemonės skyrelyje. Tačiau jau šiame etape su didele tikimybe galima teigti, **jog suprojektuoti vykdomoji funkcinė struktūra yra ne blogesnė (o dažnai ir žymiai geresnė) už tą funkcinę struktūrą, kurią turėjo organizacija iki projektavimo pradžios.**

Aukščiau aprašytu būdu suprojektuotos vykdomosios funkcinės struktūros kokybė labai priklauso nuo normatyvinio ir deskriptyvinio modelių detalumo, pilnumo ir racionalumo. Jei kokia nors reikalinga funkcija dėl normatyvinio modelio projektuotojų neįžvalgumo ar aplaidumo bei dėl veikiančios sistemos struktūros analizuotojų neatsakingumo ar paviršutiniškumo būtų praleista abiejuose modeliuose, tai ta funkcija negalėtų nei iš šio, nei iš to atsirasti patikslintame ir suderintame funkcinės struktūros projekte, gautame, jungiant normatyvinį ir deskriptyvinį modelius. Iš kitos pusės, jei kokia nors nereikalinga funkcija patektų į normatyvinį ir deskriptyvinį modelius, tai ji neišvengiamai patektų ir į tų modelių jungimo rezultatą – patikslintą funkcinės struktūros projektą.

Šių eilučių skaitytojui, pastebėjusiam, koks sudėtingas funkcinės struktūros projektavimo procesas siūlomas šioje metodinėje priemonėje, gali kilti klausimai: **„Ar reikia taip jau bijoti kokių nors reikalingų funkcijų praleidimo kuriamame projekte ir bereikalingų funkcijų atsiradimo bei palikimo jame? Ar apsimoka įdėti tiek daug darbo į dviejų modelių – normatyvinio ir deskriptyvinio – kūrimą ir po to dar – į tų dviejų modelių sintezę? Ar nebūtų verta, taupant projektavimo laiką ir lėšas, nebūti tokiais skrupulingais ir apsiriboti koku nors paprastesniu projektavimo metodu?“**

Atsakant į pirmąjį šių klausimų, būtina pažymėti, jog, kuriant sudėtingos organizacijos kompiuterizuotą informacinę (ir valdymo) sistemą, funkcinės struktūros **pilnumo problema žymiai paaštrėja: „rankinėje“ sistemoje žmogus, vykdydamas savo pareigas ir žinodamas tuos tikslus, kurių siekia jis pats ir visa organizacija, gali pastebėti kokio nors iš anksto neplanuoto darbo reikalingumą ir jį padaryti net tada, kai toks darbas nenumatytas nei SNOD, nei TNOD, nei PNOD, tačiau kompiuteriai, vykdydami į jų atmintį patalpintas programas, veiks tiksliai pagal tas programas ir jokių papildomų neplanuotų, bet reikalingų darbų nepastebės ir nevykdys, nors nuo to galėtų sutrikti visos sistemos veikimas ir įvykti kokia nors stambi avarija, panašiai kaip nuo vieno mažo sraigtelio gedimo lėktuvo variklyje gali ištikti aviakatastrofa.**

Analogiška situacija susidaro ir bereikalingų funkcijų, sudarančių „biurokratinį balastą“, nuo kurio kartkartėmis reikia apsivalyti, atžvilgiu: **„rankinėje“ sistemoje anksčiau ar vėliau atsiranda žmogus, kuris pastebi, jog vykdomas tas ar kitas bereikalingas darbas, ir jį eliminuoja iš savo ir visos organizacijos veiklos; kompiuterinėje sistemoje tai pastebėti žymiai sunkiau, nes mažai kam „skauda galvą“, kaip kompiuteris gauna**

reikalingus rezultatus, svarbu tik, kad gauna. Net jei ir atsirastų toks žmogus, kuris, turėdamas laisvo laiko, ryžtųsi pasidomėti, ar visa, ką daro kompiuterinė sistema, yra tikrai reikalinga, jam patenkinti savo smalsumą būtų ne taip jau paprasta: susigaudyti kieno nors sukurtoje programinėje įrangoje ir surasti joje nereikalingus modulius neretai būna net sudėtingiau, negu pačiam iš naujo parašyti visas programas.

Todėl, automatizuojant organizacijos veiklą, būtina tą veiklą labai detalčiai ir labai pilnai aprašyti, nepraleidžiant nė vienos reikalingos operacijos, po to tą veiklą racionalizuoti, atsižvelgiant į naujas galimybes, kurios atsiranda, naudojantis kompiuteriais ir matematiniais skaičiavimo, modeliavimo bei optimizavimo metodais, ir tik po to pradėti projektuoti automatizuotos sistemos informacinę, programinę ir techninę įrangą. **Klaidos, padarytos, projektuojant sudėtingos organizacijos funkcinę struktūrą ir funkcijų vykdymo technologijas, kuriant automatizuotą sistemą, gali labai brangiai kainuoti. Brangiai kainuos ir jų ištaisymas po to, kai visa techninė, informacinė ir programinė įranga bus sukurta ir įdiegta, kai organizacijos personalas bus jau apmokytas ta įranga naudotis.**

Daug pigiau tas klaidas pastebėti ir ištaisyti projektavimo metu, negu, taupant, projektavimo laiką ir mažinant jo išlaidas, tas klaidas palikti projekte, o po to taisyti jas jau pradėjusioje veikti sistemoje. Šią tiesą jau ne kartą patvirtino sudėtingų sistemų projektavimo ir diegimo praktika, ir tai duoda atsakymą ne tik į pirmąjį, bet ir į antrąjį bei trečiąjį mūsų išsivaizduojamo (virtualinio) skaitytojo virtualinius klausimus. Biečiau tik pridėti, jog klaidų atsiradimas, projektuojant sudėtingą sistemą, kaip jau buvo minėta penktajame šios priemonės skyrelyje, yra pilnai objektyvus ir beveik neišvengiamas reiškinys, išplaukiantis ir iš projektuojamos sistemos neapžvelgiamumo vieno žmogaus intelektu ir iš jos projektavimo proceso sudėtingumo. Todėl ir tenka griebtis tokių specifinių sisteminio projektavimo priemonių, leidžiančių sumažinti galimų projektavimo klaidų skaičių, kaip įvairių projektavimo ir modeliavimo metodų sintezė ir daugkartinės iteracijos projektavimo procese. Manau, kad čia išdėstyti argumentai pakankamai paaiškina, kodėl ši metodinė priemonė rekomenduoja tokį sudėtingą vykdomosios funkcinės struktūros projektavimo procesą. Tie, kurie, išsigandę tokio sudėtingumo, bandys projektuoti „paprasciau, pigiau ir greičiau“, gana greitai įsitikins, jog teisinga senų žmonių patarlė: „Šykštus du kartus moka“. Šių eilučių autorius prieš kokius keturiasdešimt metų, būdamas dar pakankamai jaunas ir nepatyręs, irgi bandė panašiai „taupyti“ ir „savo kailiu“ patyrė tokio „taupymo“ pragaištingumą. Ilgainiui atėjo supratimas, jog žymiai pigiau yra mokytis iš svetimų klaidų, negu iš savų. Tačiau gyvenimo praktika rodo, jog daugumai žmonių tas supratimas ateina tik po to, kai jie padaro pakankamai didelį nuosavų klaidų skaičių. Todėl šių eilučių autorius nelabai nustebęs, jei ir SIS kūrėjų tarpe atsiras tokių, kurie nepatikėję, jog sudėtingumo dėsnis yra toks pat „nepermaldaujamas“ kaip ir energijos tvarumo dėsnis, ryšis „pergudrauti gyvenimą“. Kadangi tokiuose „bandymuose“ bus rizikuojama ne savais, o valdiškais pinigais, tai gana tikėtina, kad tokie „bandymai“, neįsiklausius į šių eilučių autoriaus perspėjimus ar nepatikėjus jais, vis dėlto bus daromi. Gali būti, kad SIS projektavimo pradžioje nebus ryžtasi pradėti valstybinės statistinės sistemos struktūros pertvarkymą, pritaikant ją prie naujų sąlygų, o taip pat nebus ryžtasi kardinaliai perprojektuoti turimą SNOD, įtraukiant į tą perprojektavimą statistinės sistemos vadovus ir darbuotojus. Gana tikėtina, jog bus bandoma apsiriboti tik duomenų bankų ir jų panaudojimui reikalingos programinės ir techninės įrangos projektavimu, vykdant tą projektavimą ir projektų diegimą „greitosiomis“, gerai neįsigilinus į tai, kokia turėtų būti statistinė sistema artimoje ateityje, ką toje sistemoje turėtų vykdyti kompiuteriai, o ką ir kaip – žmonės. Sukūrus bendrąją duomenų bankų sistemą (ko tikrai reikia ir ką jau seniai reikėjo padaryti) ir kai kuriuos programinės įrangos fragmentus bei papildžius turimų kompiuterių tinklą, matyt, bus „atraportuota“ apie SIS sukūrimą, nors jau pačioje suprojektuotos „sistemos“ diegimo pradžioje paaiškės, jog daug kas padaryta ne taip ir daug ką reikia tuoj pat keisti. Prasidės sukurtos „sistemos“ „lopymas“, kurio metu paaiškės, kad jam galo nėra, tuo labiau, kad iki to laiko gyvenimas privers daug ką pakeisti statistinės sistemos struktūroje ir statistinių

duomenų apdorojime bei statistinės informacijos panaudojime. Tuomet pagaliau bus suprasta, kad sistemą reikia perprojektuoti iš naujo, tačiau dabar dar gana sunku prognozuoti, ar atsiras tokiame perprojektavime reikalingų lėšų ir entuziazmo. Neužilgo taip pat bus įsitikinta, kad naujos technikos ir naujos įrangos eksploatavimo efektyvumas labai priklauso nuo darbuotojų sugebėjimo ir noro ta technika ir įranga naudotis, o taip pat nuo organizacijos personalo veiklos organizuotumo, nuo jos sukoordinavimo. Taip palaipsniui bus prieita prie statistinės sistemos personalo kvalifikacijos kėlimo ir atsakomybės didinimo būtinumo bei prie personalo veiklos tobulinimo, projektuojant ir diegiant SNOD, TNOD ir PNOD tikslingumo supratimo. Šių eilučių autorius taip drąsiai prognozuoja būsimųjų įvykių eigą todėl, jog jau ne kartą yra stebėjęs čia aprašomas metamorfozes. **Anksčiau ar vėliau dauguma žmonių, padarę daugiau ar mažiau klaidų, pagaliau įsitikina šioje metodinėje priemonėje aprašomos metodologijos ir jos principų teisingumu, tačiau neretai padarytų klaidų ištaisymui nebelieka nei laiko, nei lėšų, nei jėgų.** Būtų žymiai geriau, jeigu pačioje sudėtingos sistemos kūrimo pradžioje būtų pasistengta viską kuo rimčiau pasverti ir išvengti nors pačių stambiausių dažnai daromų klaidų. Šios metodinės priemonės autorius laikytų savo (kaip specialisto, atsakingo už sisteminės metodologijos taikymą SIS projektavime) misiją įvykdyti, jei jam pavyktų taip paveikti SD ir jo tarnybų vadovus bei SIS projektuotojus, kad ką tik aprašytos liūdnokos prognozės neišsipildytų, kad minėtieji vadovai ir projektuotojai, supratę sisteminės metodologijos esmę, pasinaudotų ja ir pasirinktų šiek tiek ilgesnį ir sudėtingesnį, tačiau žymiai produktyvesnį SIS kūrimo kelią, vedantį visą statistinę sistemą į naują būseną, žymiai labiau atitinkančią Lietuvos valstybės ir jos visuomenės poreikius.

Kalbant apie sudėtingos organizacijos veiklos tobulinimo, ją kompiuterizuojant, sudėtingumą, norėtusi pažymėti, jog ši metodinė priemonė rekomenduoja nors ir sudėtingą, tačiau dar ne patį sudėtingiausią kompiuterizuotos organizacinės sistemos kūrimo kelią. Kai kurie patyrę sisteminiai projektuotojai, atsižvelgdami į didelę sudėtingų sistemų projektavimo klaidų kainą ir norėdami dar labiau sumažinti jų skaičių, naudojami dar sudėtingesniais metodais. Pavyzdžiui, sudarydami organizacijos, kurioje nėra pakankamai tobulos SNOD, TNOD ir PNOD, funkcinės struktūros deskriptyvinį modelį, jie taiko metodą „nuo apačios“ ir „nuo viršaus“ sintezę: viena projektuotojų grupė, bendradarbiaudama su organizacijos ir jos padalinių vadovais, sudarinėja hierarchinę funkcijų ir jų ryšių schemą, pradėdama nuo organizacijos tikslų bei pačių stambiausių visos organizacijos funkcijų, palaipsniui leisdama „žemyn“ iki elementarių funkcijų, kurias vykdo atskiri pareigūnai; kita grupė, bendradarbiaudama su eiliniaisiais pareigūnais, pradeda nuo elementarių funkcijų ir palaipsniui kyla „aukštyn“; po to abi gautos schemas lyginamos, derinamos ir jungiamos į vieną – į deskriptyvinį vykdomosios funkcinės struktūros modelį, kuris mūsų jau aprašytu būdu lyginamas, derinamas ir jungiamas su normatyviniu vykdomosios funkcinės struktūros modeliu į patikslintą vykdomosios funkcinės struktūros projektą.

Baigiant šį – septintąjį – skyrelį, norėtusi atkreipti skaitytojų dėmesį į vieną aplinkybę, dėl kurios nežinojimo galėtų kilti kai kurių neaiškumų, skaitant įvairių autorių parašytas knygas bei straipsnius apie funkcinį ir objektinį organizacinių sistemų projektavimo metodus bei jų sintezę. Kai kurie autoriai šiame skyrelyje aprašytą **funkcinės struktūros transformavimą į vykdomąją struktūrą priskiria ne funkcinės struktūros makroprojektavimo, o funkcinio ir objektinio projektų sintezės loginiam etapui**, kuris bus aprašytas devintame šios priemonės skyrelyje. Toks priskyrimas turi savo „racijos“ ir, teoriškai mąstant, yra teisingas, nes, projektuojant vykdomąją struktūrą, tenka operuoti ne tik su funkcijomis, o tai būdinga funkcinio projektavimo loginiam etapui, bet ir su veiklos subjektais ir objektais, o tai būdinga objektinio projektavimo loginiam etapui. Taip pat tenka priskirinėti funkcijas veiklos subjektams, nustatant ryšius tarp funkcijų ir subjektų bei objektų, o tai būdinga funkcinio ir objektinio projektavimo sintezės loginiam etapui. Tačiau sisteminio projektavimo praktikoje visiškai nesvarbu, kuriam iš minėtųjų trijų loginių projektavimo etapų priskiriamas organizacinės sistemos vykdomosios struktūros formavimas iš jos funkcinės struktūros, nes organizacinės sistemos struktūros projektavimo metu visi tie trys loginiai etapai vyksta kartu, vienas kitą

papildydami, vienas kitam padėdami ir sudarydami **vieningą sisteminio projektavimo procesą, vieną to proceso chronologinį etapą (chronologinę fazę)**. Tame procese ne visada lengva nustatyti, kuriam iš loginių etapų turėtų priklausyti tas ar kitas projektuotojų veiksmas, nes yra tokių veiksmų, kuriuos vykdydami projektuotojai sprendžia iškart kelių loginių etapų uždavinius. Tokie grupėi veiksmų priklauso ir veiksmas, kuriuos vykdo projektuotojai, formuodami organizacijos vykdomąją struktūrą iš jos funkcinės struktūros. Normatyvinio ir deskriptyvinio funkcinės struktūros modelių palyginimą, derinimą ir sintezę į patikslintą funkcinės struktūros projektą žymiai lengviau realizuoti, kai normatyvinis modelis jau turi vykdomosios funkcinės struktūros formą. **Todėl šiame metodinės priemonės skyrelyje ir buvo aprašytas normatyvinio funkcinės struktūros modelio transformavimas į normatyvinį vykdomosios funkcinės struktūros modelį, kurio metu normatyvinis funkcinis modelis būna papildomas tokiu palyginimui reikalingomis valdymo funkcijomis.** Taigi, šio transformavimo metu sprendžiami visų minėtųjų trijų loginių projektavimo etapų uždaviniai: papildomas funkcijų sąrašas, formuojami veiklos subjektai ir nustatomi ryšiai tarp funkcijų ir veiklos subjektų bei objektų. Kadangi, grupuojant funkcijas į paketus, tenka atsižvelgti ne tik į jų funkcinį, bet ir į objektinį, informacinį bei algoritminį giminingumą, tai vykdomosios funkcinės struktūros formavimas neįmanomas, remiantis vien tik funkcinio projektavimo rezultatais ir nesinaudojant objektiniu projektavimu. Todėl apie vykdomąją funkcinę struktūrą ir jos formavimą tenka kalbėti, aprašant kiekvieną organizacijos struktūros ir sudėties projektavimo loginį etapą. Tai ir daroma šioje metodinėje priemonėje.

Sąryšyje su tuo, kas čia buvo pasakyta, norėdusi dar kartą atkreipti skaitytojo dėmesį į **skirtumą tarp loginio ir chronologinio etapų sąvokų**, apie ką jau buvo rašyta šeštajame šios priemonės skyrelyje, aprašant sistemotechninio projektavimo loginius etapus ir chronologines fazes (chronologinius etapus). **Vieno chronologinio etapo metu gali lygiagrečiai vykti keli loginiai projektavimo etapai, o vienas loginis etapas gali būti tęsiamas keliuose chronologiniuose etapuose.** Pavyzdžiui, funkcinio projektavimo loginis etapas apima ne tik funkcinės struktūros ir SNOD projektavimo fazę, bet ir funkcijų vykdymo bei TNOD ir PNOD projektavimo, o po to ir kompiuterizuotos sistemos programinės įrangos taikomųjų programų projektavimo chronologines fazes. Analogiškai objektinio projektavimo loginis etapas apima ne tik organizacinės sistemos sudėties ir jos objektinės struktūros projektavimo fazę, bet ir organizacinės sistemos elementų bei posistemių sąveikavimo procesų projektavimo, o po to ir kompiuterizuotos sistemos informacinės įrangos, tos sistemos funkcionavimui reikalingų informacinių bankų ir jų eksploatavimui reikalingos programinės įrangos projektavimo chronologines fazes.

Gali kilti klausimas, **kodėl, aprašant sudėtingos sistemos sisteminio projektavimo procesus, neapsiribojama vien tik to proceso dekompozicija laiko koordinatės atžvilgiu ir chronologinių projektavimo etapų įvedimu bei aprašymu, kodėl įvedami dar ir loginiai etapai**, kurie gaunami, dekomponuojant sudėtingus (kompleksinius) sistemos sukūrimo bei tobulinimo tikslus ir jų siekimo procesus į sudėtinę dalis.

Taip daroma todėl, kad **loginių etapų įvedimas padeda struktūrizuoti sudėtingą sisteminio projektavimo procesą, išskiriant lygiagrečiai vykstančius paprastesnius procesus, ir leidžia įžvelgti sisteminio projektavimo procese tas trajektorijas, kurios vaizduoja nuosekliai vykdomų ir viena iš kitos logiškai išplaukiančių operacijų sekas.** Loginių etapų išskyrimas leidžia geriau suprasti sisteminio projektavimo esmę, apžvelgti jo sudėtį ir struktūrą, geriau suvokti sisteminio projektavimo logiką, kurią kartais ne taip jau lengva įžvelgti ir suvokti, kai nagrinėjami ir aprašomi iškart visi vienu metu vykstantys sisteminio projektavimo darbai. Tiesiog **etapai todėl ir vadinami loginiais, kad atitinka projektavimo darbų loginę seką, kai tuo tarpu chronologiniai etapai vaizduoja visų lygiagrečiai vykstančių projektavimo darbų išsidėstymą laike.** Aprašant sudėtingus loginius etapus, kartais būna tikslinga dekomponuoti juos į smulkesnius. Pavyzdžiui, sudėtingos organizacinės sistemos struktūros ir sudėties projektavimo procese galima išskirti tris loginius etapus: funkcinio projektavimo, objektinio projektavimo ir abiem metodais

gaunamų rezultatų sintezės etapą, kurio metu nustatomi ryšiai tarp esybių ir funkcijų. Funkcinio projektavimo loginį etapą galima savo eile dekomponuoti vėl į tris loginius etapus: normatyvinio projektavimo, deskriptyvinio modeliavimo ir abiejų šių paprocesių bei jų rezultatų sintezės loginį etapą. Į analogiškus etapus galima dekomponuoti ir objektinio projektavimo loginį etapą. Čia išvardinti loginiai projektavimo etapai reikalingi, projektuojant ne tik organizacinės sistemos struktūrą ir sudėtį bei jos SNOD, bet ir organizacijoje vykstančius procesus bei juos reglamentuojančią TNOD ir PNOD.

Kadangi loginiai projektavimo etapai yra viena iš kitos logiškai išplaukiančių projektavimo darbų sekos, tai, **pažeidus tų etapų darbų tvarką, pažeidžiama sisteminio projektavimo logika**. Tokio pažeidimo pavyzdžiu gali būti bandymas pradėti projektavimo darbus ne nuo loginių etapų pradžios, o nuo tuos etapus vaizduojančių trajektorijų kokių nors vidinių taškų, tarkim, bandymas suprojektuoti sudėtingos organizacinės sistemos funkcionavimui reikalingą informacinę ir programinę įrangą, nesudarius tos sistemos funkcionavimą aprašančių modelių. Darydami tokį bandymą, gautume jau **ne sisteminį, o fragmentinį projektavimą** su visais būdingais jam trūkumais ir neišvengiamomis klaidomis, už kurias vėliau tektų brangiai mokėti.

Tuo ir baigsime sudėtingos organizacinės sistemos struktūros funkcinio projektavimo loginio etapo aprašymą ir pereisime prie tos sistemos sudėties ir struktūros objektinio projektavimo loginio etapo nagrinėjimo.

8. OBJEKTINIS SUDĖTINGOS ORGANIZACINĖS SISTEMOS SUDĖTIES IR STRUKTŪROS PROJEKTAVIMAS

Objektinis projektavimas kaip procesas savo chronologinėmis fazėmis yra daug kuo panašus į funkcinį projektavimą, tačiau skiriasi nuo jau aprašyto funkcinio projektavimo visų pirma tais pagrindiniais elementais, kuriais operuojama projektavimo metu: **funkcinio projektavimo pagrindiniai elementai – organizacijoje vykdomos funkcijos, o objektinio – organizacijoje egzistuojančios ir jos veikloje dalyvaujančios bei naudojamos esybės su savo savybėmis (atributais) ir tarpusavio santykiais (įvairių tipų ryšiais).**

Todėl objektinio projektavimo aprašymą pradėsiu nuo jo **pagrindinių elementų apibrėžimų bei aprašymo**.

Esybės gali būti konkrečios ir abstrakčios. Kompiuterizuojant organizacijų veiklą, tenka projektuoti ne tik pirmojo, bet ir antrojo tipo sistemas (žiūr. 5–6 psl.), operuojant abiejų tipų esybėmis.

Konkrečiomis esybėmis vadinami daiktai, egzistuojantys erdvėje ir laike, t.y. realūs objektai, kurių savybės mes galime pastebėti, naudodamiesi jutimo organais ir įvairiais registravimo bei matavimo instrumentais.

Iš turimų konkrečių esybių, naudojantis dekompozicijos (skaidymo) ir sintezės (jungimo) operacijomis, gali būti konstruojamos naujos konkrečios esybės. Konkrečios esybės, kurių skaidymas į smulkesnes duotuoju momentu laikomas netikslingu ar negalimu, vadinamos **elementariomis esybėmis**. Elementarios esybės sąvoka yra reliatyvi, nes viena ir ta pati esybė vienoje situacijoje gali būti traktuojama kaip elementari, kitoje – kaip **sudėtinė**.

Abstrakčios esybės gaunamos iš konkrečių esybių, jų atributų ir santykių tarp konkrečių esybių bei atributų, naudojantis tokiomis loginėmis (mąstymo) operacijomis kaip abstrahavimas, apibendrinimas ir abstrahavimo rezultatų loginė sintezė.

Abstrahavimas – tai **mąstymo operacija, kurios metu iš kokios nors vientisos visumos kas nors išskiriama** (ištraukiama), atskiriama (atitraukiama) ir tas „kas nors“ mąstomas kaip nauja savarankiška sąvoka – abstrakti esybė. Abstrahavimas gali būti atributinis, reliacinis, apibendrinantis (kartais vadinamas tiesiog apibendrinimu) ir modelinis (metamodelinis).

Atributiniu abstrahavimu vadinamas toks abstrahavimas, kuomet nuo kokios nors esybės „atitraukiama“ kokia nors jos savybė ir ta savybė mąstoma kaip atskira savarankiška

sąvoka, kaip abstrakti esybė, reiškianti minėtos savybės esmę. Pavyzdžiui, galima mąstyti ne tik apie rutulį bet ir apie rutulio apvalumą ir lyginti jį su elipsoido ar kiaušinio apvalumu; taip pat galima mąstyti apie kreidos baltumą bei lyginti jį su sniego baltumu, nors pats apvalumas negali egzistuoti atskirai nuo rutulio ar elipsoido, o baltumas – atskirai nuo kreidos ar sniego.

Reliaciniu abstrahavimu vadinamas toks abstrahavimas, kada „atitraukiamas“ koks nors santykis, egzistuojantis tarp kokių nors esybių bei jų savybių, ir tas santykis mąstomas kaip atskira savarankiška sąvoka, kaip abstrakti esybė. Pavyzdžiui, galima mąstyti apie kokių nors žmonių giminybę ar draugystę, apie kreidos ir sniego baltumų bei rutulio ir kiaušinio formų skirtumus, apie tai, kad trys kilogramai cukraus yra daugiau, negu du kilogramai, įsivedant tokią sąvoką, kaip „daugiau“, žymimą ženklu „>“.

Apibendrinančiu abstrahavimu vadinamas toks abstrahavimas, kada iš daugelio esybių ar jų savybių bei santykių išskiriama tai, kas tose esybėse, jų savybėse ar santykiuose yra bendra, ir šitas „tai“ mąstomas kaip atskira nauja sąvoka, kaip nauja, abstrakti esybė. Pavyzdžiui, galima mąstyti apie tokią abstrakčią bendrąją esybę kaip „spalva“, jungiančią tokias esybes, kaip „sniego baltumas“, „kraujo raudonumas“, „dangaus žydrumas“ ir t.t. Apibendrinančio abstrahavimo rezultatas yra tokios bendrosios abstrakčios sąvokos, kaip „skonis“, „jausmas“, „objektas“, „sistema“, „būsena“, „rodiklis“, „failas“, „modelis“, „savybė“, „santykis“, „esybė“, „sąvoka“ ir pan.

Modeliniu abstrahavimu vadinamas toks abstrahavimas, kuomet nuo kokio nors stebimo objekto „atitraukiamas“ jo koncepcinis modelis ir tas modelis mąstomas kaip atskirai nuo to objekto egzistuojanti abstrakti esybė. Tai esybei galima suteikti tą ar kitą išraišką (formą). Taip atsiranda įvairūs „fiziniai“ ir informaciniai (tame tarpe matematiniai, grafiniai, verbaliniai) sistemų ir procesų modeliai, įvairūs tų modelių tipai (žiūr. ketvirtąją šios priemonės skyrelį). Modelius galima vartoti kaip tikrovės aprašymo ir analizės priemonę, tačiau galima analizuoti ir pačius modelius kaip savarankiškas esybes, susidarant jų koncepcinius modelius, vadinamus modeliuojamų objektų bei procesų **metamodeliais**. Mąstydami apie objekto ar proceso modelį kaip apie atskirą savarankišką esybę, mes sukuriame savo galvoje koncepcinį modelio modelį, būtent, to objekto ar proceso metamodelį – **metamodelinio abstrahavimo rezultatą**. Modeliuoti galima ne tik tai, kas jau egzistuoja, bet ir tai, kas galėtų egzistuoti. Tokių modelių abstrahavimas – taip pat abstrakčių esybių kūrimo šaltinis.

Abstrakčių esybių **loginės sintezės** operacija vadinama tokia **mąstymo operacija, kurios metu, jungiant abstrakčias esybes, konstruojama nauja abstrakti esybė**. Ši loginė operacija analogiška **konkrečių esybių fizinės sintezės operacijai** ir skiriasi nuo pastarosios tik tuo, kad loginės sintezės operacijos operandai ir rezultatas yra abstrakčios esybės, o konkrečių esybių fizinės sintezės operacijos operandai ir rezultatas – konkrečios esybės.

Esybės, reikalingos **objektinio deskriptyvinio modelio** sudarymui, išskiriamos, nagrinėjant tobulinamos (pavyzdžiui, kompiuterizuojamos) organizacijos veiklą, nustatant tos veiklos subjektus, objektus, priemones, aplinkybes, rezultatus ir t.t. Kai projektuojama nauja organizacija, objektiniam projektavimui reikalingos esybės gali būti išskirtos, nagrinėjant analogiškos projektuojamai, (tačiau jau veikiančios) organizacijos veiklą. Kai analogiškų organizacijų nėra, tai objektinį deskriptyvinį modelį sudaryti neįmanoma ir tenka apsiriboti tik **objektinio normatyvinio modelio** sudarymu. Tokio modelio sudarymui reikalingos esybės suformuojamos (nustatomos), analizuojant veiklą tos įsivaizduojamos organizacijos, kuri sugebėtų realizuoti visas normatyviniame funkciname modelyje numatytas funkcijas. Todėl objektinio normatyvinio modelio sudarymas gali būti pradėtas tik po to, kai jau sudarytas **funkcinis normatyvinis modelis**. Normatyvinio objektinio projektavimo procesas gali vykti kartu su funkcinės normatyvinės struktūros transformavimu į vykdomąją funkcinę struktūrą, projektuojant kartu normatyvinę SNOD ir normatyvinę TNOD bei PNOD. Objektinio deskriptyvinio modelio sudarymas gali būti pradėtas nepriklausomai nuo funkcinio deskriptyvinio ir normatyvinio modelių sudarymo, tačiau gana

dažnai objektinio deskriptyvinio modelio sudarymas vykdomas lygiagrečiai su **funkcinio deskriptyvinio modelio** sudarymu, analizuojant funkcionuojančios organizacijos veiklą. Abu deskriptyviniai modeliai vienas kitą papildo, todėl tų modelių sintezės metu, apie ką bus rašoma devintajame šios priemonės skyrelyje, gaunamas pilnesnis ir tikslesnis deskriptyvinis organizacijos modelis. Tobulinant veikiančią sudėtingą organizaciją, **objektinio projektavimo** metu tikslinga naudotis **abiejų objektinių modelių (deskriptyvinio ir normatyvinio) sudarymu ir jų sinteze**, panašiai, kaip tai daroma **funkcinio projektavimo** metu (žiūr. septintąją šios priemonės skyrelį). **Šiuo atžvilgiu objektinis projektavimas yra panašus į funkcinį, nes abu šie projektavimo procesai vykdomi, vadovaujantis sistetine projektavimo metodologija, jos principais** (žiūr. šios mokymo priemonės penktąją skyrelį).

Kadangi sudėtingos organizacinės sistemos dekompozicija į posistemius ir elementus gali būti atliekama įvairiai (įvairiais pjūviais, pasirenkant tuos ar kitus dekomponavimo būdus), tai ir išskiriamų esybių sąrašai, kuriuos gali sudaryti skirtingi sisteminiai analitikai, analizuodami vieną ir tą pačią sistemą, gali būti nevienodi. Tai yra natūralu, kaip yra natūralu, kad skirtingi žmonės vieną ir tą patį reiškinį gali aprašyti ne visai vienodai, naudodamiesi ne visada tais pačiais žodžiais.

Tačiau panašiai kaip aprašymo kokybė priklauso nuo pasirinktų žodžių, taip ir objektinių modelių pilnumas bei tikslumas priklauso nuo to, kokiomis esybėmis bus operuojama objektinio projektavimo bei modeliavimo metu.

Sistemų teorija ir sisteminė metodologija rekomenduoja, kad sisteminiai analitikai, analizuodami veikiančias ir projektuodami naujas sudėtingas sistemas, stengtųsi naudotis tik tokiomis esybėmis, kurios tenkina objektyvumo, esmiškumo, interpretuojamumo, išmatuojamumo, operacionalumo ir praktinio patikrinamumo reikalavimus. Sisteminiai analitikai turėtų vengti tokių esybių, kurios netenkina kurio nors iš minėtų reikalavimų.

Sakoma, jog esybė tenkina **objektyvumo** reikalavimą, jei ji arba pati objektyviai egzistuoja (ar galėtų objektyviai egzistuoti), arba jei ji, būdama abstrakčia esybe, atspindi savo turiniu kažką tokio, kas objektyviai egzistuoja (ar galėtų egzistuoti): kitas objektyvias esybes, objektyvias jų savybes, objektyvius santykius tarp esybių bei jų savybių ir pan.

Esybė tenkina **esmiškumo** reikalavimą, jei ji arba pati yra pakankamai svarbi toje srityje, toje veikloje ar toje situacijoje, kuriose ji yra naudojama, arba atspindi savo turiniu kažką svarbaus, esminio: kokią nors kitą esminę esybę, esminę kokios nors esminės esybės savybę, esminį santykį tarp kokių nors esminių esybių bei esminių jų savybių, kokią nors esminį aspektą, leidžiantį pastebėti ką nors naujo ir esminio, ir pan. Suprantama, jog **esybės esmiškumas yra reliatyvus: viena ir ta pati esybė vienoje situacijoje gali būti esminė, kitoje – neesminė.** Operavimas esybėmis, netenkinančiomis esmiškumo reikalavimo, tik apsunkina sisteminę analizę, modeliavimą bei projektavimą ir todėl yra ne tik nepageidautinas, bet ir žalingas. Tačiau nuspręsti, kokia esybė analizuojamoje situacijoje yra esminė, kokia – neesminė, ne visada yra paprasta. Todėl kartais apsidraudimo sumetimais gali būti tikslinga pradžioje nagrinėti ir esybes, kurių esmiškumas yra abejotinas.

Nagrinėjant konkrečias esybes, nekyla jų **interpretuojamo bei fizinio vaizdumo problemos**, nes tos esybės vaizduoja pačias save, t.y. „fiziškai apčiuopiamas“ realijas. Tačiau, formuojant ir nagrinėjant abstrakčias esybes, interpretuojamumo problema tampa gana aktuali, nes nuo esybių interpretuojamumo labai priklauso, ar tos abstrakčios esybės bus visų (ir projektuotojų, ir vartotojų) **vienodai suprastos ir teisingai panaudotos.**

Norėdami suformuluoti interpretuojamumo reikalavimą taip, kad pats šis reikalavimas būtų visų vienodai suprantamas, turime pradžioje įsivesti keletą naujų sąvokų, charakterizuojančių abstrakčias sąvokas ir jas atitinkančias abstrakčias esybes.

Įvesdami abstrakčią sąvoką ir norėdami, kad visi ją vienodai suprastų, mes turime ne tik duoti tai sąvokai **pavadinimą** (t.y. žodį bei žodžių grupę ar kitą kurį nors ženklą – identifikatorių, kodą ir pan., kuriuo tą sąvoką žymėsime), bet ir aprašyti tos sąvokos **turinį**, o

taip pat nurodyti jos **apimtį**. Tik tuomet galima tikėtis, jog ir visi kiti žmonės vienodai supras tos **sąvokos prasmę**, vadinamą sąvokos **konceptu**.

Esybės (ir ją atitinkančios, t.y. turinčios tą patį pavadinimą, sąvokos) turinį sudaro tos esybės savybės, todėl norėdami aprašyti esybę, turime sudaryti jos atributų sąrašą.

Sąvokos apimtį sudaro ta elementų aibė, kurią ta sąvoka aprėpia ir žymi savo pavadinimu, t.y. aibė visų tų esybių, kurių kiekviena turi visas savybes, sudarančias tos sąvokos turinį.

Tarp sąvokos turinio ir jos apimties egzistuoja apytikrė atvirkštinė priklausomybė: didinant vieną, mažėja kita ir atvirkščiai. Pavyzdžiui, sąvokos „žmogus“ turinys yra žymiai turtingesnis už sąvokos „gyvūnas“ turinį, nes sąvokos „žmogus“ atributų sąrašas turi visus tuos atributus, kurie įeina į sąvokos „gyvūnas“ atributų sąrašą, ir dar daug papildomų atributų, charakterizuojančių žmogų ir išskiriančių jį iš kitų gyvūnų tarpo. Tačiau sąvokos „žmogus“ apimtis yra žymiai mažesnė už sąvokos „gyvūnas“ apimtį, nes žmonės sudaro tik dalį visų gyvūnų.

Tie elementai, kuriuos apima kokia nors sąvoka, vadinami jos **denotatu**. Šis žodis reiškia tai, kas žymima kokiu nors vardu, ir yra kilęs iš lotyniško žodžio „notatio“, reiškiančio žymėjimą kokiu nors ženklu – „nota“. **Sąvokos denotatas apibrėžia jos apimtį, o sąvokos turinį apibrėžia jos prasmė, jos konceptas.**

Įvedus tokius naujus terminus kaip konceptas, denotatas ir žymėjimas („nota“), galima paaiškinti, ką reiškia abstrakčios esybės interpretuojamumas.

Sakoma, kad **abstrakti esybė tenkina interpretuojamumo bei „fizinio vaizdumo“ reikalavimus, jei ji turi lengvai įsivaizduojamus bei suprantamus denotatus**, t.y. tuos konkrečius objektus, jų konkrečias savybes bei konkrečius santykius, iš kurių, abstrahuojant bei sintezuojant, buvo suformuotas tos esybės konceptas. Lengvai įsivaizduojamų ir suprantamų denotatu nurodymas žymiai palengvina esybės sampratą – jos koncepto, išplaukiančio iš esybės bei jos turinio apibrėžimo (aprašymo) ir intuityvaus suvokimo, – suformavimą ir padeda patikslinti tą sampratą.

Interpretuojamumas yra giminingas objektyvumui, nes abi šios savybės remiasi konkrečia tikrove, tačiau šios sąvokos nesutampa. **Objektyvumas reiškia atitikimą tikrovei, o interpretuojamumas – interpretavimo lengvumą:** intuityvų suprantamumą, abstrakčios sąvokos denotatų ir pačios sąvokos „fizinį vaizdumą“. Objektyvumo ir interpretuojamumo reikalavimai vienas kitą papildo ir padeda formuoti tokias esybes, kurių konceptai būtų objektyvūs ir akivaizdūs. Be to, šių reikalavimų tenkinimas padeda išvengti kai kurių mąstymo klaidų bei „žaidimo“ „tuščių“ (beprasmių) sąvokomis (t.y. sąvokomis, neturinčiomis objektyvių denotatų ir objektyvaus turinio).

Sakoma, kad esybė tenkina **išmatuojamumo** reikalavimą, jei tai, kas sudaro jos turinį ir apimtį, galima išmatuoti (ar kaip nors kitaip kiekybiškai ar kokybiškai įvertinti) jos denotatuose. Žinoma, ne kiekviena esminė esybė gali tenkinti išmatuojamumo reikalavimą, tačiau kuo didesnė dalis esybių, naudojamų sisteminėje analizėje, sisteminiame modeliavime ir sisteminiame projektavime, tenkins šį reikalavimą, tuo lengviau bus taikyti sistemotyroje matematinį aparatą.

Esybė tenkina **operacionalumo** reikalavimą, jei ji išreiškia tai, su kuo galima atlikti tas ar kitas operacijas, tame tarpe ir kiekybinės matematikos bei logines operacijas. Kitaip tariant, esybės operacionalumas reiškia, kad tos esybės konceptas, jos turinys ar jos denotatai gali tapti tų ar kitų operacijų operandais bei rezultatais. Esybių, tenkinančių operacionalumo reikalavimą, žymėjimai (identifikatoriai) gali tapti tų ar kitų algebrų (pavyzdžiui, matricų algebros, reliacinės algebros, struktūrų algebros ir t.t.) ženklais, su kuriais būtų galima vykdyti toms algebroms būdingas formalias operacijas, tame tarpe ir operacijas, realizuojamas kompiuteriuose. Galima kalbėti apie vidinį ir išorinį sąvokų operacionalumą. **Vidiniu sąvokų operacionalumu** vadinamas toks operacionalumas, kai su sąvokomis galima vykdyti logines (ir matematinės) operacijas žmogaus galvoje.

Išoriniu sąvokų operacionalumu vadinamas toks operacionalumas, kai su tomis sąvokomis (bei jų žymėjimais) galima vykdyti logines (ir matematines) operacijas ne tik žmogaus galvoje, bet ir jos išorėje: popieriuje, naudojantis rašymo priemonėmis, formaliais metodais ir galva (pavyzdžiui, dauginant daugiaženklus skaičius), ar kompiuteryje, naudojantis programine ir technine įranga. Išorinis operacionalumas duoda žmogui galimybę palengvinti ir pagreitinti intelektualų darbą, naudojantis išorinėmis priemonėmis, ir net padaryti tai, ko, naudodamasis vien galva, žmogus padaryti nepajėgtų. Tačiau ne kiekviena sąvoka ir ne kiekviena operacija, kurios yra operacionalios vidiniai, šiuo metu yra operacionalios išoriškai, nes ne visas sąvokas pavyksta formalizuoti ir ne viską, kas vyksta žmogaus galvoje, pavyksta išreikšti taip, kad būtų galima realizuoti kompiuteryje. Suprantama, kad ne visos esybės, naudojamos sisteminėje analizėje, sisteminiame modeliavime ir sisteminiame projektavime, tenkina išorinio operacionalumo reikalavimą, tačiau, norint tuose procesuose kuo plačiau taikyti formalius metodus ir kompiuterinę techniką, reikia stengtis pirmoje eilėje naudotis tokiomis esybėmis, kurios būtų išoriškai operacionalios. Derėtų atkreipti skaitytojo dėmesį į tai, jog **operacionalių esybių aibė yra žymiai platesnė, negu ji atrodo tam, kas žino tik klasikinę matematiką**. Šiuolaikinė matematika operuoja ne tik su skaičiais bei su kiekybinės matematikos formulėmis, bet ir su loginėmis formulėmis, su struktūromis ir jas vaizduojančiais grafais bei su organizuotais (struktūrizuotais) informaciniais masyvais, su geometrinėmis figūromis ir įvairiomis dinaminėmis jų konstrukcijomis, vaizduojančiomis sudėtingus procesus, kaip tai daroma, pavyzdžiui, kompiuteriniuose žaidimuose, ir t.t. Todėl **šiuolaikinė matematika žymiai išplečia operacijos ir operacionalumo sąvokas, o kartu su tuo – ir operacionalių esybių aibę**.

Sakoma, jog esybė tenkina **praktinio patikrinamumo** reikalavimą, jei įvairių operacijų su tomis esybėmis rezultatus, o taip pat įvairius vertinimus bei matavimus, atliktus, nagrinėjant tų esybių turinį bei denotatus, galima patikrinti praktikoje, praktiškai įsitikinant tų rezultatų bei vertinimų objektyvumu bei teisingumu (ar neobjektyvumu bei neteisingumu). Tai, kad egzistuoja nemažai esybių, tenkinančių šiuos reikalavimus, kaip tik ir sudaro sąlygas įvairiems testavimams, natūriniais ir matematiniams eksperimentams, plačiai taikomiems sisteminiame projektavime, sisteminiame modeliavime, hipotezių tikrinime ir, bendrai, sisteminėje analizėje. Todėl sisteminiai analitikai, norėdami, jog būtų išvengta neleistinai ilgo kaupimo neišvengiamai atsirandančių klaidų ir jų pasekmių, privalo stengtis naudotis esybėmis, tenkinančiomis praktinio patikrinamumo reikalavimą, kad patys sisteminiai analitikai galėtų kuo dažniau ir kuo pilniau tikrinti savo darbo rezultatus, naudodamiesi specialiais testavimo metodais. Vienas tokių metodų, vadinamas **žingsninės kontrolės ir kontrolinių taškų metodu**, aprašytas šios priemonės septynioliktajame puslapyje paminėtos antrosios knygelės 191÷193 puslapiuose.

Esybių praktinis patikrinamumas ir kontrolės metodų egzistavimas sudaro galimybes sisteminio projektavimo bei modeliavimo eigos tikrinimui paties projektavimo ir modelių kūrimo metu, nelaukiant galutinių rezultatų ir nekaupiant tų procesų eigoje atsirandančių klaidų. Tas galimybes realizuoti tampa žymiai lengviau, kai stengiamasi kurti tokius deskriptyvinius modelius ir taip formuluoti hipotezes, kad jų adekvatumą tikrovei galėtų lengvai patikrinti patys tų modelių bei hipotezių autoriai, o taip pat kai projektuojami tokie projektai, kuriuos lengva testuoti, naudojantis nesudėtingais matematiniais ir natūriniais eksperimentais. **Kuriant kompiuterizuotas sisteminės analizės, sisteminio modeliavimo ir sisteminio projektavimo sistemas, tikslinga sukurti** (kaip sudėtinę tokių automatizuotų sistemų dalį) **pusiau automatiniame režime veikiančią kontrolės sistemą**, reguliariai tikrinančią sisteminių analitikų darbo rezultatus, kaupiamus kompiuterių atmintyje, testuojančią sukurtus modelius bei projektus ir automatiškai signalizuojančią apie aptiktus prieštarigus ar abejotinus jų apdorojimo ir modelių bei projektų testavimo rezultatus. Tokios kompiuterizuotos sistemos žymiai padidina sisteminių analitikų darbo efektyvumą.

Objektiniame projektavime naudojamoms esybėms, norint, kad jos būtų lengvai skiriamos viena nuo kitos ir visų vienodai atpažįstamos bei traktuojamos (suprantamos), turi būti **suteikti pavadinimai** (o kartais dar ir identifikatoriai bei kodai) ir sudaryti **apibrėžimai**, tenkinantys tapatumo (vienareikšmiškumo), o taip pat **eksplikato ir eksplikando atitikimo** (apibrėžiamos sąvokos turinio ir apimties sutapimo su intuityviu tos sąvokos įvaizdžiu, atitinkančiu tos sąvokos apibrėžimą ir pavadinimą) reikalavimus. Vienas iš plačiai paplitusių esybės aprašymo būdų yra visų esminių jos savybių išvardinimas. **Formaliose sistemose** (pavyzdžiui, informaciniuose bankuose) **esybės tiesiog sutapatina su jų esminiu atributu rinkiniais ir apibrėžiamos kaip atributu rinkiniai (failai), turintys esybių pavadinimus ar jų kodus**. Esybės apibrėžiantys atributų rinkiniai gali turėti gana daug atributų, tačiau egzistuoja ir tokios esybės, kurių atributų rinkiniuose yra tik po vieną atributą.

Formalūs sąvokų ir jas atitinkančių esybių apibrėžimai padeda **suvienodinti intuityvų sąvokų suvokimą**, kuris pas įvairius žmones gali būti šiek tiek skirtingas, dėl ko žmonės kartais nepajėgia pakankamai tiksliai suprasti vienas kitą. Be to, **formalūs apibrėžimai reikalingi, bandant kompiuterizuoti intelektualią veiklą, nes šiuolaikiniai kompiuteriai neturi žmonių mąstymui būdingos intuicijos**. Sąvokos, kurios bus vartojamos kompiuteriuose, turi būti formaliai aprašytos, nurodant jų turinį ir apimtį.

Formaliai apibrėžę esybės sąvoką, pavadindami esybę atributų rinkinį, turintį apibrėžiamos esybės vardą, suteikiama taip, kad skirtingi rinkiniai turėtų skirtingus vardus, o vienas ir tas pats rinkinys – vieną ir tą patį vardą, galėtume pareikalauti, kad būtų formaliai apibrėžtos ir atributo bei santykio sąvokos.

Šios sąvokos, būdamos daugumai žmonių intuityviai pakankamai aiškos, formaliai apibrėžiamos gana sudėtingai. Tas sudėtingumas išplaukia iš to, kad tos sąvokos yra abstrakčios ir laikomos **pirminėmis** (o ne išvestinėmis).

Išvestinių sąvokų apibrėžimai konstruojami, palyginti, paprastai: išvestinės sąvokos **nominaliai apibrėžiamos** (aprašomos, išreiškiamos), naudojantis kitomis jau žinomomis (koku nors būdu anksčiau apibrėžtomis) sąvokomis, kurios apibrėžiamų išvestinių sąvokų atžvilgiu laikomos pirminėmis. Pavyzdžiui, nominaliai apibrėždami esybę kaip atributų rinkinį, turintį tos esybės vardą, mes išvestinę sąvoką „esybė“ išreiškiame per sąvokas „atributas“ ir „rinkinys“, kurias „esybės“ sąvokos atžvilgiu šiuo atveju laikome pirminėmis.

Ką laikyti pirminėmis, o ką – išvestinėmis sąvokomis, konstruojant nominalius apibrėžimus, yra, galima sakyti, susitarimo dalykas, nes nominalūs apibrėžimai gali būti „apverčiami“, sukeičiant vietomis išvestinę sąvoką su kuria nors pirmine. Pavyzdžiui, iš aukščiau cituoto nominalaus esybės apibrėžimo galima gauti atributų rinkinio apibrėžimą: atributų rinkinys yra tai, kas skiria vieną esybę nuo kitos, kitaip tariant, atributų rinkinys yra esybės turinys.

Tačiau, kartą pasirinkus pirmines sąvokas, tokio pasirinkimo reikia laikytis ir toliau. Nesilaikant šio reikalavimo, daroma **taip vadinamoji „ydingojo rato“ klaida**. Tuomet galima gauti vietoj nominalių apibrėžimų nieko neapibrėžiančius „tuščius“ loginius ciklus, kuriuose neapibrėžtos sąvokos tampa „apibrėžiamos“ per pačias save.

Kad apibrėžtume išvestines sąvokas, reikia turėti tam tikrą kiekį pačių pirminių sąvokų. Kaip apibrėžti pastarąsias? Aišku, kad joms negali būti taikomas išvestinių sąvokų nominalinio apibrėžimo metodas, nes jau nebėra sąvokų, kurias būtų galima laikyti „pirmesnėmis“ už pačias pirmines.

Konkrečios pirminės sąvokos, atitinkančios konkrečias esybes, gali būti apibrėžtos, parodant konkretų daiktą ar veiksmą (sąvokos denotatą) ir suteikiant tam daiktui ar veiksmui pavadinimą (vardą, identifikatorių, kodą). Tokie apibrėžimai vadinami **ostensiniais**. Analogiškai apibrėžiamos konkrečios savybės ir konkretūs santykiai. Taip, pavyzdžiui, formuojasi pirminės konkrečios sąvokos mažų vaikų sąmonėje, kai jie mokosi kalbėti „gimtąja kalba“.

Kadangi rodomiems konkretiems daiktams, veiksams, savybėms ir santykiams pavadinimus galima suteikti labai įvairiai, tai susidaro galimybė atsirasti įvairioms kalboms,

kuriose tos pačios sąvokos žymimos (vadinamos) skirtingais žodžiais. Ta galimybė žmonės plačiai pasinaudojo senovėje natūralių kalbų formavimosi metu ir tebesinaudoja dabar, kurdami įvairias dirbtines kalbas, pavyzdžiui, tokias kaip esperanto, matematinė kalba bei įvairios algoritminės kalbos.

Abstrakčių pirminių sąvokų apibrėžimui „parodymo metodas“ netinka, nes negalima parodyti tokio konkretaus denotato, kuris tiksliai apibrėžtų abstrakčios sąvokos apimtį ir išreikštų tos sąvokos turinį. Abstrakčios sąvokos denotatą, jos turinį ir apimtį galima tik vienaip ar kitaip iliustruoti, aiškinti ir po to daugiau ar mažiau tiksliai intuityviai suvokti. Tačiau tą intuityvų suvokimą neįmanoma perduoti šiuolaikiniams kompiuteriams, neišreiškus jo formaliai. Kyla gana netriviali problema: „Kaip suformuoti formalų abstrakčios pirminės sąvokos apibrėžimą, „suprantamą“ kompiuteriui?“

Tai galima padaryti, naudojantis taip vadinamu „**aksiomatinio metodu**“. Iš žodžių, žyminčių jau žinomas (tuo ar kitu būdu anksčiau apibrėžtas) sąvokas, ir žodžių, žyminčių tas abstrakčias pirmines sąvokas, kurias norima formaliai apibrėžti, sudaroma loginių funkcijų – predikatų – sistema. Tie predikatai, aprašantys tam tikrus santykius, egzistuojančius tarp sąvokų, žymimų į jų sudėtį įeinančiais žodžiais, vadinami **aksiomomis**. **Į aksiomų sistemą galima žiūrėti ir ja naudotis dvejopai.**

Kai žodžiams, žymintiems apibrėžiamas pirmines abstrakčias sąvokas, suteikiamos tos reikšmės, kurias mes intuityviai priskiriame apibrėžiamoms pirminėms sąvokoms, tai **aksiomų – predikatų sistema pavirsta į aksiomų –teisingų pirminių teiginių sistemą**, leidžiančią išvesti iš jos, naudojantis dedukcija, antrinius teisingus teiginius – **teoremas**.

Kai apriori priimama, jog aksiomos – predikatai yra teisingi teiginiai, ir kai apibrėžtinas sąvokas žymintys žodžiai laikomi tų predikatų (loginių funkcijų) argumentais, kurių reikšmės reikia nustatyti taip, kad jos tenkintų predikatinę lygčių sistemą, **tai ta aksiomų – loginių lygčių – sistema tampa abstrakčių pirminių sąvokų – loginių funkcijų argumentų reikšmių – netiesioginiu formaliu apibrėžimu**. Kad ši loginių lygčių sistema turėtų sprendinį, reikia, kad aksiomų sistema būtų neprieštaringa. Kad formalus pirminių abstrakčių sąvokų apibrėžimas būtų vienareikšmiškas, reikia, jog nepriklausomų aksiomų būtų pakankamai daug, t.y. tiek, kad loginių lygčių sistema turėtų tik vieną sprendinį. Tokie netiesioginiai pirminių abstrakčių sąvokų apibrėžimai vadinami **implicitiniais** (arba vidiniais, indukciniiais), o antrinių sąvokų apibrėžimai per pirmines – **eksplicitiniais** (arba išoriniais, dedukciniais).

Kaip matome, abstrakčių pirminių sąvokų formalus implicitinis apibrėžimas tikrai yra gana sudėtingas, todėl jis naudojamas tik sistemose, turinčiose gana aukštą formalizavimo bei automatizavimo lygį, pavyzdžiui, formaliose aksiomatinėse teorijose ir kai kuriose dirbtinio intelekto sistemose.

Automatizuotose projektavimo sistemose, veikiančiose žmogaus dialogo su kompiuteriu režime, pirminių abstrakčių sąvokų loginiais formalaus apibrėžimo metodais nesinaudojama. Tokiose sistemose pirminių sąvokų suvokimas gali būti paliktas žmogui. Žmogus atpažįsta pirmines sąvokas (t.y. nustato „kas yra kas“), ir informaciją, vaizduojančią bei aprašančią tas sąvokas (pavyzdžiui, kokios nors esybės atributus), įrašo į iš anksto žmogaus paskirtas kompiuteryje saugomo informacinio banko vietas – „pozicijas“, turinčias vienareikšmiškai joms skirtus vardus – „adresus“. Reikalui esant, naudodamasis tais vardais, žmogus gali iš tų „pozicijų“ gauti jam reikalingą informaciją. Tai reiškia, kad, bendraujant su kompiuteriu dialogo režime, **abstrakčią pirminę sąvoką galima „formaliai apibrėžti“, naudojantis ne tik loginiais, bet ir „poziciniais“ apibrėžimais**, „suprantamais“ kompiuteriui. Tokių „pozicinių apibrėžimų“ esmę galima išreikšti taip: „Tam tikra sąvoka formaliai yra ta informacija, kuri užrašyta informaciniame banke tam tikroje specialiai skirtoje vietoje“. „Poziciniu“ formalaus apibrėžimo metodu dažniausiai ir naudojamosi kompiuterizuotose projektavimo sistemose, „apibrėžiant“ esybes, atributus bei santykius kaip tam tikras vietas kompiuterio atmintyje, kuriose saugomi tų sąvokų turinį simbolizuojantys kodai, ir paliekant žmogui sąvokų atpažinimo, informacijos apie

jas surinkimo ir įrašymo į tam skirtas vietas kompiuterio atmintyje funkcijas.

Aprašant **organizacijos objektinę struktūrą grafinėmis priemonėmis**, esybė vaizduojama stačiakampiu (kartais stačiakampiu su suapvalintais kampais), kurio viduje užrašomas esybės pavadinimas bei kodas ir išvardijami tos esybės esminiai atributai. Santykiai tarp esybių vaizduojami minėtus stačiakampius jungiančiomis linijomis, šalia kurių žodžiais bei sutartiniais ženklais išreiškiama tų santykių esmė. Tokias schemas, vadinamas **esybių ryšių diagramomis (ERD)**, gali braižyti projektuotojai, grafiškai vaizduodami veikiančios organizacijos analizės bei projektavimo rezultatus, o taip pat ir kompiuteriai, apdorodami įvestą informaciją apie organizacijos objektinę struktūrą. Automatizuotose projektavimo sistemose ERD braižo kompiuteriai, palengvindami sisteminiams analitikams surinktos informacijos sistematizavimą, apžvelgimą ir analizę, nes schema, vaizduojančią esybių ryšius (informacinius, pavaldumo, priklausomybės ir t.t.) grafiškai, lydimą tas esybes, jų atributus ir ryšius aprašančio teksto, žmogui analizuoti žymiai patogiau, negu vien tik žodinių tekstą be schemos. Schemos ir žodiniai tekstai vieni kitus papildo: schema duoda taip vadinamą „**integralinį vaizdą**“, padėdama žmogui iškart aprėpti visą visumą, o tekstas padeda išsiaiškinti to vaizdo detales, papildo jį ta informacija, kurios neatspindi schema. Grafiniai modeliai padeda ne tik vaizdžiai ir „integraliai“ išreikšti modeliavimo bei projektavimo rezultatus, bet ir leidžia greičiau pastebėti modeliavimo bei projektavimo metu daromas klaidas (praleidimus, nutrūkstancias grandines, „iš niekur atsirandančią“ informaciją, bereikalingus ryšius ir pan.). Kitaip tariant, **grafiniai modeliai gali būti ne tik projektavimo rezultatu, bet ir veiksmingu projektavimo instrumentu, panašiai kaip ir matematiniai modeliai**. Todėl projektavime taip plačiai naudojamos schemos ir brėžiniai. Tiesa, schemų bei brėžinių pagaminimas reikalauja daug darbo ir laiko, tačiau kompiuterinės grafikos atsiradimas žymiai palengvino ir pagreitino grafinių projektavimo priemonių ir grafinės projektinės produkcijos gamybą. **Visos automatizuoto projektavimo sistemos (APS) savo sudėtyje turi specialią programinę įrangą ir technines priemones, reikalingas inžinerinės grafikos uždavinių sprendimo automatizavimui**. Turi jas ir CASE.

Kaip jau buvo minėta, **objektiniam projektavimui reikalingas esybės, jų atributus ir jų ryšius galima nustatyti dviem būdais (dviejuose objektinio projektavimo loginiuose etapuose)**:

- 1) **normatyvinio objektinio projektavimo loginiame etape**, projektuojant organizacijos normatyvinę vykdomąją funkcinę struktūrą ir normatyvinius funkcijų vykdymo procesus, **nustatant tų funkcijų vykdymo subjektus ir objektus, reikalingą tam vykdymui informaciją, įvairias priemones ir ryšius**, kitaip tariant, **projektuojant organizacijos normatyvinę sudėtį, normatyvinę struktūrą ir normatyvinį funkcionavimą**; šis būdas gali būti taikomas tik kartu su funkciniu projektavimu;
- 2) **deskriptyvinio objektinio modeliavimo loginiame etape, analizuojant veikiančios organizacijos sudėtį ir struktūrą**, organizacijoje esančią SNOD, o taip pat organizacijoje vykstančius procesus bei juos aprašančią TNOD ir PNOD (jei tokia dokumentacija organizacijoje yra).

Projektuojant sudėtingas organizacines sistemas, nė vienas iš šių būdų nėra pakankamas, nes negali užtikrinti nei projekto pilnumo, nei optimalumo, o kartais – net jo realizuojamumo, todėl, norint sumažinti galimų klaidų skaičių, būtinai reikalinga abiejų šių būdų, o taip pat objektinio ir funkcinio projektavimo metodų sintezė.

Vieną **normatyvinio funkcinio – objektinio projektavimo**, kurio metu išaiškėja projektuojamai sistemai reikalingos esybės, **loginio etapo pavyzdį jau esame nagrinėję**. Tai šios metodinės priemonės septintajame skyriuje (51–54 puslapiuose) aprašytas **normatyvinės funkcinės struktūros transformavimas į normatyvinę vykdomąją struktūrą**. To transformavimo metu **formuojamos tokios normatyvinės esybės kaip funkcijų bei jų paketų vykdytojai, nustatomi ryšiai tarp jų ir sudaromos esybių ryšių diagramos (ERD)**, grafiškai vaizduojančios normatyvinę vykdomąją funkcinę struktūrą.

Projektuojant šią struktūrą, kaip jau buvo minėta, tenka atsižvelgti ne tik į funkcinį, objektinį, informacinį bei algoritminį giminingumą, bet ir į funkcijų vykdymo normatyvines trukmes. Tai reiškia, jog **projektuojant normatyvinę vykdomąją struktūrą, tenka projektuoti ir funkcijų vykdymo normatyvinius procesus**, jų TNOD bei PNOD. Tokių procesų ir tokios TNOD bei PNOD projektavimas – kitas **normatyvinio funkcinio – objektinio projektavimo**, leidžiančio nustatyti ne tik veiklos subjektus ir objektus, bet ir normatyvinius santykius tarp projektuojamos organizacijos esybių, pavyzdys. Jis bus detaliau aprašytas specialiai tam skirtame šios priemonės dešimtajame skyrelyje, todėl čia jo dabar nenagrinėsiu.

Trečiuoju **normatyvinio objektinio – funkcinio projektavimo**, skirto normatyvinių esybių ir jų ryšių diagramų formavimui, pavyzdžiu gali būti šios metodinės priemonės 54÷58 psl. aprašytas organizacijos valdymo sistemos valdymo kontūrų piramidės normatyvinis projektavimas tuo atveju, kai valdymo objektas jau funkcionuoja ir jo funkcionavimo technologijos radikaliai keisti nesiruošiama. Valdymo kontūrų piramidės projektavimas pradedamas nuo valdymo objekto (valdomų procesų) dekompozicijos į tokius posistemius, kurie galėtų būti apžvelgiami ir valdomi vieno valdymo subjekto, priklausančio žemutiniam valdymo sistemos ešelonui. Kiekvienam posistemiiui, išskirtam šios dekompozicijos metu, sudaromas valdymo kontūras ir projektuojama tame kontūre vykdomų pirminių ir antrinių (žiūr. 55÷56 šios priemonės puslapius) valdymo funkcijų TNOD bei PNOD, o taip pat kontūro subjekto SNOD. Projektuojant valdymo kontūrus, nustatomi vertikalūs ryšiai (informaciniai, pavaldumo) tarp valdymo subjektų ir objektų, o taip pat horizontalūs ryšiai tarp žemutiniojo ešelono subjektų. Po to giminingi šio ešelono valdymo kontūrai ir jų subjektai, atsižvelgiant į 30-31 ir 53-54 puslapiuose minėtus valdomumo normatyvus, jungiami į grupes, atitinkančias stambesnius organizacijos posistemius (pavyzdžiui, skyrius), ir projektuojami valdymo sistemos antrojo ešelono valdymo kontūrai. Kiekvienam iš šių stambesniųjų posistemių projektuojama SNOD ir pirminių bei antrinių valdymo funkcijų TNOD bei PNOD. Tuo pačiu metu nustatomi vertikalūs ir horizontalūs ryšiai antrajame valdymo sistemos ešelone bei vertikalūs tarpešeloniniai ryšiai, koordinuojami pirmojo ešelono subjektų tikslai ir veiksmai. Analogiškai, slenkant „nuo apačios į viršų“, projektuojama visa valdymo kontūrų piramidė. Šią piramidę vaizduojanti ERD sujungiama su valdymo objektuose vykstančių valdomųjų procesų ERD ir tokiu būdu gaunamas **organizacijos normatyvinis objektinis grafinis modelis – visos organizacijos normatyvinė ERD**. Ji gali būti saugoma kompiuterių atmintyje, o analizei, palyginimams ir korekcijoms (pavyzdžiui, funkcinio ir objektinio projektavimo ar normatyvinio projektavimo ir deskriptyvinio modeliavimo sintezės metu) tos ERD fragmentai gali būti pavaizduoti displėjų ekranuose bei atspausdinti spausdintuvais.

Manau, kad šie trys pavyzdžiai pakankamai pilnai atskleidžia **normatyvinio objektinio projektavimo esmę** – normatyvinių esybių ir jų normatyvinių ryšių projektavimą, išplaukiantį iš organizacijos tikslų formulavimo ir jos vykdomosios funkcinės struktūros bei funkcijų atlikimo TNOD projektavimo. Todėl toks projektavimas ir buvo čia pavadintas **normatyviniu funkcinio – objektinio projektavimu**.

Deskriptyvinį objektinį organizacijos modelį galima sudarinėti panašiai kaip ir normatyvinį – „nuo apačios į viršų“, tačiau dažniausiai jis sudarinėjamas „nuo viršaus į apačią“, pradedant nuo organizacijos dekompozicijos į stambias esybes (stambius posistemius) ir ERD sudarymo tų esybių lygyje. Po to posistemiai dekomponuojami į smulkesnes esybes, jų lygyje sudaromos kiekvieno posistemo ERD, ir toks dekompozicijos procesas tęsiamas iki tol, kol gaunamos elementarių esybių ERD.

Kokių nors ypatingų sunkumų, kuriant deskripcinį objektinį organizacijos modelį, nepasitaiko, nes tereikia tik užfiksuoti visas realioje veikiančioje organizacijoje egzistuojančias, esmines esybes bei esminius ryšius tarp jų ir pavaizduoti visa tai atitinkamose ERD.

Kartais, kaip ir funkcinio metodo taikymo metu, naudojama organizacijos objektinio

deskripcinio modelio sudarymo **metodų „nuo viršaus į apačią“ ir „nuo apačios į viršų“ sintezė**, padedanti sumažinti deskripcinio objektinio modelio klaidų – nepastebėtų esybių ir jų ryšių – skaičių, o taip pat nereikalingų esybių ir jų ryšių skaičių.

Normatyvinio projektavimo ir deskriptyvinio modeliavimo metodų panaudojimo atžvilgiu objektinio projektavimo loginiai etapai yra panašūs į funkcinio projektavimo loginius etapus: ir objektyviame, ir funktyviame projektavime randame normatyvinio projektavimo, deskriptyvinio modeliavimo ir jų sintezės loginius etapus.

Normatyvinio objektinio projektavimo ir deskriptyvinio objektinio modeliavimo rezultatų sintezės loginiame etape normatyvinio objektinio projektavimo metu gauti rezultatai (projektiniai sprendimai) lyginami su deskriptyvinio objektinio modeliavimo metu sudarytais deskripciniais modeliais bei kita informacija, aprašančia veikiančią organizaciją. **Palyginimo metu surasti lokalūs nesutapimai bei prieštaraavimai ir iškilę kiti neaiškumai detalieai analizuojami, nustatomos nesutapimų bei prieštaraavimų priežastys ir priimami atitinkami sprendimai** (panašiai, kaip jau buvo aprašyta septintajame šios priemonės skyrelyje, kalbant apie funkcinio normatyvinio ir deskriptyvinio projektavimo procesų sintezės loginį etapą¹). Viso to darbo rezultate **gaunamas patikslintas objektinis organizacijos sudėties ir struktūros projektas, kurio sudarymas ir yra objektinio normatyvinio projektavimo ir objektinio deskriptyvinio modeliavimo procesų sintezės loginio etapo tikslas chronologinėje sistemos sudėties ir struktūros projektavimo fazėje.**

Tačiau normatyvinio objektinio projektavimo loginis etapas, kaip jau buvo minėta, negali būti pradėtas, kol normatyvinio funkcinio projektavimo loginiame etape nebus sudaryta organizacijos normatyvinė funkcinė struktūra. Todėl normatyvinis objektinis projektavimas, nors ir galėtų vykti lygiagrečiai su deskriptyvinių objektyviu modeliavimu, gana dažnai pradedamas šiek tiek vėliau negu pastarasis. Po to abu šie loginiai etapai tęsiami lygiagrečiai išlaikydami atitinkamą lagą. Kartu vyksta ir abiejų tų procesų sintezė.

Deskriptyvinis objektyvinis modeliavimas pradedamas kartu su deskriptyviniu funktyviu modeliavimu, nes, **analizuojant organizacijos sudėtį, struktūrą ir veiklą, patogu ir tikslinga vienu metu (lygiagrečiai) aiškintis ir organizacijoje vykdomas funkcijas, ir jų subjektus, objektus, priemones, informaciją, funkcijų vykdymo metu gaunamus rezultatus, funkcijų vykdymo algoritmus, ir t.t.** Kartu su deskriptyvinio modeliavimo loginiais etapais gali būti pradėtas (ir taip pat vyksti lygiagrečiai) dar ir normatyvinio funkcinio projektavimo loginis etapas, o suprojektavus normatyvinę funkcinę struktūrą, – įsijungti ir normatyvinio objektyvinio projektavimo loginis etapas.

Galimas ir kitoks loginių projektavimo etapų išdėstymo laike variantas, kuomet normatyvinis funkcinis projektavimas pradedamas šiek tiek anksčiau už kitus penkis funkcinio ir objektyvinio projektavimo loginius etapus. Taip padaryti galima, pačioje projektavimo pradžioje, išsiaiškinus organizacijos tikslus ir juos dekomponavus iki elementarių funkcijų. Sudarius normatyvinę funkcinę struktūrą, normatyvinio funkcinio projektavimo loginį etapą galima tęsti, pradedant projektuoti normatyvinę vykdomąją funkcinę struktūrą, ir tuo pačiu metu pradėti normatyvinį objektyvinį projektavimą, deskriptyvinį funkcinį ir objektyvinį modeliavimą bei visų tų procesų sintezės loginius etapus.

Objektyvinis projektavimas skiriasi nuo funkcinio dar ir savo rezultatais. Tai antrasis pagrindinis funkcinio ir objektyvinio projektavimo metodų skirtumas.

Funkcinio projektavimo pagrindiniai rezultatai:

- 1) pradinėje chronologinėje fazėje (pradiniame chronologiniame etape) – tikslų medžiai (funkcijų hierarchijos diagramos²), **hierarchinė funkcinė organizacijos struktūra;**
- 2) vėlesnėje chronologinėje fazėje – **vykdomoji funkcinė organizacijos struktūra ir organizacijos SNOD pirmoji redakcija;**

¹ žiūr. 66-69 psl., o taip pat septynioliktąjį principą 32 psl.

² žiūr. 4-ąjį šios metodinės priemonės priedą.

- 3) dar vėlesnėje chronologinėje fazėje – **funkcijų vykdymo technologijos ir jas aprašanti TNOD (funkcijų vykdymo algoritmai) ir PNOD bei papildyta ir pakoreguota SNOD**;
- 4) kompiuterizuotos sistemos projektavimo baigiamojoje chronologinėje fazėje (funkcinio projektavimo loginio etapo pabaigoje) – **taikomoji programinė įranga ir kompiuterizuotos sistemos funkcionavimo sąlygoms pritaikyta SNOD, TNOD ir PNOD**.

Objektinio projektavimo pagrindiniai rezultatai:

- 1) pradinėje chronologinėje fazėje – **esybių aprašymai ir organizacijos elementu informaciniai išoriniai modeliai** (tame skaičiuje ir elementų matematiniai išoriniai modeliai¹, kai juos pavyksta sudaryti);
- 2) vėlesnėje chronologinėje fazėje – **esybių ryšių diagramos² (objektiniai reliaciniai modeliai) ir organizacijos bei jos posistemių sudėties ir struktūros projektai (organizacijos posistemių informaciniai vidiniai modeliai ir visos organizacijos informacinis vidinis modelis³)**.
- 3) dar vėlesnėje chronologinėje fazėje – **organizacijos, jos posistemių ir elementų funkcionavimo informaciniai modeliai (fazinių išėjimų ir įėjimų trajektorijų⁴ – aprašymai)**, o taip pat kai kurių organizacijoje vykstančių procesų **optimizaciniai modeliai⁵**;
- 4) objektinio projektavimo loginio etapo pabaigoje – **organizacijos informacinės įrangos (tame skaičiuje ir DB bei DBSV) ir ją aptarnaujančių programinės ir techninės įrangos (tame skaičiuje ir ADV tinklo) projektai**.

Cia išvardinti pagrindiniai funkcinio ir objektinio projektavimo rezultatai yra **gaunami, sąveikaujant abiemis šiems lygiagrečiai vykstantiems projektavimo procesams**, ir tik projektavimo aprašymo patogumo dėlei priskirti tam ar kitam loginiam etapui. Tikrovėje sisteminiai analitikai, atlikdami projektavimo darbus, neretai net negalėtų tiksliai atsakyti į klausimą, kurio loginio etapo darbus jie vykdo šiuo momentu, ir dėl to nesuka sau galvos, nes tikrovėje, siekiant projektavimo tikslų, įvairūs loginiai projektavimo etapai, būdami reikalingi ir vienas kitą papildydami, **sintezuojasi į vieningą organizacijos ar jos patobulinto varianto sisteminio projektavimo procesą**, apie ką bus šiek tiek detaliau parašyta devintajame šios priemonės skyrelyje, prie kurio ir pereiname.

9. FUNKCINIO IR OBJEKTINIO PROJEKTAVIMO SINTEZĖ. SISTEMINIS ORGPROJEKTAVIMAS.

Lygindami sudėtingos organizacinės sistemos funkcinio ir objektinio projektavimo procesus su šeštame šios priemonės skyrelyje aprašytu sistemotechniniu sudėtingos sistemos projektavimu, galime pastebėti eilę analogijų. Pavyzdžiui, normatyvinis funkcinis orgprojektavimas atitinka tai, kas sistemotechnikoje buvo pavadinta makroprojektavimu, o normatyvinis objektinis orgprojektavimas – tai, ką sistemotechnikoje vadina mikroprojektavimu. Septintame šios priemonės skyrelyje aprašytąją normatyvinės funkcinės struktūros projektavimą, naudodamiesi sistemotechnine terminologija, galėtume pavadinti išoriniu makroprojektavimu, o normatyvinį funkcijų vykdymo procesų projektavimą, apie ką bus rašoma dešimtajame šios priemonės skyrelyje – vidiniu makroprojektavimu. Analizuodami organizacinių sistemų vidinį makroprojektavimą, pastebėtume, kad jame galima išskirti tuos pačius vidinio makroprojektavimo loginius etapus, kurie būdingi sistemotechniniam

¹ Žiūr. šios metodinės priemonės 4-ojo skyrelio 9-12 psl.

² Žiūr. 5-ąjį šios metodinės priemonės priedą.

³ Žiūr. šios metodinės priemonės 4-ojo skyrelio 10-12 psl.

⁴ Žiūr. šios metodinės priemonės 4-ojo skyrelio 10-11 psl.

⁵ Žiūr. šios metodinės priemonės 4-ojo skyrelio 13 psl.

projektavimui: „vienetinių siūlų“, „pilno apkrovimo“ („vienetinių siūlų“ sintezės) ir „paprocesų rungtyniavimo“ („vienetinių siūlų“ derinimo).

Sudėtingų organizacijų sisteminiam projektavimui kaip ir sudėtingų techninių sistemų sistemotechniniam projektavimui būdinga makroprojektavimo ir mikroprojektavimo sintezė. Sisteminiame orgprojektavime ji vadinama funkcinio ir objektinio projektavimo sinteze. Su tokios sintezės būtinumu mes jau buvome susidūrę šios priemonės septintajame skyrelyje, aprašydami normatyvinės funkcinės struktūros transformavimą į vykdomąją struktūrą. Funkcinio ir objektinio projektavimo metodų sintezės būtinumą patvirtina ir normatyvinis funkcijų vykdymo procesų projektavimas, apie kurį bus kalbama dešimtajame šios priemonės skyrelyje. Čia apsiribosime tik pastaba, kad suprojektuoti pakankamai pilną TNOD ir kompiuterizuojamų funkcijų vykdymo algoritmus bei programas, naudojantis vien tik funkcinio projektavimo sąvokomis bei metodais ir nesinaudojant objektinio projektavimo sąvokomis bei metodais, neįmanoma. Funkcinio ir objektinio projektavimo sintezė reikalinga ir projektuojant kompiuterizuotos informacinės sistemos informacinę bei programinę įrangą.

Funkcinio ir objektinio projektavimo sintezė sisteminio projektavimo procese vyksta lygiagrečiai su funkciniu ir objektiniu projektavimu. Tam yra dvi priežastys: 1) sisteminis organizacijos ir jos veiklos projektavimas, kaip ką tik buvo minėta, negali apsiriboti vien tik funkcinio ar vien tik objektinio projektavimo elementais, nes negalima suprojektuoti funkcijų vykdymo procesų, nenagrinėjant tų funkcijų vykdytojų, tam vykdymui reikalingų priemonių, informacijos ir t.t., 2) abiejų projektavimo metodų sintezė, vykstanti lygiagrečiai su funkciniu ir objektiniu projektavimu, leidžia greičiau pastebėti funkcinio ir objektinio projektavimų klaidas, nekaupiant jų iki atskirų loginių projektavimo etapų pabaigos.

Tačiau sudėtingo sisteminio projektavimo proceso aprašymo supaprastinimo dėlei šioje metodinėje priemonėje funkcinio ir objektinio projektavimo bei jų sintezės procesus nagrinėjame atskirai – kaip atskirus sisteminio projektavimo loginius etapus: funkcinio projektavimo – septintame skyrelyje, objektinio – aštuntame, o jų sintezės loginį etapą – šiame – devintame skyrelyje.

Paprastumo dėlei tarkime, jog jau turime funkcinio organizacijos struktūros projektavimo rezultatą – organizacijos normatyvinę funkcinę struktūrą, išreikštą funkcinio normatyvinio projektavimo ir funkcinio deskriptyvinio modeliavimo loginių etapų sintezės metu patikslintomis funkcijų hierarchijos diagramomis (FHD). Tarkime taip pat, jog jau turime objektinio organizacijos struktūros ir sudėties projektavimo rezultatą – organizacijos sudėties struktūros projektą (sudėties schemą), išreikštą objektinio normatyvinio projektavimo ir objektinio deskriptyvinio modeliavimo loginių etapų sintezės metu patikslintomis esybių ryšių diagramomis (ERD). Tarkime, jog mums reikia **sintezuoti funkcinio ir objektinio projektavimo rezultatus, t.y. palyginti FHD ir ERD, sujungti šiuos informacinius organizacijos struktūros ir sudėties modelius į vieną bendrą modelį, sudarant ryšių tarp organizacijos veiklos funkcijų ir organizacijoje esančių esybių matricą – funkcijų ir esybių sąveikos matricą (FESM) bei informacinių srautų diagramas (ISD).** ir tuo pačiu metu patikslinti bei papildyti funkcinį ir objektinį organizacijos struktūros ir sudėties modelius.

Funkcinio ir objektinio organizacijos struktūros ir sudėties projektų sintezė pradedama nuo jau minėtos FESM sudarymo.

FESM sudarymo algoritmas formaliai yra analogiškas ERD sudarymo algoritmui, nes ir FESM ir ERD yra informaciniai modeliai, vaizduojantys ryšius (santykius) tarp elementų. Pagrindinį skirtumą tarp FESM ir ERD sudaro tai, kad **ERD vaizduoja ryšius tarp esybių, o FESM – ryšius tarp funkcijų ir esybių.** Tai, kad FESM dažniausiai pateikiama matricinėje formoje, o ERD – grafinėje, nėra esminis FESM ir ERD skirtumas: FESM, reikalui esant, gali būti pateikta ir grafinėje formoje, o ERD – matricinėje.

Grafinėje FESM formoje funkcijos ir esybės vaizduojamos geometrinėmis figūromis,

kurių viduje užrašomi funkcijų bei esybių pavadinimai ir jų atributai, o ryšiai (santykiai) tarp funkcijų ir esybių vaizduojami atitinkamas geometrines figūras jungiančiomis linijomis, šalia kurių užrašoma informacija, apibūdinanti tuos ryšius. Kartais ryšiai grafinėje FESM formoje numeruojami. Tuomet informacija, charakterizuojanti ryšius, pateikiama ne šalia tuos ryšius vaizduojančių linijų, o atskirame sąraše – **FESM specifikacijoje**. Kadangi kai kurie ryšiai gali kartotis, tai toks būdas sutrumpina ryšių aprašymą: vienodi ryšiai žymimi vienodais numeriais, kurių kiekvienas specifikacijoje aprašomas tik po vieną kartą. Labai dažnai besikartojančius ryšius galima žymėti ir specialiais mnemoniniais žymėjimais. Tai palengvina ir pagreitina grafinės FESM formos skaitymą ir analizę.

Matricinėje FESM formoje ryšio tarp funkcijos f_i ir esybės e_j numeris (kodas ar koks nors kitas žymėjimas) įrašomas į matricos langelį, esantį i -sios eilutės ir j -tojo stulpelio susikirtime, o patys tie numeriai (kodai ar kitokie žymėjimai) iššifruojami FESM specifikacijoje, kuri pridedama prie matricos. Jei ryšio tarp f_i ir e_j nėra, tai langelis (i,j) paliekamas tuščias arba į jį įrašomas koks nors ženklas, vaizduojantis ryšio nebuvimą (pavyzdžiui: „–“, „0“, „Ø“). FESM specifikaciją gali sudaryti ne tik ryšių tarp funkcijų ir esybių aprašymai, iššifruojantys tų ryšių žymėjimus matricos langeliuose, bet taip pat ir pačių funkcijų bei esybių aprašymai (arba nuorodos, kur tuos aprašymus galima rasti).

Kartais naudojamosi ir tokios FESM, kuriose ryšiai tarp funkcijų ir esybių nespecifikuojami. Tuomet FESM langeliuose pažymimas tik ryšių buvimas („+“, „1“) ar nebuvimas („–“, „0“, „Ø“). Tokios FESM nurodo, kokios esybės reikalingos tos ar kitos fiunkcijos vykdymui bei jos TNOD sudarymui, o patys ryšiai tarp funkcijos ir tų esybių aprašomi pačioje TNOD, kuri ir specifikuoja FESM.

Reikia pažymėti, kad FESM sudarymas ir TNOD projektavimo bei tobulinimo procesai (apie šiuos procesus kalbėsime dešimtajame šios priemonės skyrelyje) turi vykti kartu. Iš vienos pusės, neįmanoma aprašyti ryšių tarp funkcijų ir esybių, nenagrinėjant funkcijų vykdymo, t.y. neanalizuojant TNOD ir PNOD, jei tokią dokumentaciją organizacija jau turi, arba neprojektuojant jos, jei organizacijoje TNOD dar nėra. Iš kitos pusės, į FHD, ERD ir FESM sudarymą galima žiūrėti kaip į SNOD, TNOD ir PNOD projektavimo bei tobulinimo etapus, nes, pridėję prie FHD, ERD ir FESM informaciją apie elementarių funkcijų normatyvines trukmes, gautume visą informaciją, reikalingą SNOD ir TNOD projektavimui bei tobulinimui. Todėl normatyvines FHD, ERD ir FESM galima laikyti savotiška SNOD ir TNOD, šios normatyvinės dokumentacijos „pirmavaizdžiu“.

Kai organizacijoje vykdomų funkcijų ir joje egzistuojančių esybių sąrašai nėra dideli, tai funkcinio ir objektinio projektų sintezės rezultata galima atvaizduoti vienoje FESM, turinčioje tiek eilučių, kiek funkcijų yra organizacijoje vykdomos veiklos funkcijų sąraše, ir tiek stulpelių, kiek esybių yra organizacijoje egzistuojančių esybių sąraše. Žinoma, vieną bendrą visai organizacijai FESM galima sudaryti ir tuo atveju, kai minėti sąrašai yra dideli, tačiau tokia FESM būtų sunkiai apžvelgiama ir nepatogi analizei, kai tą analizę vykdo žmonės. Tokiais atvejais vietoj vienos FESM sudaroma **hierarchinė funkcijų ir esybių sąveikos matricų sistema (HFESMS)**, atitinkanti organizacijos dekompoziciją į posistemius. HFESMS sudarymą aprašysime šiek tiek vėliau, o dabar panagrinėsime atvejus, kurie gali atsitikti, sintezuojant FHD ir ERD į vieną bendrą FESM, kai funkcinis ir objektinis projektai dar nėra pakankamai tobuli. Tokie atvejai gali atsitikti, pavyzdžiui, kai sudėtingos organizacijos didelė FESM projektuojama, naudojantis kompiuteriu ir APS programine įranga.

Pavyzdžiui, **gali paaiškėti, kad automatizuotu būdu gautoje bendroje FESM viena arba kelios eilutės yra tuščios. Galimos tokios šio reiškinių, rodančio, kad projektuojant kažkur apsirakta, priežastys:** 1) projektuotojai, sudarydami FESM, nepakankamai pilnai išnagrinėjo funkcijų ryšius su esybėmis arba pamiršo įvesti į kompiuterį kai kuriuos ryšius; 2) projektuotojai, sudarydami ERD, šį darbą darė nepakankamai atsakingai ir nepastebėjo bei neįtraukė į objektinį projektą kai kurių esminių esybių; 3) projektuotojai, sudarydami FHD, „persistengė“ ir, giliau nepaanalizavę, įtraukė į funkcinį projektą kai kurias nereikalingas

funkcijas; 4) projektuotojai, būdami nepakankamai kompetetingi ir atsakingi, padarė daug klaidų, priklausančių keliems (galbūt, ir visiems) iš trijų čia išvardintų tipų.

Susidūrus su reiškiniu, kad kai kurios FESM eilutės yra tuščios, visų pirma, reikia išsiaiškinti, dėl kurios iš aukščiau išvardintų priežasčių taip atsitiko.

Jei taip **atsitiko dėl pirmosios priežasties**, tai reikia ne tik įvesti į kompiuterį papildomą informaciją, reikalingą kai kurių langelių tuščiose eilutėse užpildymui, bet ir patikrinti, ar nėra panašių praleidimų (egzistuojančių ar turinčių egzistuoti ryšių neužfiksavimo ir informacijos apie juos neįvedimo) ir visose kitose matricos eilutėse. Tokių praleidimų buvimas byloja apie nepakankamą sudarytos TNOD kokybę ir jos tobulinimo būtinumą.

Jei **tuščios eilutės atsirado dėl antrosios priežasties**, tai reikia ne tik papildyti FESM naujais stulpeliais, atitinkančiais nepastebėtas bei praleistas esybes, surastas nagrinėjant tų funkcijų, kurių eilutės buvo tuščios, vykdymą, bet ir patikrinti, nagrinėjant kitų funkcijų vykdymą, ar nepastebėtų esybių nebuvo daugiau. Praleistų esybių buvimas byloja apie nepakankamą objektinio projekto kokybę ir jo tobulinimo būtinumą.

Jei **tuščios eilutės atsirado dėl trečiosios priežasties**, tai reikia ne tik išbraukti iš FESM nereikalingas eilutes bei iš FHD nereikalingas funkcijas, bet ir paanalizuoti funkcinių modelių, ar į jį nebuvo įtraukta dar ir kitų bereikalingų funkcijų. Bereikalingų funkcijų buvimas funkciniam projekte byloja apie nepakankamai kokybišką funkcinio projekto sudarytojų darbą ir to darbo tobulinimo būtinumą.

Jei **tuščios eilutės atsirado dėl ketvirtosios priežasties**, tai reikia imtis ne tik atitinkamų aukščiau išvardintų priemonių, bet ir susirūpinti bendra visų projektavimo darbų kokybe ir projektuotojų žema kvalifikacija bei silpnu jų atsakomybės jausmu, silpna projektavimo darbų kokybės kontrole.

Gali atsitikti ir taip, kad automatizuotu būdu gautoje didelės sudėtingos organizacijos **bendroje FESM vienas ar keli stulpeliai bus tušti**. Tai reikštų, kad: 1) arba objektinio modelio kūrėjai „persistengė“ ir įvedė į tą modelį kai kurias neesmines esybes, 2) arba funkcinio projekto sudarytojai nepastebėjo kai kurių funkcijų, kurių vykdymui reikalingos esybes, dabar turinčios funkcijų ir esybių sąveikos matricoje tuščius stulpelius, 3) arba atsitiko ir pirma, ir antra. Todėl, atsiradus FESM tuščių stulpelių, visų pirma būtina tiksliai nustatyti tokio reiškinio priežastis ir kaltininkus.

Jei **tušti stulpeliai atsirado dėl pirmosios priežasties**, tai reikia ne tik pakoreguoti objektinį projektą, eliminuojant neesmines esybes, bet ir išsiaiškinti su to projekto kūrėjais, kodėl tos neesminės esybės pateko į projektą. Juk gali būti ir taip, kad objektinio projekto kūrėjai, įtraukdami tas esybes į projekto sudėtį, vadovavosi svariais argumentais, į kuriuos funkcinio projekto projektuotojai neatkreipė dėmesio. Galbūt informacijos apie tas esybes buvimas leistų patobulinti funkcijų vykdymą ir padidinti organizacijos efektyvumą.

Jei **tušti stulpeliai atsirado dėl antrosios priežasties**, tai reikia papildyti funkcinį projektą, tačiau tik gerai įsitikinus, kad tos funkcijos, kurių vykdyme naudojamos esybės iš kol kas tuščių FESM stulpelių, yra tikrai reikalingos. Įsitikinus, kad buvo praleistos tikrai reikalingos funkcijos, būtina rimtai susirūpinti funkcinio projektavimo kokybe ir kruopščiai patikrinti funkcinį projektą, ar nebuvo praleista ir daugiau reikalingų funkcijų.

Trečiuoju iš aukščiau išvardintų atvejų reikia ne tik ištaisyti pastebėtas klaidas, bet ir imtis priemonių projektavimo kokybei gerinti ir projektuotojų atsakingumui didinti. Panašių priemonių tuo labiau reikia imtis, jei gautoje FESM yra ir tuščių eilučių, ir tuščių stulpelių.

Kai bendroji FESM tampa pernelyg didelė ir dėl to sunkiai apžvelgiama bei nepatogi analizei, tai, kaip jau buvo minėta, vietoj jos sudaroma HFESMS, kurios žemutiniojo sluoksnio funkcijų ir esybių sąveikos matricos vaizduoja organizacijos žemutiniojo sluoksnio posistemų struktūrą ir sudėtį, o aukštesniųjų sluoksnių FESM – žemesniųjų sluoksnių posistemų ryšius aukštesniųjų sluoksnių posistemiuose.

Sudarant HFESMS, organizacijos dekompozicija į posistemius gali būti vykdoma „nuo viršaus į apačią“ ir atitikti organizacijos tikslų dekompoziciją į tikslų medžius, kuria

naudojamasi normatyvinės funkcinės struktūros projektavimo metu (žiūr. septintąjį šis priemonės skyrelį). Tuomet hierarchinėje FESM sistemoje gaunama tokia ryšių tarp FESM struktūra, kuri analogiška FHD struktūrai.

Tačiau posistemų išskyrimas gali būti vykdomas ir „nuo apačios į viršų“, panašiai kaip projektuojama normatyvinė vykdomoji struktūra, gaunant ją iš normatyvinės funkcinės struktūros (žiūr. 51÷58 puslapius). Tuomet ryšių tarp FESM struktūra bus analogiška organizacijos vykdomajai struktūrai.

Pakankamai tobulai suprojektuotoje organizacijoje **struktūros, gautos tikslų dekompozicijos metu, ir struktūros, gautos metodu „nuo apačios į viršų“, privalo jei ne pilnai sutapti, tai bent jau apytikriai atitikti viena kitą. Todėl metodų „nuo viršaus“ ir „nuo apačios“ sintezė gana dažnai naudojama, tobulinant sudėtingos organizacijos struktūrą, o taip pat projektuojant sudėtingas sistemas bei jas gaminant (realizuojant).** Pavyzdžiui, techninės sistemos projektuojamos „nuo viršaus į apačią“, o gaminamos „nuo apačios į viršų“.

Dekomponavus sudėtingą organizaciją į posistemius ir sudarius tų posistemų hierarchinę sistemą, sudaromos žemutiniojo posistemų sistemos sluoksnio visų posistemų FESM. Sudarant tas FESM, gali atsitikti taip, kad kai kurių FESM kai kurių eilučių ar kai kurių stulpelių visi langeliai liks tušti. Tokio reiškimo priežastys gali būti tokios pačios, kaip jau išnagrinėtos tuščių eilučių bei tuščių stulpelių atsiradimo bendrojoje visos organizacijos FESM priežastys. Jos šalinamos anksčiau aprašytais būdais. Tačiau gali būti ir naujų priežasčių, atsiradusių dėl funkcijų ir esybių aibės dekompozicijos į poaibius neatitikimo. Visais tokiais atvejais reikia detalai išsiaiškinti, kokios esybės reikalingos, sudarant tų funkcijų, kurių eilutės liko tuščios, TNOD, o taip pat, kokių funkcijų vykdymui reikalingos tos esybės, kurių stulpeliai liko tušti. Turint tokią informaciją ir lyginant ją su į funkcinį projektą ir į organizacijos posistemų funkcinis projektus įtrauktais funkcijų sąrašais bei į objektinį projektą ir į organizacijos posistemų objektinis projektus įtrauktais esybių sąrašais, galima nustatyti tikrąsias tuščių eilučių ir tuščių stulpelių priežastis, pašalinti jas ir ištaisyti padarytas klaidas. Gali tekti pakoreguoti posistemų FESM, papildant ar eliminuojant jų eilutes bei stulpelius, perkelti kai kurias funkcijas bei esybes iš vieno posistemio į kitą posistemio FESM.

Sudarant posistemų FESM, galima susidurti ir su funkcijų bei esybių dubliavimu: tos pačios funkcijos, o taip pat ir tos pačios esybės gali būti sutinkamos keliuose žemutiniojo sluoksnio FESM. Tokį dubliavimą reikia stengtis mažinti, pertvarkant FESM sudėtį t.y. perkelti kai kurias funkcijas bei esybes iš vieno FESM į kitas. Tačiau visiškai likviduoti dubliavimą ne visada pavyksta, nes tenka susidurti ir su tokiais atvejais, kai ta pati esybė ir ta pati funkcija yra būtina kelių posistemų sudėtinė dalis. Kartais, projektuojant kontrolės posistemius bei siekiant didesnio organizacijos veiklos patikimumo, funkcijų bei esybių dubliavimas skirtinguose posistemiuose įvedamas specialiai.

Kalbant apie FESM sudarymą, norėtusi dar kartą pabrėžti, kad FESM sudarymas ir to sudarymo metu išskylančių neaiškumų bei klaidų išsiaiškinimas neįmanomas be atitinkamos TNOD sudarymo bei analizės. Loginis FESM sudarymo etapas turi vykti lygiagrečiai su TNOD projektavimo bei tobulinimo etapu: TNOD sudarymas duoda informaciją, reikalinga FESM sudarymui ir korekcijai, o FESM sudarymas padeda suderinti lokalius TNOD projektus, sujungiant juos į visos organizacijos veiklos sisteminį projektą. Lygiagretus TNOD ir FESM projektavimas – tai dar vienas mikroprojektavimo ir makroprojektavimo metodų sintezės pavyzdys, rodantis tokios sintezės reikalingumą ir naudingumą.

Sudarius žemutiniojo sluoksnio FESM, reikia nustatyti ryšius tarp jų. Tai galima padaryti dviem būdais: „išoriniu“ ir „vidiniu“.

„Išoriniu“ ryšių tarp FESM nustatymo būdu vadinsime tų ryšių nustatymą

organizacijos dekompozicijos į posistemius metu, kuomet, **dekomponuojant organizaciją į posistemius, nagrinėjami organizacijoje esantys bei jai reikalingi vidiniai ryšiai ir stengiamasi išskirti posistemius taip, kad kuo didesnė tų ryšių dalis liktų vidiniais posistemių ryšiais ir kuo mažesnė jų dalis taptų tarpposisteminiais ryšiais.** Tie tarpposisteminiai ryšiai ir turi tapti ryšiais tarp FESM, vaizduojančių posistemių struktūrą ir sudėtį.

„Vidiniu“ ryšių tarp FESM nustatymo būdu vadinsime tokį būdą, kuris leidžia surasti minėtus ryšius, nagrinėjant pačias FESM bei į jų sudėtį įeinančių funkcijų TNOD.

Abu metodai vienas kitą papildo ir palengvina bei pagreitina ryšių tarp žemutiniojo sluoksnio FESM nustatymą, todėl naudojamosi tų metodų sinteze.

Kadangi „vidinis“ metodas sudaro galimybes patobulinti jau esamą sistemos bei jos posistemių struktūrą ir gali būti panaudotas ne tik deskriptyvinio modeliavimo, bet ir normatyvinio projektavimo atvejais, kai projektuojamoji organizacija dar neegzistuoja ir jos posistemiai dar tik formuojami, tai šis metodas yra galingesnis ir universalesnis už pirmąjį, kuris gali būti taikomas tik jau veikiančiai ar jau suprojektuotai organizacijai. Be to, „vidinis“ metodas gali būti formalizuotas ir net kompiuterizuotas.

Aprašysime šį metodą šiek tiek detaliau, atsižvelgdami į jo aktualumą SIS projektuotojams, kadangi, projektuodami SIS, mes stengsimės kuo plačiau naudotis projektavimo automatizavimo priemonėmis, leidžiančiomis pagreitinti projektavimą ir pagerinti projektų kokybę, didinant lokalių projektų suderinamumą, o taip pat stengsimės tobulinti organizacijos struktūrą.

Tarkime, jog jau turime visas žemutiniojo sluoksnio FESM ir nagrinėjame vieną iš jų, norėdami nustatyti jos ryšius su kitomis to sluoksnio FESM.

Išskiriame pirmąją (o vėliau, antrąją, trečiąją, ..., i-tąją ... ir paskutiniąją), įeinančią į nagrinėjamos FESM sudėtį funkciją f_1 , ir nagrinėjame tos funkcijos vykdymo TNOD (MTK, OK ar OI), nustatydami, su kokiomis esminėmis esybėmis tenka operuoti, vykdant funkciją f_1 . Jei visos esminės esybės, minimos funkcijos f_1 technologinėje normatyvinėje organizacinėje dokumentacijoje bei reikalingos tokios TNOD sudarymui, įeina į nagrinėjamos FESM sudėtį kaip jos stulpeliai, tai **funkcija f_1 vadinama vidine** nagrinėjamos FESM ir ją atitinkančio organizacijos posistemo atžvilgiu. Tokia funkcija f_1 tiesioginių ryšių su kitomis FESM neturi, nors kituose posistemiuose ji gali ir kartotis kaip kitų posistemių sudėtinė dalis bei gali kartotis ir tos funkcijos f_1 TNOD sudarymui reikalingos esybės, dubliuodamos nagrinėjamai FESM priklausančias esybes.

Jei pasirodo, kad kai kurios funkcijos f_i (ja gali būti ir funkcija f_1) TNOD sudarymui nepakanka esybių, įeinančių į nagrinėjamos FESM sudėtį, ir reikalingos dar kai kurios kitos esminės esybės, tai reikia išsiaiškinti, ar tos papildomai reikalingos esybės įeina į kitų FESM bei į organizacijos objektinio modelio ERD sudėtį.

Jei tokių esybių kitose FESM (ir objekiniame visos organizacijos modelyje) nerandame, o nagrinėjamoji funkcija f_i yra tikrai reikalinga, tai darome išvada, kad objektinis projektavimas buvo vykdomas nepakankamai atidžiai. Tuomet išsiaiškinamos priežastys, dėl kurių reikalingos esybės liko nepastebėtos ir neįtrauktos į objektinį projektą, ir sprendžiama, ar dėl panašių priežasčių negalėjo būti daugiau tokių praleidimų. Pastebėti objektinio projektavimo trūkumai pašalinami ir papildomos objektinio projekto ERD. Papildoma ir nagrinėjamoji FESM, įvedant į jos sudėtį naujus stulpelius toms papildomoms esybėms, kurios reikalingos nagrinėjamos funkcijos f_i sudarymui. Pačiai tai funkcijai paliekamas nagrinėjamo posistemo vidinės funkcijos statusas.

Jei nagrinėjamai funkcijai f_i^* reikalingos papildomos esybės įeina į kitų FESM sudėtį, tai reikia pažiūrėti, ar negalima taip pertvarkyti žemutiniojo sluoksnio funkcijų ir esybių sąveikos matricių, perkeltiant funkcijas ar/ir esybes iš vienos FESM į kitas, kad, nedubliuojant esybių, kuo daugiau funkcijų taptų vidinėmis naujų FESM atžvilgiu. Jei taip padaryti pavyksta, tai **gaunama tobulesnė organizacijos dekompozicija į posistemius, padidinamas posistemių ir visos organizacijos valdomumas.**

Tačiau padaryti visas funkcijas f_i^* vidinėmis, nedubliuojant tų pačių esybių keliose FESM, praktiškai neįmanoma, nes realiose organizacijose neegzistuoja visiškai izoliuotų posistemų, nesurištų su kitais posistemiais nei funkcijomis, nei esybėmis. Todėl nagrinėjamoje FESM gali atsirasti viena ar kelios funkcijos, kurios surištos ne tik su esybėmis, priklausančiomis nagrinėjamai FESM, bet ir su esybėmis, priklausančiomis kitoms FESM.

Tokias funkcijas f_i^* vadinsime nagrinėjamos **FESM įėjimais**, o kokios nors tos FESM esybės, reikalingas ir kitų FESM įėjimams, vadinsime tos **FESM išėjimais**. Esybes, priklausančias nagrinėjamai FESM, tačiau nesančias jos išėjimais, t.y. nesančias papildomomis esybėmis, reikalingomis kitų FESM įėjimams, vadinsime nagrinėjamos FESM ir atitinkamo posistemo **vidinėmis esybėmis**.

Ką vadinti FESM įėjimais, o ką – išėjimais, yra susitarimo dalykas, kaip susitarimo dalykas yra ir tai, ką vaizduoti FESM eilutėmis o ką – stulpeliais. Dažniausiai FESM eilutės vaizduoja funkcijas, o stulpeliai – esybes, tačiau galima daryti ir atvirkščiai: eilutes skirti esybėms, o stulpelius – funkcijoms. Todėl ir FESM esybės, kurios nėra vidinės tos FESM atžvilgiu, galima vadinti FESM įėjimais, o tos FESM išorines funkcijas – jos išėjimais. Tačiau vieną kartą priėmus kokią nors susitarimą, toliau reikia jo laikytis. **Mes laikysimės susitarimo, kad funkcijos, nesančios tik vidinėmis kokios nors FESM atžvilgiu, bus vadinamos tos FESM įėjimais, o esybės, nesančios tik vidinėmis kokios nors FESM atžvilgiu, bus vadinamos tos FESM išėjimais.** Toks pavadinimų pasirinkimas turi ir šiokią – tokią „fizinę“ bei „informacinę“ prasmę: natūralu tas esybės, kurios egzistuoja kokiam nors posistemyje, tačiau aptarnauja ne tik savo posistemo, bet ir kitų posistemų funkcijas, laikyti to posistemo išėjimais, o tas nagrinėjamojo posistemo funkcijas, kurių vykdymui reikia informacijos iš kitų posistemų – nagrinėjamojo posistemo įėjimais.

Suradę visų žemutiniojo sluoksnio FESM įėjimus ir išėjimus, mes tokiu būdu nustatome ryšius tarp to sluoksnio FESM ir galime pradėti tų FESM grupavimą ir jungimą į aukštesnio sluoksnio posistemius ir jų FESM.

Aukštesniojo sluoksnio posistemiai, naudojantis jau minėtuoju antruoju („vidiniu“) metodu, gaunami, einant „nuo apačios į viršų“ ir jungiant žemesniojo sluoksnio posistemius ir jų FESM taip, kad kuo didesnis ryšių tarp tų posistemų skaičius taptų vidiniais aukštesniojo sluoksnio posistemų ryšiais, o patys to aukštesniojo sluoksnio posistemiai netaptų neapžvelgiami žemesniojo sluoksnio posistemų stratoje.

Tokiu būdu gauti aukštesniojo sluoksnio posistemiai ir jų FESM, vaizduojančios ryšius tarp žemesniojo sluoksnio FESM įėjimų (aukštesniojo sluoksnio FESM eilučių) ir išėjimų (aukštesniojo sluoksnio FESM stulpelių), ne visuomet pilnai sutampa su tais aukštesniojo sluoksnio posistemiais, kurie gaunami, einant „nuo viršaus į apačią“, t.y. dekomponuojant organizacijos tikslus. **Abiejų struktūrų, gautų naudojantis metodais „nuo viršaus“ ir „nuo apačios“, palyginimas leidžia pastebėti kai kurias klaidas ir gauti tobulesnę organizacijos dekompoziciją į aukštesniojo lygio posistemius.**

Kai žemutiniojo sluoksnio posistemų ir jų FESM nėra daug, tai ryšių tarp tų FESM nustatymas ir tų FESM jungimas į aukštesniojo lygio posistemius bei į jų FESM nėra per daug sudėtingas. Tačiau, kai žemutiniojo sluoksnio posistemų ir jų FESM yra tiek daug, kad net aukštesniojo sluoksnio posistemų ir jų FESM pasidaro daug (o taip atsitikti sudėtingose didelėse organizacijose gali todėl, kad netikslinga daryti aukštesniojo sluoksnio posistemų neapžvelgiamais žemesniojo sluoksnio posistemų stratoje), žemutiniojo sluoksnio posistemų ir jų FESM nagrinėjimas, ryšių tarp jų nustatymas, tų posistemų ir jų FESM grupavimas į giminingų FESM paketus bei jungimas į aukštesniojo sluoksnio posistemius ir į jų FESM tampa gana sudėtingu procesu. Tokiais atvejais gali padėti kompiuterinės technikos ir atitinkamos programinės įrangos panaudojimas, o taip pat orientacija į aukštesniojo sluoksnio posistemius, kurie buvo gauti organizacijos dekompozicijos metu, einant „nuo viršaus į apačią“.

Sudarant aukštesniojo sluoksnio FESM, aprašančią atitinkamą aukštesniojo sluoksnio

posistemį, kaip jau buvo minėta, matricos eilutėmis tampa tos funkcijos, kurios buvo į tą posistemį patekusių žemesniojo sluoksnio posistemų ir jų FESM įėjimais, o stulpeliais – tos esybės, kurios buvo tų posistemų ir jų FESM išėjimais. Aukštesniojo sluoksnio FESM langeliuose surašomi atitinkamų ryšių žymėjimai ar numeriai, o FESM specifikacijose tie žymėjimai bei numeriai iššifruojami, t.y. duodami atitinkamų ryšių aprašymai. Tose specifikacijose taip pat duodamos nuorodos į aukštesniojo sluoksnio FESM eilutes atitinkančių funkcijų ir tų FESM stulpelius atitinkančių esybių aprašymus.

Analogiškai (kaip ir analizuojant žemutiniojo sluoksnio FESM) nustatomos aukštesniojo sluoksnio funkcijų ir esybių sąveikos matricų vidinės funkcijos bei vidinės esybės ir išskiriami kiekvienos tokios FESM įėjimai ir išėjimai.

Kartais, kai žemutiniojo sluoksnio FESM pasidaro labai daug, prieš pradedant tas FESM jungti į aukštesniojo sluoksnio FESM, elementarios funkcijos ir elementarios esybės **agreguojamos** į sudėtingesnes funkcijas ir sudėtingesnes esybes. Tai leidžia sumažinti analizuojamų funkcijų bei esybių skaičių ir palengvinti žemutiniojo sluoksnio FESM jungimą į aukštesniojo sluoksnio FESM bei supaprastinti tų FESM analizę. Kitaip tariant, **pereinant nuo žemesniojo sluoksnio FESM prie aukštesniojo sluoksnio matricų, gali būti pereinama nuo detalesnio aprašymo stratos prie stratos, kurioje naudojamos labiau agreguotos sąvokos, ir todėl FESM skirtinguose sluoksniuose gali būti aprašomas ir analizuojamas skirtingose stratosė, besiskiriančiose aprašymo detalumu.**

Perėjimais iš vienos stratos į kitą ir funkcijų bei esybių aprašymu ir analize skirtingose stratosė gali būti naudojamos ne tik einant „nuo apačios į viršų“, bet ir dekomponuojant organizaciją į posistemius „nuo viršaus į apačią“, kuomet, pereinant nuo aukštesniojo sluoksnio prie žemesniojo, gali kartu su organizacijos dekompozicija vykti ir esybių bei funkcijų **dezagregavimas** į smulkesnes esybes ir į smulkesnes funkcijas. Tuo gali būti naudojamos, projektuojant HFESMS „nuo viršaus į apačią“.

Kai HFESMS projektuojama „nuo apačios į viršų“ (o toks metodas, kaip jau buvo minėta, leidžia ne tik gauti organizacijos FESM, bet ir patobulinti tos organizacijos ir jos posistemų struktūrą bei sudėtį), tai nagrinėjamojo sluoksnio FESM jungiamos į aukštesniojo sluoksnio FESM (o šios – į dar aukštesniojo) ir tas jungimas vykdomas lygiai taip pat, kaip buvo atliekamas aukščiau aprašytas žemutiniojo sluoksnio FESM jungimas į aukštesniojo sluoksnio FESM.

Tęsiant šią jungimo procedūrą, galų gale gaunama paties aukščiausiojo sluoksnio FESM, vaizduojanti visos organizacijos struktūrą ir sudėtį stambiausių organizacijos posistemų stratoje. Jei, jungiant FESM į aukštesniojo sluoksnio FESM, vyko ir esybių bei funkcijų agregavimas, tai **aukščiausiojo sluoksnio FESM gaunama ne tik stambiausių posistemų bei jų FESM, bet ir labiausiai agreguotų esybių bei funkcijų stratoje.**

Visų sluoksnių FESM kartu paimtos ir sudaro organizacijos HFESMS, vaizduojančią visos organizacijos visų funkcijų ir visų esybių ryšius, hierarchinę tų ryšių sistemą. Turėdami HFESMS, tuos ryšius galime analizuoti ir „nuo viršaus į apačią“ ir „nuo apačios į viršų“.

Prijungę prie HFESMS dar ir ERD bei FHD ir organizacijos vykdomosios struktūros projektą (VSP), gauname gana pilną formalizuotą organizacijos struktūros ir sudėties aprašymą – **struktūrizuotą ir formalizuotą organizacijos struktūros ir sudėties normatyvinį modelį.**

Kaip jau buvo minėta, vietoj čia aprašytos sudėtingos hierarchinės kelių sluoksnių sistemos, sudarytos iš FESM, vaizduojančių sistemos ir jos posistemų struktūrą ir sudėtį, naudojantis kompiuterine technika ir jos programine įranga galima iškart sudaryti visos organizacijos bendrąją FESM, nedeckomponuojant organizacijos į posistemius. Tokios FESM sudarymo algoritmas yra netgi žymiai paprastesnis, negu HFESMS sudarymo algoritmas, tačiau žmonėms naudotis tokia didžiule bendrąja FESM ir ją analizuoti būtų labai nepatogu ir sunku, nes tokią didelės sudėtingos organizacijos FESM, sudarytą elementarių funkcijų ir elementarių esybių stratoje, būtų neįmanoma apžvelgti. Todėl ir sudarinėjama būtent HFESMS, todėl ir kuriama gana sudėtinga programinė įranga kompiuteriams, leidžianti

HFESMS kūrimo procesą didžiąja dalimi kompiuterizuoti.

FESM (ir HFESMS) sudarymu pasiekiamas vienas svarbiausių funkcinio ir objektinio projektavimo sintezės loginio etapo tikslų – suprojektuojama organizacijos normatyvinė struktūra ir sudėtis.

Tačiau organizacijos struktūros ir sudėties projekto sudarymas nėra vienintelis funkcinio ir objektinio projektavimo bei jų sintezės loginių etapų, vykstančių lygiagrečiai, tikslas. Kitas labai svarbus tikslas – organizacijoje turinčių vykti procesų projektavimas, jų TNOD bei PNOD sudarymas.

Kaip skaitytojas jau galėjo pastebėti, skaitydamas FESM sudarymo aprašymą, abu šie tikslai negali būti pasiekti izoliuotai vienas nuo kito: sudarant FESM, reikalinga TNOD, o sudarant TNOD ir PNOD, reikia žinoti funkcinę ir objektinę organizacijos struktūrą. Todėl ir sistemos sudėties bei struktūros ir TNOD ir PNOD projektavimas vyksta lygiagrečiai, papildydami vienas kitą, padėdami vienas kitam. Todėl ir reikalinga jų sintezė, kuri taip pat vyksta lygiagrečiai.

Tačiau aprašinėti tiek daug vienu metu lygiagrečiai vykstančių procesų (funkcinį ir objektinį projektavimą bei jų sintezę, normatyvinį ir deskriptyvinį modeliavimą bei jų sintezę, organizacijos struktūros ir procesų projektavimą bei jų sintezę, o taip pat ir visų čia išvardintų procesų sintezę) yra labai sunku, nes puslapiuose sakiniai rašomi ne lygiagrečiai, o nuosekliai. Šis sunkumas panašus į tą sunkumą, su kuriuo susiduria programuotojai, pereidami nuo nuoseklaus programavimo kompiuteriui, turinčiam vieną procesorių, prie lygiagretaus programavimo daugiaprocesorinėms kompiuterinėms sistemoms. Netgi jei būtų sudarytas „lygiagretus“ lygiagrečiai vykstančių procesų tekstinis aprašymas, pavyzdžiui, jei sakiniai, aprašantys lygiagrečiai vykstančius veiksmus, būtų rašomi ant labai plataus lapo lygiagrečiai, t.y. atskiruose to lapo stulpeliuose, nurodant ryšius tarp atskirų sakinių, panašiai, kaip tai daroma schemose, vaizduojančiose lygiagretius procesus, tokį tekstą skaityti taip pat būtų nelengva, nes turėtume, slinkdami akimis lapu žemyn, nuolat šokinėti nuo vieno stulpelio prie kito, nutraukdami stulpelių sakiniuose dėstomų minčių tėkmę.

Todėl šioje metodinėje priemonėje ir buvo pasirinktas toks lygiagrečiai vykstančių procesų aprašymo būdas, kuriuo naudojantis, lygiagrečiai vykstantys procesai, sudarantys logiškai surištų veiksmų nuoseklias grandines, aprašomi atskirai kaip tam tikri „loginiai etapai“ (šis pavadinimas buvo pasiskolintas iš sistemotechninės terminologijos, nors žodis „etapas“ čia, galbūt, nelabai vykusiai parinktas ir, matyt, geriau tiktų žodis „**grandinė**“). Aprašant tokias grandines, t.y. jų grandis ir ryšius tarp grandinės grandžių, stengiamasi kartkartėmis priminti skaitytojui ir apie ryšius, egzistuojančius tarp grandinių, o kartais ir išskirti atskirus skyrelius tų grandinių ryšių ir grandinių sintezės aprašymui. Taip, pavyzdžiui, ir daroma šioje metodinėje priemonėje: septintasis skyrelis buvo skirtas funkcinio projektavimo aprašymui, aštuntasis – objektinio, šis – devintasis – abiejų minėtųjų projektavimo procesų sintezės aprašymui, o dešimtajame skyrelyje bus aprašytas veiklos procesų ir jų TNOD bei PNOD projektavimas, t.y. kiekvienas skyrelis aprašo **vieną iš lygiagrečiai vykstančių projektavimo procesų** (su atitinkamomis nuorodomis į kitus procesus, sąveikaujančius su aprašomuoju).

Žinoma, toks aprašymo būdas turi savų trūkumų. Pirmasis iš jų – bendras bet kurios dekompozicijos trūkumas: skaidydami sudėtingą procesą į lygiagrečiai vykstančius paprocesius ir aprašydami juos atskirai, mes būname priversti nutraukti ryšius tarp tų paprocesių ir priklausomus paprocesius vaizduoti lyg ir nepriklausomais. Antrasis trūkumas išplaukia iš pastangų sušvelninti pirmojo trūkumo pasekmes: norėdami nepalikti „už borto“ egzistuojančias tarp paprocesių priklausomybes, mes, aprašinėdami vieną iš paprocesių, būname priversti kartkartėmis aprašinėti kitų paprocesių fragmentus, o aprašinėdami paprocesių sintezę – vėl aprašinėti sintezuojamus paprocesius ir jų ryšius. Dėl šių priežasčių tekste dažnai atsiranda beveik dubliuojantys vienas kitą sakiniai ir ištisos pastraipos, o tai nėra malonu tiems

skaitytojams, kurie, turėdami gerą atmintį ir vaizduotę, ir be tų pasikartojančių sakinių sugeba integraliai mąstyti ir „matyti“ aprašomuosius papročius ir visą procesą. Be to, žymiai išsiplečia aprašymo apimtis, o tai irgi vargina skaitytojus.

Tačiau su šiais trūkumais tenka taikstyti, nes geresnio aprašymo žodžiais būdo kol kas nesugalvota. Ateityje, tobulėjant kompiuterinei technikai ir jos programinei įrangai, surištų į sudėtingą procesą lygiagrečiai vykstančių procesų aprašymui bei vaizdavimui, matyt, bus naudojamos displėjų ekranuose matomos „dinaminės schemos“ (schemos, imituojančios procesus), plačiai naudojančios piktografiją ir lydimos paaiškinančių tekstų, kurių detalumo lygį ir formą (garsinę, rašytinę), o taip pat tokių „multfilmų“ kadru kaitos greitį galės pasirinkti ir keisti pats skaitytojas savo nuožiūra. Dabar gi, kol tokių „multfilmų“ dar nėra, tenka naudotis tokiais gana netobulais ir nuobodžiais projektavimo loginių etapų (loginių grandinių) aprašymais, kaip šis.

Tuo ir baigsime šį skyrelį ir pereisime dar prie vienos tokios lygiagrečiai su jau aprašytais orgprojektavimo procesais vykstančios loginės grandinės – organizacijoje turinčių vyksti veiklos procesų ir jų TNOD projektavimo – aprašymo.

10. PROCESŲ IR JŲ TNOD BEI SRAUTŲ IR JŲ DIAGRAMŲ PROJEKTAVIMAS. OPERACIJŲ TYRIMAS IR JO TAIKYMAS ORGPROJEKTAVIME

Organizacijoje vykstančiuose **veiklos procesuose** jos aktyviuosius ir pasyviuosius elementus jungia įvairūs **sąveikavimo srautai**. Tie srautai gali turėti įvairią prigimtį. Vieni iš jų skirti tam, kad palaikytų elementų judėjimą, keistų pasyviųjų elementų padėtį, judėjimo greitį bei jo kryptį, sudarytų reikalingas sąlygas visų organizacijos elementų veiklai. Tai – **energetiniai srautai**. Kiti srautai gali turėti medžiaginę prigimtį – teikti organizacijos elementų ir jos posistemių veiklai reikalingas medžiagas (pavyzdžiui, medžiagas, reikalingas organizacijos elementų atnaujinimui, naujų elementų, o taip pat ir energijos gamybai ir pan.). Tokius srautus vadinsime **medžiaginiiais**.

Energetinius ir medžiaginius objektų bei subjektų poveikius vienas į kitą vadinsime **pirminiais poveikiais**. Pirminiai poveikiai būdingi tiek pasyviems, tiek ir aktyviems organizacijų elementams. Tačiau tokiems aktyviems organizacijos elementams kaip žmonės būdinga ne tik pirmoji, bet ir **antroji signalinė sistema**. Toje sistemoje įvairūs pirminiai poveikiai gali reikšti ne tik patys save, bet ir būti kitų poveikių sąlyginiais ženklais, inicijuojančiais pastaruosius, t.y. būti ne tik pirminiais poveikiais, bet ir antriniais signalais. Tokie **antriniai signalai ir antriniai poveikiai** vadinami **informaciniais poveikiais**.* Informaciniai poveikiai gali būti būdingi ir techninėms sistemoms, sudarytoms iš pasyviųjų elementų. Todėl organizacijose egzistuoja ne tik energetiniai ir medžiaginiai, bet ir **informaciniai srautai**. Yra organizacijų, o taip pat ir techninių sistemų, kuriose informaciniai srautai yra pagrindiniai srautai. Tokių organizacijų tipui priklauso ir Statistikos departamentas, o tokios technikos tipui – kompiuterinė technika. Informaciniai srautai, būdami antriniais, žinoma, negali egzistuoti be pirminių srautų – antrinių signalų nešėjų, tačiau informaciniuose srautuose jų vaidmuo yra pagalbinis.

Aktyvieji elementai skiriasi nuo pasyviųjų dar ir tuo, kad jie turi **nuosavus poreikius**. Tik tenkindami juos, aktyvieji elementai ir egzistuoja kaip aktyvieji elementai. **Nuosavų poreikių turėjimas ir jų tenkinimas yra vienas pagrindinių gyvybės požymių, todėl juos turi ir juos stengiasi tenkinti visos gyvos būtybės ir iš jų sudarytos sistemos, tame tarpe – ir žmonės bei jų sukurtos organizacijos**. Žmonių ir jų sudarytų organizacijų poreikiai gali turėti medžiaginę, energetinę ir informacinę prigimtį, todėl organizacijose ir egzistuoja medžiaginiai, energetiniai ir informaciniai srautai.

Organizacijos ir žmonės gali tenkinti savo poreikius ne tik savo veiklos rezultate, bet ir naudotis kitų organizacijų bei kitų žmonių **paslaugomis**. Paslaugos gali turėti energetinę,

* Apie informacinius poveikius ir valdymo procesus žiūr. taip pat dar 54-58 šios metodinės priemonės puslapius.

medžiaginę, informacinę ir mišrią prigimtį. Galima kalbėti ir apie **paslaugų srautus** kaip **veiklos srautus**, egzistuojančius tarp veiklos subjektų ir veiklos objektų.

Projektuojant organizacijose turinčius vykti procesus ir jų TNOD, tenka projektuoti ne tik atskirų veiklos subjektų veiklą, bet ir jų sąveikavimą – juos jungiančius informacinius, energetinius ir medžiaginius srautus, teikiamas vienas kitam paslaugas.

Vieno veiklos subjekto kitam veiklos subjektui teikiamos paslaugos, o taip pat ir medžiagos, energija ir informacija, reikalauja subjektų veiklos ir turi jiems tam tikrą **vertę** subjektų poreikių ir tų poreikių patenkinimui reikalingos veiklos kiekio atžvilgiu, kuri nebūtinai vienoda kiekvienam iš subjektų. Tačiau kiekvienas veikos subjektas, besikeisdamas su kitu subjektu paslaugomis, medžiagomis, energija bei informacija, stengiasi nepralošti vertėje: norėdamas gauti tai, ko jam trūksta nuosavų poreikių tenkinimui, jis stengiasi atiduoti kitiems subjektams iš to, ką jis dabar turi, tiek, kad atiduodamoji vertė nebūtų jam didesnė už gaunamąją. Taip atsiranda **mainų santykiai**. Mainų santykiai yra naudingi abiem subjektams, dalyvaujantiems mainuose, nes kiekvienas iš jų mainais gauna ne mažesnę (o dažniausiai – ir didesnę) **sau vertę**.

Kadangi veiklos subjektas, norėdamas ką tai gauti mainais iš kito veiklos subjekto, ne visada turi tai, ko būtent reikėtų tam antrajam subjektui, tai atsiranda **sudėtingų mainų santykiai**. Tokių santykių palengvinimui ir supaprastinimui buvo išrastas vertės ekvivalentas – **pinigai**, ir tokiu būdu patobulinti mainų santykiai tapo **rinkos santykiais**. Todėl, nagrinėjant procesus, vykstančius organizacijose ir tarp jų, bei projektuojant TNOD, tenka kartu su jau minėtais srautais nagrinėti ir projektuoti taip pat ir **pinigų srautus**, nagrinėti vartojamos energijos, medžiagų, informacijos bei paslaugų **kainas, nagrinėti jų kitimą**.

Aprašant bei nagrinėjant tokių organizacijų bei jų konglomeratų kaip valstybės, jų regionai, miestai ir kitos gyvenvietės, valstybės ūkio šakos, įmonės, įstaigos, visuomeninės, profesinės ir politinės organizacijos, religinės koncesijos, socialiniai gyventojų sluoksniai ir t.t. sąveikavimą, tenka nagrinėti dar ir **gyventojų srautus** (gyventojų migraciją, jų gimimus, mirimus bei kitus demografinius pokyčius, **darbo jėgos judėjimą** iš vienos sferos į kitą, **gyventojų užimtumo, nedarbo bei bedarbystės lygio kitimą ir pan.**). Su analogiškais srautais susiduriama ir nagrinėjant **ekosistemų (įvairių arealų ir juose paplitusių populiacijų) sąveikavimą**, nes migruoti, gimti ir žūti bei kitaip išnykti gali ne tik žmonės, bet ir kiti įvairių arealų „gyventojai“ (gyvūnai, augalai).

Nagrinėjant ir projektuojant srautus (medžiagų, energijos, informacijos, paslaugų, pinigų, gyventojų ir t.t.), nustatoma tų **srautų sudėtis ir struktūra**, tų srautų judėjimo **maršrutai erdvėje ir laike**, srautų **greičiai ir galingumai**, srautų egzistavimui bei jų palaikymui reikalingų **kanalų ir bunkerių** charakteristikos (pavyzdžiui, informacinių srautų egzistavimui bei palaikymui reikalinga technika, informacinių bankų apimtys, jų sudėtis ir struktūra). Patys srautai, nepriklausomai nuo jų prigimties, gali būti vaizduojami **srautų informaciniais modeliais**, kurie gali būti deskriptyviniais ir normatyviniais. Vaizduojant ir projektuojant srautus, gana dažnai naudojamosi grafinė tų modelių forma – **srautų diagramomis**.

Kadangi ši metodinė priemonė skirta SIS projektuotojams ir SIS projektavimo koordinacinės tarybos nariams, kuriems aktualiausi yra informaciniai procesai, informaciniai bankai ir informaciniai srautai, atspindintys visus kitus srautus Lietuvos valstybėje, jos regionuose bei ūkio šakose, tai mes, aprašydami veiklos procesų ir jų TNOD projektavimą, pagrindinį dėmesį skirsime, būtent, informaciniams srautams ir informaciniams bankams.

Vienas pagrindinių Lietuvos statistinės sistemos tobulinimo ir SIS kūrimo klausimų yra klausimas apie Statistinėje sistemoje kauptiną, saugotiną ir apdorotiną **informacijos turinį**. Pakankamai pagrįstą ir pakankamai pilną atsakymą į šį klausimą galima gauti tik naudojantis normatyvinio ir deskriptyvinio metodų sinteze.

Deskriptyvinio metodo esmę, projektuojant SIS informacinę bazę, sudaro statistinių sistemų, egzistuojančių įvairiose valstybėse, analizė, jų sukauptos patirties

transformavimas, pritaikant Lietuvos sąlygoms. Susipažinimui su kitose valstybėse sukaupta statistinių sistemų darbo patirtimi Statistikos departamento ir jo tarnybų vadovai skiria gana daug dėmesio, o be to Statistikos departamente jo kvietimu gana dažnai lankosi kitų valstybių specialistai, kurie konsultuoja departamento darbuotojus Statistinės sistemos darbo tobulinimo ir SIS kūrimo klausimais. Visą šią gana plačią, tačiau šiek tiek epizodišką veiklą būtų naudinga griežčiau suplanuoti, pajungiant ją SIS kūrimo tikslams ir sujungiant į sistemiškai suderintą iš vieno centro koordinuojamą procesą, kuriame kiekvienas vadovas ar konsultantas taptų atsakingas už kokių nors konkrečiai apibrėžtų problemų sprendimą. Tai padidintų vykstančių į komandiruotes užsienin atsakomybę ir pačių komandiruočių efektyvumą, o taip pat atvykstančių iš užsienio specialistų konsultacijų konkretumą ir naudingumą. Jau dabar būtų galima pateikti SIS kūrimo Koordinacinei tarybai vieną – du – tris valstybinės statistinės sistemos organizavimo projektus, kuriuos apsvarsčius kaip galimas alternatyvas, būtų galima pasirinkti pakankamai racionalią ir perspektyvią SIS kūrimo strategiją.

Deskriptyvinis SIS informacinės įrangos projektavimo metodas, nors ir duoda daug naudingos informacijos ir padeda taupyti laiką ir lėšas, „neišradinėjant dviračio“, o iškart pasinaudojant kitų valstybių jau sukaupta patirtimi, vis dėlto negali garantuoti, kad, naudojantis šiuo metodu, bus sukurta tokia SIS informacinė įranga, kuri pilnai atitiks specifinius Lietuvos poreikius. Juk Lietuva savo ištekliais, gamtinėmis sąlygomis, savo pramone ir žemės ūkiu, savo visuomenės mentalitetu ir daugeliu kitų požymių skiriasi nuo Norvegijos, Švedijos ar Danijos ir tuo labiau – nuo Didžiosios Britanijos ar JAV. Todėl ne viskas, kas tinka ar reikalinga, pavyzdžiui, Didžiajai Britanijai ar Vokietijai, būtinai tinka ar reikalinga Lietuvai ir atvirkščiai.

Dėl šios priežasties, **nustatant informacinių srautų ir informacinių bankų turinį, nepakanka vien tik deskriptyvinio metodo, ir todėl SIS informacinės įrangos projektą, gautą, naudojantis deskriptyviniu metodu, tenka koreguoti ir papildyti, naudojantis normatyviniu metodu ir abiejų šių metodų sintezę.**

Kadangi valstybinės statistikos sistemos paskirtis – rinkti, kaupti, apdoroti, saugoti ir platinti statistinę informaciją, reikalingą informacinių modelių apie įvairias valstybės (jos regionų, jos ūkio šakų ir t.t.) gyvenimo sritis bei vystymosi kryptis sudarymui, kuriais galėtų naudotis įvairios valstybinės įstaigos, tarptautinės, visuomeninės, politinės ir t.t. organizacijos, įvairios įmonės, firmos, ir pan., priimdamos tuos ar kitus svarbius sprendimus, tai, **nustatant informacijos, kurią turėtų kaupti valstybinė statistikos sistema, turinį, būtina žinoti aukščiau išvardintų statistinės informacijos vartotojų informacinius poreikius. Geriausiai tuos poreikius privalėtų žinoti patys vartotojai. Todėl, nustatant kaupiamos statistinės informacijos ir tos informacijos srautų turinį normatyviniu metodu, būtina bendradarbiauti su tos informacijos vartotojais.** Kitaip tariant, normatyvinio metodo taikymas, nustatant statistinės informacijos turinį, išveda SIS projektuotojus už statistinės sistemos ribų, tuo pačiu įrodydamas, kad **SIS yra sudėtinė LVII ir VADIS dalis ir turi būti projektuojama kartu su visa VADIS bendromis VADIS ir SIS projektuotojų bei Vyriausybės ir kitų valstybinių įstaigų vadovų bei darbuotojų jėgomis.** Tik suprojektavus tokių sudėtingų organizacijų, kaip valstybė ir jos posistemiai, valdymo sistemas (jų funkcines ir objektines struktūras, jų funkcionavimo algoritmus), galima pakankamai pagrįstai ir pakankamai tiksliai nustatyti Vyriausybės ir kitų valstybinių įstaigų informacinius poreikius ir sukurti racionalią ir efektyvią valstybinės statistikos sistemą. Tai – milžiniškas darbas. Tačiau jis jau pradėtas, nes jau pradėti SIS ir VADIS kūrimo darbai, ir todėl turėtų būti vykdomas sistemiškai, sudarius bendrą suderintą visų VADIS kūrimo (tame skaičiuje ir SIS kūrimo) darbų planą, sukūrus bendrą visus darbus koordinuojantį tiesiogiai Vyriausybei pavaldų padalinį, sudarytą iš aukštos kvalifikacijos specialistų – įvairių valstybinių bei mokslo institucijų atstovų. Dabar gi, šių eilučių autoriaus nuomone, VADIS (ir SIS) kūrimo darbas vykdomas nepakankamai sistemiškai ir net pats VADIS projektuotojų kolektyvas organizuojamas, vadovaujantis ne tiek bendravalstybiniais, kiek grupiniais

interesai. Yra rimtas pavojus, kad VADIS kūrimas, kaip ir daugelis kitų valstybinių programų, gali tapti tam tikrų grupuočių pasipelnymo šaltiniu ir neduoti tų rezultatų, kurių galima būtų laukti, realizuojant gana perspektyvią VADIS sukūrimo idėją. Šių eilučių autorius mano, kad Statistikos departamento vadovybė ir SIS projekto vadovai privalėtų ryžtingiau išreikšti savo poziciją VADIS kūrimo atžvilgiu atitinkamose Vyriausybės institucijose ir **pareikalauti, kad būtų sukurtas kompleksinis kompetetingų VADIS projektuotojų kolektyvas, į kurį SIS projektuotojai įeitų ne tik kaip eilinė, bet ir kaip viena iš inicijuojančių (vedančiųjų) sudėtinių dalių.** Tai būtų visai natūralu, nes tokioje sistemoje kaip VADIS Statistinės informacijos sistema (SIS) turėtų užimti vieną pačių svarbiausių vietų – pagrindinio informacinio banko ir pagrindinio informacinio tiekėjo visoms valstybinėms įstaigoms ir tarptautinėms organizacijoms vietą. Čia iškelta SIS ir VADIS projektuotojų bendradarbiavimo organizavimo problema turėjo būti išspręsta jau pačioje VADIS ir SIS kūrimo pradžioje, ir dabar ją reikėtų spręsti nedelsiant. Kuo anksčiau prasidės VADIS ir SIS projektuotojų bendradarbiavimas plataus VADIS projektuotojų kolektyvo sudėtyje, tuo produktyvesnis bus ir VADIS ir SIS kūrėjų darbas. SIS kūrimas galėtų ir turėtų tapti viena prioritetinių VADIS kūrimo krypčių, o SIS kūrėjai ir jų sukurti projektai galėtų būti pavyzdžiais toms VADIS projektuotojų grupėms, kuriose specialistų darbo su kompiuterizuotomis informacinėmis sistemomis patirtis yra mažesnė.

Valstybinė statistikos sistema, privalėdama tenkinti Vyriausybės ir kitų valstybinių įstaigų bei oficialių tarptautinių organizacijų informacinius poreikius, galėtų dar ir prekiauti statistine informacija, specialiai rinkdama, apdorodama ir parduodama įvairioms Lietuvos ir užsienio firmoms bei kitoms pelno siekiančioms organizacijoms jas dominančią informaciją. Todėl, nustatydami informacinių srautų ir informacinių bankų statistikos sistemoje turinį, **SIS projektuotojai privalėtų išnagrinėti ir statistinės informacijos paklausą informacijos rinkoje.** Tokio tipo tyrimai jau neįeitų į VADIS kūrimo planą ir, matyt, negalėtų tikėtis valstybinio finansavimo. Tačiau jie galėtų būti numatyti SIS kūrimo planuose ir galėtų būti vykdomi „ūkiskaitos“ sąlygomis: išlaidos, padarytos tiriant informacijos rinką ir projektuojant bei diegiant prekybos statistine informacija sistemą, turėtų būti sugrąžintos kreditoriams ne vėliau, kaip per penkis – šešis metus nuo sistemos funkcionavimo pradžios. Tokios sistemos sukūrimą galėtų kredituoti ne tik valstybė, bet ir būsimieji statistinės informacijos pirkėjai.

Nustatant informacinių srautų ir informacinių bankų turinį, visų pirma reikia pasirinkti elementarų to turinio vienetą. Statistinės informacijos turinio vienetu galėtų būti pasirinktas rodiklis.

Rodiklį sudaro jo pavadinimas ir jo reikšmė. **Rodiklio reikšmė** – tai ko nors kiekybinis ar kokybinis įvertinimas. Statistinės informacijos sistemose tie įvertinimai dažniausiai būna kiekybiniai ir išreiškiami kokiais nors skaičiais – skaičiavimo, matavimo ar kokio nors kitokio įvertinimo rezultatais. Žinoma, galimi ir kokybiniai įvertinimai.

Rodiklių **reikšmių vertinimui** gali būti naudojamos keturių tipų **matavimo skalės**: 1) nominalinės (pavadinimų), 2) rangų (tvarkos), 3) intervalinės ir 4) santykių (reliacinės). Pirmųjų dviejų tipų matavimo skalės taikomos kokybiniais rodikliams, o paskutiniųjų dviejų – kiekybiniais.

Nominalinė skalė, vadinama dar **pavadinimų** bei **klasifikavimo** skale, leidžia tik **atskirti** rodiklio reikšmes vieną nuo kitos ir **klasifikuoti** jas, priskiriant tai ar kitai **kategorijai** (populiacijos poaibiui, skyriui, klasei, grupei ir pan.). Tų reikšmių nei palyginti, nei atlikti su jomis kokias nors aritmetines operacijas (net tada, kai tos reikšmės koduojamos skaičiais) pavadinimų skalė neleidžia. Nominalinei skalei priklauso, pavyzdžiui, tokių rodiklių kaip „tautybė“, „lytis“ ir pan. reikšmės: „lietuvis“, „vokietis“, „anglas“, „rusas“, „lenkas“ ir t.t.; „vyras“, „moteris“.

Rangų (tvarkos, prioritetų) skalė leidžia rodiklių reikšmes ne tik atskirti vieną nuo kitos bei klasifikuoti, bet ir palyginti jas vieną su kita ir to palyginimo pasekmėje tvarkyti (rikiuoti) objektus tos reikšmės (rango, prioriteto) didėjimo ar mažėjimo tvarka. Rodiklio reikšmių, išmatuotų rangų skalėje, **statistinė eilutė** jau galima sutvarkyti į **variacinę**

eilutę. Rangų skalei priklauso, pavyzdžiui, tokių rodiklių kaip „sporto rungtynėse ar grožio konkurse užimta vieta“, „mokslo laipsnis“, „mokinio pažymys“ ir pan. reikšmės: „pirmoji“, „antroji“, „trečioji“ ir t.t. vieta; „bakalauro“, „magistro“, „daktaras“, „habilituotas daktaras“, „puikiai“ („10“), „labai gerai“ („9“), „gerai“ („8“) ir t.t. Su rangais, kai jie išreikšti skaičiais, tam tikrais atvejais jau galimos aritmetinės operacijos (pavyzdžiui, surinktos balų sumos, vidutinio balo apskaičiavimas ir pan.).

Intervalų skalė jau leidžia rodiklių reikšmes ne tik klasifikuoti ir tvarkyti, bet ir kiekybiškai vertinti rodiklio reikšmių skirtumus. Nors intervalų skalėje ir pradinis atskaitos taškas (nulinis taškas), ir matavimo vienetas gali būti laisvai pasirenkami, tačiau dviejų tos skalės intervalų santykis yra tiems intervalams būdingas konkretus dydis, nepriklausantis nei nuo nulinio taško, nei nuo matavimo vieneto pasirinkimo. Su dydžiais, išmatuotais intervalų skalėje galima atlikti tokias aritmetines operacijas: sudėti, atimti, daugyti iš skaičiaus, dalyti iš skaičiaus. Intervalų skalėje matuojama temperatūra (Celsijaus, Farenheito, Reomiūro skalės), kalendorinis laikas (metų skaičiavimas: nuo Kristaus gimimo, nuo Mahometo išvykimo į Mediną, nuo Romos įkūrimo ir pan.), intelekto koeficientas IQ ir t.t.

Santykių skalė skiriasi nuo intervalų skalės tik tuo, kad joje yra apibrėžta absoliuti atskaitos pradžia – absoliutus nulio taškas, rodantis, kad tiriamasis objektas matuojamojo požymio visiškai neturi. Skaičių, gautų matuojant požymius, santykis, parodo, kiekybinį matuojamojo požymio santykį, nepriklausantį nuo pasirinktų matavimo vienetų. **Su dydžiais, išmatuotais santykių skalėje, galimos visos matematinės operacijos.** Santykių skalėje matuojamas ilgis, svoris, greitis, įtampa, srovės stiprumas, gyventojų skaičius, pajamos, pelnas ir pan.

Rodiklio pavadinimas identifikuoja tai, kas (o kartais ir lyginant su kuo) yra vertinama. **Rodiklio pavadinimas kartu su nurodyta (išreikšta skaičiumi ar kaip kitaip) konkrečia to rodiklio reikšme sudaro tai, ką matematinėje logikoje būtų galima pavadinti teiginiu, kuris, kai rodiklio reikšmė konkrečiai dar nenurodyta, galėtų būti interpretuojamas kaip predikatas.** Tuomet rodiklio pavadinimas, tampa kokio tai kintamojo žymėjimu, su kuriuo galima operuoti kaip ir su bet kuriuo matematikoje vartojamų kintamųjų žymėjimu.

Rodiklio pavadinimą kaip kokio nors kintamojo žymėjimą sudaro žodžiai bei žodžių grupės, žyminčios esybes, jų atributus bei santykius. Tie žodžiai (ar žodžių grupės) vadinami **rodiklio rekvizitais**. Rodiklio rekvizitas, žymintis tą esybę (tai gali būti ir kokios nors kitos esybės atributas bei kokių nors esybių ar atributų santykis, interpretuojamas kaip atskira esybė), kurią išreiškia, matuoja, vertina rodiklio reikšmė, vadinamas **rodiklio pagrindu**, o visi kiti rekvizitai, sudarantys rodiklio pavadinimą, – **rodiklio požymiais**. Naudojantis rodiklių kaip kintamųjų žymėjimais matematinėse formulėse, rodiklių pagrindai dažniausiai žymimi raidėmis (bei jų grupėmis), o rodiklių požymiai – tų raidžių indeksais, kurie savo eile gali būti žemutiniais arba viršutiniais. **Viršutiniai indeksai** dažniausiai naudojami **rodiklio tipo** žymėjimui, o **žemutiniai** – rodiklio požymių priklausančių tam tikroms požymių klasėms, kaip tų **klasių elementų (bei tų elementų vietų klasėse)** žymėjimui (numeravimui). Pavyzdžiui, rodiklio pavadinime „vidutinė vieno kilogramo kiaulienos kaina Panevėžyje 1996 m. vasario mėn. (litas)“ žodis „kaina“ yra šio rodiklio pagrindas, o visi kiti rekvizitai – to pagrindo požymiai. Rekvizitai „kiaulienos“, „Panevėžyje“, „1996 m. vasario mėn.“ nurodo konkrečius požymius, tų požymių „vietas“ („numerius“) požymių klasėse, nusakančiose rodiklio pagrindo objektą ir jo vietą erdvėje ir laike, t.y. tų konkrečių požymių „vietą“ analogiškų požymių sąrašuose (pavyzdžiui, požymio „kiauliena“ „vietą“ mėsos rūšių sąraše: „aviena“, „jautiena“ ir t.t.; požymio „Panevėžys“ „vietą“ Lietuvos miestų sąraše: „Alytus“, „Biržai“ ir t.t.; požymio „1996 m. vasario mėn.“ „vietą“ laiko intervalų sąraše). Todėl šių rekvizitų identifikatoriai (numeriai ar kokie nors kiti žymėjimai) turėtų būti raidės, vaizduojančios žodį „kaina“, žemutiniais indeksais, o vaizduojant rodiklių reikšmes daugiamačiuose duomenų masyvuose, atitikti tų masyvų „koordinačių“ („eilučių“ x ir y ,

„stulpelių“ z ir t.t.) reikšmes. Rekvizitai „vidutinė“, „vieno kilogramo“, „litas“ nurodo rodiklio pagrindo „kaina“ tipą, specifikuoja pagrindą, t.y. nurodo, kad ta kaina yra vidutinė, kad ji išreikšta litais tokiam kiekiui vienetai, kuris vadinamas kilogramu. Šie rekvizitai charakterizuoja patį pagrindą, o ne tuos objektus, kuriuos charakterizuoja nagrinėjamas rodiklis, ir todėl šiuos rekvizitus būtų galima prijungti prie rodiklio pagrindo kaip to pagrindo patikslinimą. Šių rekvizitų identifikatoriai (pavyzdžiui, kodai) turėtų būti tos raidės, kuri vaizduoja rodiklio pagrindą, viršutiniais indeksais, o vaizduojant rodiklį reikšmės duomenų masyvais, **charakterizuoti ne masyvų elementus ir jų vietas masyvuose, o pačius masyvus ir įeiti į tų masyvų pavadinimus** (ar į jų kodus) kartu su rodiklio pagrindo pavadinimu.

Kitaip tariant, tie rekvizitai – požymiai, kurie turėtų būti žymimi rodiklio pagrindo žemutiniais indeksais, yra nagrinėjamų **objektų bei procesų aibės klasifikavimo požymiai**, tos aibės, vaizduojamos daugiamatės erdvės taškais, **taškų koordinatės**, o tų rekvizitų klasės – tos **erdvės koordinačių ašys**, leidžiančios nustatyti kiekvieno objekto bei proceso vietą jų aibėje. Rekvizitas – pagrindas ir tie rekvizitai – požymiai, kurie turėtų būti vaizduojami rodiklio pagrindo viršutiniais indeksais, nurodo minėtosios **objektų bei procesų aibės pavadinimą ir jos analizės bei vertinimo aspektą**, matavimo bei kitokio vertinimo vienetus ir būdus, t.y. **identifikuoja atitinkamą aibę ar tos aibės ir jos elementų pjūvį**, vaizduojantį atitinkamas tų elementų savybes bei jų santykius.

Rekvizitus, nusakančius rodiklio reikšmės matavimo vienetus – rodiklio dimensiją, neretai perkelia iš rodiklio pavadinimo į rodiklio reikšmę, t.y. vietoj predikato „vidutinė vieno kilogramo kiaulienos kaina litais Panevėžyje 1996 m. vasario mėn.“ = x “ vartoja predikatą „vidutinė kiaulienos kaina Panevėžyje 1996 m. vasario mėn.“ = x Lt/kg“.

Įstatę vietoj x jo reikšmę 13,98, gautume teiginį „vidutinė kiaulienos kaina Panevėžyje 1996 m. vasario mėn.“ = 13,98 Lt/kg“.

Toks dimensijos perkėlimas turi prasmę, nes adityvines matematines operacijas (sudėtį, atimtį), galima vykdyti tik su tokiomis rodiklių reikšmėmis x , kurios turi vienodas dimensijas, o vykdant multiplikatyvines matematines operacijas (daugybą, dalybą), su rodiklių dimensijomis operuojama lygiai taip pat, kaip ir su rodiklių reikšmėmis. Pavyzdžiui, apskaičiuojant stačiakampės prizmos tūrį, dauginamos ne tik trijų tos prizmos matavimų reikšmės x , y , z , bet ir jų dimensijos, ir todėl tokio rodiklio kaip „tūris“ dimensija yra jau ne metrai, centimetrai ar milimetrai, o kubiniai metrai (m^3), kubiniai centimetrai, kubiniai milimetrai. Dalinant vieno rodiklio reikšmę iš kito rodiklio, turinčio tą pačią dimensiją, reikšmės, gaunamas rodiklis, turintis nulinio laipsnio dimensiją, t.y. vienetai. Tokie santykiniai rodikliai, turintys „nulinę dimensiją“, t.y. matuojami bedimensiniais vienetais, gana dažnai vartojami statistinėje informacijoje. Viena tokių rodiklių rūšių **aprašomojoje statistikoje** vadinama **indeksais**: pavydžiui, VKI (vartojimo kainų indeksas), HDI (Human Development Index – žmogaus socialinės raidos indeksas) ir kiti. **Čia pavartoto žodžio „indeksas“, reiškiančio tam tikros rūšies statistinį rodiklį, prasmę reikia skirti nuo rodiklio pagrindo žymėjimo viršutiniojo bei žemutiniojo indekso, reiškiančio kokio nors kintamojo požymį, prasmę**. Toks žodžių daugiaprasmiškumas (homonimų egzistavimas) sudaro tam tikrų nepatogumų, tačiau **su homoniškumo reiškiniais tenka susidurti gana dažnai (net pats žodis „rodiklis“ aprašomojoje statistikoje ir matematikoje reiškia ne tą patį)** ir apie žodžio – homonimo prasmę tenka spręsti iš konteksto.

Pervedus rekvizitus, nusakančius rodiklio reikšmės matavimo vienetus, iš rodiklio pavadinimą ir jo reikšmę išreiškiančios lygybės kairiosios pusės į dešiniąją, likusius rodiklio pavadinimo rekvizitus, atitinkančius viršutinius jo žymėjimo indeksus, galima prijungti prie rodiklio pavadinimo pagrindo, sudarant taip vadinamą **specifikuotą pagrindą**, identifikuojantį ne tik rodiklio pagrindą atitinkančią esybę, bet ir jos tipą. **Tuomet rodiklio pavadinimo žymėjimą sudarys specifikuoto pagrindo žymėjimas, vaizduojantis tam tikro tipo rodiklių aibės (masyvo) pavadinimą, ir jo žemutiniai indeksai, nusakantys konkretaus rodiklio vietą minėtoje aibėje (masyve).**

Aprašant bei žymint rodiklius, kintamaisiais galima laikyti ne tik rodiklių reikšmes, bet ir rodiklių pavadinimų žymėjimų žemutinius indeksus, galinčius turėti tas ar kitas reikšmes iš atitinkamų sąrašų – registrų. Tuomet rodiklių pavadinimų žymėjimai vaizduos jau nebe atskirus konkrečius rodiklius, o jų klases. Pavyzdžiui, jei rodiklio „vidutinė kiaulienos kaina Panevėžyje 1996 m. vasario mėn.“ žymėjime rekvizito „Panevėžyje“ konkretų žymėjimą (kokį tai kodą ar patį žodį „Panevėžyje“) pakeistume kintamuoju v , galinčiu turėti reikšmes, priklausančias Lietuvos miestų sąrašui, tai gautume nebe vieną konkretų rodiklį, o tam tikrą tokių rodiklių klasę, sudarančią rodiklių vektorių, kurio koordinačių reikšmės būtų lygios kiaulienos vidutinėms kainoms 1996 m. vasario mėn. įvairiuose Lietuvos miestuose, žymimuose kintamojo v reikšmėmis. Papildomai pakeitę rekvizitą „kiaulienos“ kintamuoju μ , galinčiu turėti reikšmes iš įvairių mėsos rūšių sąrašo, gautume žymėjimą, vaizduojantį rodiklių klasę, sudarančią matricą, turinčią tiek eilučių, kiek miestų yra Lietuvoje, ir tiek stulpelių, kiek rūšių mėsos minima parduodamos mėsos sąrašuose. Pakeitę rekvizitą „1996 m. vasario mėn.“ kintamuoju τ , galinčiu turėti reikšmes iš tam tikro tam tikrų laiko intervalų sąrašo (pavyzdžiui, iš 1994÷1996 m.m. mėnesių sąrašo), gautume jau trimatį rodiklių masivą, vaizduojantį vidutinės mėsos kainas Lietuvos miestuose 1994÷1996 m.m. mėnesiais. **Žymėjimą $(v.k)_{\mu,v,\tau}$ (arba $\bar{k}_{\mu,v,\tau}$), kur $(v.k)$ arba $\bar{k}$ yra specifikuoto pagrindo „vidutinė kaina“ žymėjimas, galima laikyti priklausomo kintamojo $(v.k)$ (arba $\bar{k}$), priklausančio nuo nepriklausomų kintamųjų μ , v ir τ , žymėjimu, kitaip tariant, funkcijos $\bar{k}$, priklausančios nuo argumentų μ , v ir τ , žymėjimu.** Šios funkcijos reikšmių lentelė kaip tik ir sudaro jau minėtąjį trimatį vidutinių mėsos kainų Lietuvos miestuose 1994÷1996 m.m. mėnesiais masivą.

Įstatę į funkciją $\bar{k}_{\mu,v,\tau}$ vietoj jos argumentų μ , v ir τ konkrečias jų reikšmes, pavyzdžiui, $\mu^* = \text{kiauliena}$, $v^* = \text{Panevėžys}$, $\tau^* = \text{1996 m. vasario mėn.}$, gautume konkrečią tos funkcijos reikšmę $\bar{k}_{\mu^*,v^*,\tau^*} = 13,98 \text{ Lt/kg}$ – vieną trimačio kainų masyvo elementą.

Funkcijos $\bar{k}$ argumentų μ , v ir τ apibrėžimo sritis galima išplėsti, o išplėtus – struktūrizuoti, struktūrizuojant argumentų μ , v ir τ reikšmių sąrašus bei tų reikšmių kodus. Pavyzdžiui, argumento μ apibrėžimo sritį galima išplėsti, įtraukiant į ją ne tik mėsos, bet ir kitų produktų rūšis ar net įvairių kitų prekių rūšis. Struktūrizavus prekių sąrašą ir įvedus įvairių prekių klasių sąvokas bei jų kodus ar žymėjimus, kintamąjį μ galima paversti sudėtingu kintamuoju, priklausančiu nuo paprastesnių kintamųjų, turinčių mažesnius jų galimų reikšmių sąrašus. Toks struktūrizavimas palengvintų ir pagreitintų mus dominančios prekių rūšies vidutinės kainos paiešką prekių vidutinių kainų masyve bei to masyvo rodiklių atitinkamų klasių ir poklasių išskyrimą. Analogiškai galima išplėsti (ar susiaurinti) ir kintamojo v apibrėžimo sritį, pavyzdžiui, papildant Lietuvos miestų sąrašą kai kuriais Latvijos, Estijos, Baltarusijos ir Lenkijos ar net ir tolimesnių valstybių miestais. Kintamojo v reikšmių sąrašą irgi galima struktūrizuoti, o ir patį kintamojo v žymėjimą pakeisti sudėtingesniu struktūrizuotu žymėjimu. Gali būti struktūrizuota ir išplėsta ar susiaurinta ir kintamojo τ apibrėžimo sritis, o taip pat struktūrizuotas ir pats to kintamojo žymėjimas, suskaidant tą žymėjimą į kelių kintamųjų žymėjimus, kurių pirmasis žymėtų metus, antrasis – jų mėnesius, trečiasis – jų savaites ar dienas.

Rodiklių pavadinimus sudarančių rekvizitų klasifikavimu ir rodiklių ir jų aibių pavadinimų bei pačių tų aibių struktūrizavimu naudojamosi, sudarant informacinių bankų ir informacinių srautų, o taip pat ir atskirų pranešimų apie rodiklių reikšmes, projektus.*

Pranešimo, pateikto standartizuoto dokumento forma (spausdinto, rašytinio, pavaizduoto displėjaus ekrane), turinį dažniausiai sudaro ne vienas rodiklis, o jų grupė, kurioje rodiklių rekvizitai arba sutampa, arba priklauso toms pačioms klasėms. Todėl tokiuose

* Su įvairiomis rodiklių reikšmių klasifikavimo sistemomis galima susipažinti Statistikos departamento klasifikacijų poskyryje.

dokumentuose gana dažnai naudojamosi matricine informacijos pateikimo forma, suteikiant matricos stulpeliams ir eilutėms minėtųjų klasių elementų pavadinimus, o į matricos langelius įrašant tik pačias atitinkamų rodiklių reikšmes. Išskėlę į dokumento pavadinimą tuos rekvizitus, kurie yra bendri rodiklių grupėms, o taip pat nurodę rodiklių reikšmių dimensijas (pačiame pavadinime ar specialiai išskirtame stulpelyje), gauname gana kompaktišką pranešimą apie tų ar kitų rodiklių reikšmes.

Žinoma, pranešimus galima teikti ne tik informacinių **failų**, bet ir **objektinių informacinių modelių pavidalu***, kuriuos taip pat galima daugiau ar mažiau unifikuoti. Pranešimo forma pasirenkama priklausomai nuo pranešimo paskirties.

Iš to, kas buvo pasakyta apie rodiklius ir jų aprašymą bei žymėjimą, reikėtų padaryti dvi svarbias išvadas:

- 1) aprašant rodiklius ir įvedant jų žymėjimus, nepakanka apytikriai išsiaiškinti rodiklio pavadinimą sudarančių rekvizitų prasmę, o **reikia apibrėžti** ir tų kintamųjų, kurių reikšmėmis yra rodiklio pavadinimą sudarantys rekvizitai, galimo (ar leistino) **kitimo sritis**, reikalui esant, **struktūrizuoti jas**, įvedant požymių klases ir poklasių, rodiklių pagrindų tipus;
- 2) padarę tai, ko reikalauja pirmoji išvada, gautume ne tik informacinių srautų bei informacinių bankų turinio, bet ir jų struktūros aprašymus, nes tokioje struktūrizuotoje rodiklių rekvizitų ir jų žymėjimų sistemoje rodiklio rekvizitai nusakytų **ne tik rodiklio turinį, bet ir jo vietą informaciniuose bankuose ir informaciniuose srautuose, rodiklio santykį su kitais rodikliais.**

Pirmoji išvada teigia, kad projektuojant bei analizuojant TNOD, reikia ne tik nustatyti funkcijų vykdymui reikalingus rodiklius, bet ir klasifikuoti juos, struktūrizuoti jų aibes, įvedant rodiklių ir jų masyvų bei klasių žymėjimus. Tai gana didelis ir sudėtingas darbas, kuris anksčiau, kuriant kompiuterizuotas informacines sistemas, užimdavo daug laiko. Dabartiniu metu, **pasitelkiant kompiuterinę techniką, galinčią operuoti ne tik su skaitmenine, bet ir su žodine bei grafine informacija, informacijos klasifikavimo, jos struktūrizavimo bei kodavimo darbai atliekami žymiai greičiau.** Projektuotojams pakanka tik nustatyti reikalingos informacijos turinį ir tiksliai apibrėžti jos rodiklius bei jų rekvizitų galimas reikšmes. Visa kita daro pats kompiuteris.

Antroji išvada išplaukia iš tos aplinkybės, kad, aprašydami rodiklius, jų tipus ir klases, vaizduodami rodiklius kaip jų požymių funkcijas, mes beveik automatiškai gauname informacinių srautų ir informacinių bankų turinio ir jų failinės struktūros aprašymus, nes rodiklių tipus atitinka rodiklių reikšmių masyvai, struktūrizuoti pagal požymius, atitinkančius tų rodiklių žymėjimų žemutinių indeksų klases.

Rodiklių pavadinimų bei jų žymėjimų gramatikos sudarymas ir rodiklių pavadinimus sudarančių rekvizitus atitinkančių kintamųjų leistinių (galimų) reikšmių sričių apibrėžimas leidžia nustatyti santykius tarp rodiklių ir projektuoti ne tik failines, bet ir **modelines** informacinių srautų ir informacinių bankų struktūras, t.y. projektuoti ne tik **duomenų bankus** ir **duomenų srautus**, bet ir taip vadinamus **žinių bankus** ir **žinių srautus**, kuriuose informacija pateikiama ne kaip pabiri duomenys, organizuoti į struktūrizuotus duomenų masyvus, o kaip objektų ir procesų modeliai (žiūr. šios priemonės ketvirtąjį skyrelį).

Modelinis informacijos organizavimo ir pateikimo būdas yra progresyvesnis, negu duomenų srautai ir duomenų bankai, nes žinių srautuose ir žinių bankuose esantys informaciniai modeliai leidžia nustatyti ne tik juose esančių rodiklių reikšmes, bet ir santykius tarp jų, t.y. leidžia ne tik rasti, bet ir išvesti (apskaičiuoti) reikalingą informaciją, kuri nebūtinai turi būti tiesiogiai įrašyta į informacinius bankus ir informacinius srautus. Ji gali ir išplaukti iš įrašytų į bankus bei srautus rodiklių reikšmių ir santykių tarp jų.

Projektuojant kompiuterizuotų informacinių sistemų informacinę įrangą, tenka atsižvelgti ir į tai, kuriame iš režimų informacija bus teikiama subjektams: reglamentuotame, signaliniame ar užklausimų (užklausų) režime.

* Pirmuoju pavidalu naudojamosi **duomenų bankuose (DB)**, o antruoju – **žinių bankuose**.

Reglamentuotu (reglamentiniu) režimu vadinamas toks informacijos tiekimo režimas, kuriam būdingos dvi savybės:

- 1) iš anksto nustatomi (reglamentuojami) informacijos tiekimo momentai (datos, o kartais net valandos ir minutės);
- 2) iš anksto nustatomi (reglamentuojami) kiekvienam reglamentuotam informacijos tiekimo momentui būdingi tiekiamos informacijos (rodiklių) sąrašai ir net tų rodiklių išdėstymo informaciniuose pranešimuose tvarka (dokumentų formos).

Reglamentuotu informacijos tiekimo režimu tikslinga naudotis tada, kada iš anksto žinoma, kam, kada ir kokios informacijos tikrai reiks, pavyzdžiui, aprūpinant informacija reglamentuotą funkcijų vykdytojus. Tuo režimu, pavyzdžiui, naudojamas, aprūpinant organizacijų ir jų padalinių vadovus informacija, reikalinga organizacijų bei jų padalinių būsenos ir aplinkos periodiniam įvertinimui, ataskaitų bei apžvalginių pranešimų sudarymui.

Informacinių sistemų veiklos analizė rodo, kad reglamentuoti pranešimai gana dažnai „perkraunami“ informacija: vartotojams pateikiami didžiuliai lapai ar net foliantai su daugybės rodiklių reikšmių lentelėmis, jų kitimo kreivėmis ir t.t. ne tik tada, kai visa ta informacija jiems tikrai reikalinga, bet ir tada, kada jie jos ne tik pilnai panaudoti, bet ir apžvelgti nebesusėja.

Tokiais atvejais žymiai efektyvesnis būtų **signalinis** informacijos tiekimo režimas. Jis gana plačiai taikomas techninėse reguliavimo ir automatinio valdymo sistemose, realizuojant taip vadinamą **grįžtamąjį (atbulinį) ryšį**, tačiau organizacijose dar nėra pakankami paplitęs.

Signaliniam režimui būdingos šios savybės:

- 1) signalinė informacija siunčiama tada, kada organizacijos ir jos padalinių funkcionavimo deskriptyvinių trajektorijų nuokrypiai nuo atitinkamų normatyvinių trajektorijų pasidaro didesni, negu iš anksto nustatytos normaliomis sąlygomis leistinos tų nuokrypių reikšmės;
- 2) jei deskriptyvinių trajektorijų nuokrypiai nuo normatyvinių trajektorijų išlieka leistinose ribose, signalinė informacija nesiunčiama;
- 3) daugiaėšeloninėse valdymo sistemose leistinų nuokrypių ribos yra skirtingos skirtingiems ešelonams ir didėja, kylant valdymo kontūrų piramide aukštn; tai reiškia, kad apie deskriptyvinių trajektorijų nuokrypius nuo normatyvinių trajektorijų anksčiau sužino žemesnieji ešelonai, o aukštesnieji ešelonai sužino tik tada, kai žemesnieji ešelonai nesugeba sustabdyti tų nuokrypių didėjimo;
- 4) automatizuotose signalinio informavimo sistemose kiekvienas valdymo ešelonas gauna jam priklausančią signalinę informaciją automatiškai, nepriklausomai nuo kitų ešelonų noro ar nenoro, kad tokia informacija būtų siunčiama ir gaunama;
- 5) signalinio informavimo sistemos funkcionavimui reikalingas ne tik deskriptyvinės informacijos rinkimas, bet ir normatyvinių funkcionavimo trajektorijų išankstinis nustatymas, o taip pat leistinų ribų deskriptyvinių trajektorijų nuokrypiams nuo normatyvinių trajektorijų nustatymas kiekvienam valdymo sistemos ešelonui;
- 6) signalinis informavimo režimas, padidindamas informaciniuose bankuose saugomos ir kompiuteriais apdorojamos informacijos apimtį, leidžia žymiai sumažinti informacinių srautų tarp organizacijos subjektų apimtį ir subjektų informacinį apkrovimą, nes gerai tvarkomose organizacijose signalinės informacijos nebūna daug;
- 7) signalinis informavimo režimas padeda organizacijos ir jos padalinių vadovams greičiau pastebėti nepageidautinus bei kitus ypatingus reiškinius ir imtis reikalingų priemonių, nelaukiant, kada tie reiškiniai taps pavojingais ar pražiopsotomis galimybėmis;
- 8) net pats signalinės informacijos srautų intensyvumo kitimas teikia signalinę informaciją: to intensyvumo mažėjimas signalizuoja, kad organizacijos funkcionavimas tampa stabilesnis, o to intensyvumo didėjimas signalizuoja organizacijos ir jos padalinių vadovams, kad organizacijos ir jos padalinių veikla

pradeda vis stipriau „šlubuoti“ ir kad reikalinga skubi ir gili to „šlubavimo“ priežasčių analizė bei radikalios priemonės tų priežasčių pašalinimui.

Signalinį informavimo režimą projektuoti ir diegti yra žymiai sunkiau, negu reglamentinį, tačiau toks režimas, sumažindamas valdymo sistemos subjektų informacinį apkrovimą, leidžia žymiai padidinti informacijos panaudojimo efektyvumą ir valdymo operatyvumą. Todėl signalinio informavimo sistemos kūrimo sąnaudos gana greitai atsiperka. Žinoma, signalinis informavimo režimas organizacijose negali pilnai pakeisti reglamentinio režimo, o tik papildo ir patobulina jį, sumažindamas reglamentuotos informacijos srautus ir reglamentuotų dokumentų apimtį bei jų pateikimo dažnius.

Abu šiuos informavimo režimus – reglamentinį ir signalinį – papildo ir dar labiau patobulina **užklausimų režimas**. Tai toks informavimo režimas, kai informacija tiekama vartotojui, tik jam užprašius. Užklausimų režimu naudojama, kai sprendimų priėmimui nepakanka turimos reglamentinės ir signalinės informacijos. Taip gali atsitikti, pavyzdžiui, gavus signalinę informaciją ir norint išsiaiškinti padidėjusių nuokrypių priežastis bei ieškant jų pašalinimo ar panaudojimo (jei tie nuokrypiai signalizuoja apie atsiradusias naujas galimybes) būdų. Užklausimų režimas taip pat gali būti reikalingas, kuriant normatyvinius modelius, prognozuojant, priimant strateginius sprendimus ir t.t.

Užklausimų režime informaciją galima gauti dviem būdais:

- 1) išvardijant rodiklius, kurių reikšmės domina užklausėją;
- 2) nurodant **tipinių probleminių situacijų** ar jas atitinkančių **informacijos „paketų“** pavadinimus (kodus).

Pirmasis būdas yra paprastesnis projektavimo ir diegimo atžvilgiu, nes, norint tą užklausimų pateikimo ir patenkinimo būdą įdiegti, pakanka tik išsiaiškinti, kokių rodiklių reikšmių kada nors galėtų prireikti ir sudaryti atitinkamą informacinį banką bei programinę to banko palaikymo ir informacinės paieškos įrangą. Tačiau tokia užklausimų režimo forma nėra pakankamai operatyvi ir patogi vartotojams, todėl ji naudojama tik kaip „paketinės“ formos papildymas.

„Paketinės“ užklausimų formos „paketų“ turinys ir struktūra nustatomi iš anksto, sistemos projektavimo metu, nagrinėjant tipines problemas situacijas ir sudarant sprendimų priėmimo scenarijus. „Paketinis“ užklausimų režimas palengvina ir pagreitina informacijos vartotojų dialogą su kompiuteriu, todėl **„paketinė“ užklausimų forma turėtų būti pagrindinė, o „rodiklinė“ turėtų tik papildyti ją tais atvejais, kai susiduriama su situacija, kurios nėra tipinių situacijų sąrašė, arba kai tipinėje situacijoje vartotojui prireikia tokios informacijos, kurios nėra „pakete“.**

Užklausimų režime plačiai naudojama **hierarchinėmis meniu sistemomis**, leidžiančiomis gana greitai suformuoti bet kurį tipinį užklausimą: pradžioje kompiuterio ekrane pasirodo hierarchinės meniu sistemos aukštutiniojo ešelono meniu, kuriame vartotojas pasirenka ir nurodo tą sritį, kuriai priklauso formuojamas užklausimas; pasirodžius tos srities posistemų sąrašui, išsirenkamas reikalingas posistemis ir t.t. iki tol, kol išsirenkamas konkretus rodiklių „paketas“ ar net atskiras rodiklis. Žinoma, jei vartotojas žino paketo pavadinimą, jis tą paketą gali išsikviesti iškart.

Projektuojant užklausimų sistemą, reikia sudaryti tipinių užklausimų (tipinių situacijų, kada gali kilti užklausimo reikiamybė) sąrašą, o taip pat tose tipinėse situacijose reikalingų rodiklių sąrašus bei vartotojo dialogo su kompiuteriu galimus scenarijus, sukurti atitinkamą programinę įrangą ir sudaryti užklausimų patenkinimui reikalingos informacijos banką.

Racionaliai derinant reglamentinį, signalinį ir užklausimų režimus, galima sukurti informacinio aprūpinimo sistemą, tenkinančią **informavimo efektyvumo principo** reikalavimus.*

* Šio ir kitų informacinių principų formuluotes bei aprašymus skaitytojas gali rasti šių eilučių autoriaus knygutėje „Pirmosios kartos ĮAVS neefektyvumo priežastys. Antrosios kartos ĮAVS projektavimo principai“ 151-172 puslapiuose.

Informavimo sistema laikoma efektyvia, jei subjektai gauna tik tokią informaciją, kurios panaudojimo efektas yra ne mažesnis, negu tos informacijos gavimo ir pateikimo išlaidos, ir gauna tą informaciją kaip tik tada, kai ta informacija subjektui tampa (arba galėtų tapti) reikalinga. Dauguma dabar veikiančių informacinių sistemų (tame skaičiuje ir Lietuvos valstybinė statistinės informacijos sistema) netenkina informavimo efektyvumo principo reikalavimų, nes jose egzistuojantys informaciniai srautai labai dažnai būna perkrauti silpnai panaudojama (arba net visai nepanaudojama) informacija, o be to didžiulėse lentelėse bei foliantuose ne visada patogiu ieškoti reikiamos informacijos.

Gana artimi informavimo efektyvumo principo reikalavimams yra ir **informavimo savalaikiškumo, informacijos patikimumo ir informavimo pilnumo** principų reikalavimai, o taip pat **informacijos nuoseklaus ir lygiagretaus perteklingumo mažinimo principų reikalavimai.** Visi šie principai (ir daug kitų informacinės įrangos projektavimo ir diegimo principų) yra aprašyti jau minėtoje šių eilučių autoriaus knygutėje, kurią SIS projektuotojams būtų naudinga perskaityti. Todėl čia apie juos plačiau nebekalbėsiu, o apsiribosiu tik nuoroda į tuos principus ir jų aprašymą knygutėje, tikėdamasis, jog skaitytojas turės noro ir ras laiko tą knygutę perskaityti. Tai padėtų jam išvengti kai kurių dažnai pasitaikančių klaidų ir sukurti pakankamai efektyvią informacinę sistemą.

Projektuojant informacinius srautus ir bankus, reikia nustatyti ne tik tų srautų ir bankų turinį bei struktūrą, bet ir jų **apimtis.** Tų apimčių žinojimas reikalingas, apskaičiuojant kompiuterinės atminties normatyvinį dydį ir informacinių kanalų normatyvinę praleidžiamąją galią. Informacijos apimtis galima matuoti bitais, baitais, kilobaitais ir megabaitais bei gigabaitais ir taip pat ženklų skaičiumi, o kanalų praleidžiamąją galią – baitų bei ženklų per sekundę skaičiumi. Kadangi kanalų informacinis apkrovimas gali būti nepastovus, tai reikia nustatyti ir nurodyti ne tik vidutinius, bet ir maksimalius bei minimalius apkrovimus, galimas informacijos perdavimo seansų trukmes. Nustatant perduodamos informacijos apimtį, reikia atsižvelgti ir į informacijos patikimumo užtikrinimo poreikius, dėl kurio tos apimtys gali žymiai išaugti. Taip pat svarbu išsiaiškinti informacijos gavimo bei perdavimo dažnius, atsiradimo šaltinius, judėjimo maršrutus ir greičius, vartotojų adresus, įvertinti informacijos paruošimo vartojimui, jos saugojimo bei perdavimo sąnaudas.

Informacinės įrangos ir informacinių srautų diagramų (ISD) projektavimas, kaip jau buvo minėta, vyksta lygiagrečiai su funkcinio ir objektinio organizacijos struktūros ir sudėties projektavimu, o taip pat ir su TNOD projektavimu bei tobulinimu.

Projektuojant bei tobulinant TNOD, nustatomi kiekvieno veiklos subjekto bei objekto (organizacijos, jos padalinio, organizacijos darbuotojo, kompiuterio bei jo informacinės ir programinės įrangos, kiekvieno įrenginio, veikiančio automatiniaiame ar automatizuotame režime) **informaciniai poreikiai bei informacinė produkcija** – veiklos subjektų ir objektų **informaciniai įėjimai ir išėjimai**, t.y. nustatoma, **kas, kokią** informaciją, **kokiame režime, kada ir iš kur** turi teisę ar privalo **gauti**, o taip pat **kas, kokią** informaciją, **kokiame režime, kada ir kam perduoda.**

Nustatomi ir informacijos perdavimo kanalai, nurodant jų tipą (pavyzdžiui, kurjerinio ar valstybinio pašto dokumentiniai kanalai, vietinio ar oficialaus ryšių tinklo elektriniai bei radijo kanalai, kompiuterinių tinklų kanalai ir pan.), projektuojamas informaciniais kanalais perduodamų pranešimų turinys ir jų formos (pavyzdžiui, dokumentų formos, displėjaus ekranų langai ir pan.).

Kartu su tuo projektuojami **informacijos surinkimo, perdavimo, apdorojimo, saugojimo, atvaizdavimo ir panaudojimo procesai**, informacijos apdorojimo **algoritmai** ir juos realizuojančios **kompiuterinės programos, informaciniai bankai** (jų sudėtis ir struktūra, jų sudarymui, palaikymui ir naudojimui reikalinga programinė įranga).

Vykstanti kartu su TNOD projektavimu bei tobulinimu sistemos struktūros ir sudėties funkcinio ir objektinio projektavimo rezultatų sintezė padeda sujungti lokalių informacinių srautų ir atskirų informacijos apdorojimo procesų projektus į bendrą visos organizacijos informacinės įrangos projektą, o taip pat patikslinti ir papildyti

lokalių informacinių srautų projektus – bendros ISD fragmentus.

Trumpai apžvelgsime kai kuriuos informacinius procesus, apsistodami ties tais jų ypatumais, kurie yra svarbūs SIS projektuotojams.

Visų informacinių sistemų (o statistinių – ypač) efektyvumas labai priklauso nuo informacijos surinkimo procesų organizavimo, todėl, projektuojant SIS, būtina suprojektuoti racionalius informacijos surinkimo ir perdavimo į jos apdorojimo, panaudojimo bei saugojimo centrus procesus. Tai padaryti yra būtina, nes dabar Statistikos departamente egzistuojančios informacijos surinkimo ir perdavimo technologijos nėra pakankamai tobulos.

Renkant statistinę informaciją, galima naudotis ištisiniais ir pasirinktiniais tyrimais.

Ištisiniu tyrimu arba **surašymu** vadinamas toks kokios nors tiriamos **populiacijos** (ansamblio, bendruomenės, visumos, aibės) tyrimas, kada duomenys renkami iš kiekvieno tos populiacijos individo (aibės elemento). Ištisinio tyrimo pavyzdys – visuotinis gyventojų surašymas. Kai tiriamąją populiaciją sudaro tokie objektai kaip prekės, inventorių ar kiti negyvi objektai, tai jų surašymą vadina **inventorizacija**. Tai irgi ištisinio tyrimo pavyzdys.

Ištisiniai tyrimai, kai populiacijos yra didelės, gana brangiai kainuoja, todėl jė didelėse populiacijose taikomi retai ir tik esant rimtoms priežastims.

Tiriant dideles populiacijas, dažniausiai taikomi **pasirinktiniai tyrimai**, kuomet duomenys renkami tik iš dalies populiacijos narių. Ta populiacijos dalis vadinama **imtimi**, o imties apimties (jos narių skaičiaus) santykis su visos populiacijos apimtimi vadinamas **imties koeficientu**. Pasirinktiniai tyrimai gali būti daromi ir tiriant nedideles populiacijas, kai jų ištisiniai tyrimai dėl tų ar kitų priežasčių yra neįmanomi ar nepageidautini.

Parenkant populiacijos elementus į imtį, atsiranda tam tikras atsitiktinumas – populiacijos elemento patekimo į imtį atsitiktinumas. Kai imtis sudaroma taip, kad elemento patekimo į imtį tikimybė tampa proporcinga (ar bent jau apytikriai proporcinga) tokio elemento atsitiktiniam paėmimui iš populiacijos, tai tokią imtį vadina **tikimybine**. Pavyzdžiui, **paprastoji atsitiktinė imtis** iš homogeninės tiriamojo požymio atžvilgiu populiacijos, kai elementai iš jos imami į imtį visiškai atsitiktinai, priklauso tikimybinių imčių klasei. Tikimybinių imčių klasei priklauso ir **sluoksninė atsitiktinė, lizdinė atsitiktinė, sisteminė atsitiktinė imtis**, apie kurias kalbėsime šiek tiek vėliau.

Tačiau kartais imtys sudaromos taip, kad populiacijos elemento patekimo į imtį atsitiktinumas tampa subjektyvus, t.y. priklausančias nuo imties sudarytojo „skonio“ ir valios. Taip sudarytos imtys vadinamos **subjektyviomis (netikimybinėmis)**. Netikimybinių imčių pavyzdžiai: 1) **ekspertinės imtys**, gaunamos kai populiacijos elementai parenkami į imtį, vadovaujantis ekspertų nuomone; 2) **proginės imtys**, gaunamos, kai į imtį ištraukiami pirmi pasitaikę populiacijos elementai; 3) **kvotinės imtys** gaunamos, kai atsižvelgiant į heterogeninės populiacijos sandarą, iš anksto nustatomos kvotos, po kiek elementų iš kiekvieno populiacijos sluoksnio reikia paimti į imtį, kad imtyje sluoksnių atstovų skaičiai būtų bent apytikriai proporcingi sluoksnių dydžiams, o po to organizuojamos ekspertinės imtys iš atskirų populiacijos sluoksnių, vadovaujantis nustatytomis kvotomis.

Reiktų pažymėti, kad kartais netikimybinės imtys gali tapti tikimybinėmis. Pavyzdžiui, ekspertinė imtis iš homogeninės populiacijos gali tapti paprastąja atsitiktine imtimi, o kvotinė imtis – sluoksnine atsitiktine imtimi. Taip atsitiks tada, kai ekspertų nuomonės bus moksliskai pagrįstos ir ekspertai sugebės sudaryti tikimybines imtis.

Ir tikimybinės ir netikimybinės imtys gali būti grąžintinės ir negrąžintinės.

Grąžintine imtimi vadinama tokia imtis, kurioje populiacijos elementas, paimtas į imtį, po jo ištyrimo (pvz., parametrų išmatavimo) tuoj pat grąžinamas į populiaciją dėl ko atsiranda tikimybė paimti jį į imtį dar kartą ir ta tikimybė išlieka ta pati, kaip ir pirmąjį kartą.

Grąžintinės imtys taikomos tais atvejais, kai populiacijos elementų savybės imties ėmimo metu ir jų tyrimo metu nesikeičia.

Negrąžintine imtimi vadinama imtis, kurioje populiacijos elementas, patekęs į imtį, atgal į populiaciją nebegrąžinamas, dėl ko keičiasi sekančio elemento patekimo į imtį tikimybė.

Negražintinės imtys taikomos tais atvejais, kai elemento, patekimo į imtį, savybės jo tyrimo metu keičiasi (kartais tyrimo metu elementas gali būti net visiškai sugadinamas).

Elementų patekimo į negražintinę imtį tikimybių apskaičiavimas tampa sudėtingesnis negu gražintinės imties atveju.

Kai populiacija pakankamai didelė, gražintinės ir negražintinės imties tikimybių charakteristikų apskaičiavimo rezultatai praktiškai nesiskiria, nes kelių ar keliolikos elementų paėmimas į imtį praktiškai nekeičia elementų ir jų charakteristikų pasiskirstymo populiacijoje.

Tokiais atvejais gana dažnai imamos negražintinės imtys, o elementų patekimo į imtį tikimybės skaičiuojamos, naudojantis paprastesnėmis formulėmis, taikomomis gražintinėms imtims.

Pagrindinis pasirinktinių tyrimų trūkumas – jie neleidžia nustatyti tikslų **populiacijos parametru** reikšmių, nes, tirdami imtį vietoj visos populiacijos, gauname tik **statistinius populiacijos parametru įverčius**, vadinamus **statistikomis**. Kitaip tariant, tiriant imtį vietoj visos populiacijos, neišvengiamai padaroma **imties paklaida**. Imties paklaida vadinamas skirtumas tarp tikslios parametro reikšmės ir jos statistinio įverčio. Nustatyti tikslų imties paklaidą galima tik įvykdžius abu tyrimus – pasirinktinį ir ištisinį. O tai kartais daroma, tiriant pasirinktinių metodų tikslumą. Tačiau net tuomet mes negalime garantuoti, kad imties paklaida nustatyta visiškai tiksliai, nes atliekant ištisinius tyrimus irgi galimi apsirikimai. Todėl, bendrai paėmus, imties paklaida yra atsitiktinis dydis ir gali būti vertinama statistiškai, tai yra, nurodant tos paklaidos **tikimybinį pasiskirstymą** (skirstinį).

Imties paklaidą gali sudaryti dvi sudėtinės jos dalys: atsitiktinė ir sistemingoji paklaidos.

Atsitiktinė imties paklaida atsiranda todėl, kad imties tyrimas negali suteikti visiškai tikslios informacijos apie visą populiaciją. Atsitiktinės paklaidos išvengti neįmanoma, ją galima tik sumažinti, tinkamai nustačius imties dydį ir tinkamai parinkus populiacijos elementus į imtį. Nors, bendrai paėmus, atsitiktinė paklaida mažėja, didėjant imties apimčiai, tačiau šis mažėjimas yra lėtėjantis ir, pasiekus tam tikrą imties dydį, tampa praktiškai nebereikšmingas. Todėl iš atsitiktinės paklaidos mažėjimo, didinant imties apimtį, dar neišplaukia, kad visuomet būtina stengtis imti dideles imtis. Vienas iš šiuolaikinių viešosios nuomonės apklausų metodikų kūrėjų, Dž. Gelapas (George Gallup), tai aiškina maždaug taip: „Ar norite atlikti apklausą Niujorko valstijoje, ar Baton Ruzo mieste (160 000 gyventojų), jums reikia apklausti tiek pat gyventojų. Čia nėra jokio stebuklo – kai verdami du puodai sriubos ir vienas iš jų yra dešimt kartų didesnis už kitą, virėjui nereikia semti dešimt kartų daugiau šaukštų iš didesniojo puodo, kad nustatytų sriubos skonį“.

Matematinės statistikos vadovėliuose pateikiami imties dydžio nustatymo metodai.

Sisteminioji imties paklaida atsiranda dėl netinkamo populiacijos elementų parinkimo į imtį, t.y., būtent dėl to, kad netinkamai parinkti imties elementai iškreiptai atstovauja populiaciją. Tuomet sakoma, jog imtis nepakankamai **reprezentatyvi**, jog egzistuoja **imties iškreiptis**. Sistemingosios paklaidos galima išvengti, taikant tinkamus **imties ėmimo metodus**. Žinoma, tai nėra paprasta, nes labai sunku atsižvelgti į visus faktorius, kurie gali turėti įtakos imties reprezentatyvumui. Didindami imties apimtį, mes galime sumažinti atsitiktinę imties paklaidą, tačiau tai nereiškia, kad tuomet būtinai sumažės ir sistemingoji imties paklaida. Kai imtis sudaroma netinkamai (iškreiptai), kartais imties apimties didinimas gali net padidinti sistemingąją paklaidą. Pailiustruosime šį teiginį istoriniu pavyzdžiu.

1936 metais, kai JAV prezidento rinkimuose Alfredas Lendonas balotiravosi prieš Frankliną D. Ruzveltą, žurnalas „Literatūrinė apžvalga“ („Literary Digest“) organizavo vieną didžiausių ir brangiausių apklausų. Remdamiesi visais JAV telefonų sąrašais, žurnalų prenumeratorių sąrašais, klubų ir asociacijų sąrašais bei kitais šaltiniais, žurnalo darbuotojai sudarė 10 milijonų adresatų sąrašą. Kiekvienam, pakliuvusiam į tą sąrašą, buvo pasiųstas netikras balsavimo biuletenis ir buvo prašoma užpildyti tą biuletenį ir atsiųsti žurnalui atgal. Į tą prašymą atsiliepė maždaug 2,4 mln. žmonių. Pagal gautus apklausos duomenis žurnalas

prognozavo, jog už Lendoną balsuos 57% rinkėjų, o už Ruzveltą – 43%. Tikrieji rinkimų rezultatai sukėlė šoką: 62% rinkėjų balsavo už Ruzveltą, o už Lendoną – tik 38%. Kaip matome, šio pasirinktinio tyrimo imties paklaida buvo 19%, ir beveik visą tą paklaidą sudarė sistemingoji paklaida. Tuo pat metu Dž.Gelapas, remdamasis daug kartų mažesne imtimi (apie 50 000 žmonių), sugebėjo numatyti Ruzvelto pergalę.

Tokią didelę „Literatūros apžvalgos“ apklausos paklaidą lėmė dvi priežastys. Visų pirma, imtis buvo sudaryta, remiantis telefonų abonentų, žurnalų prenumeratorių ir klubų narių sąrašais. Tačiau 1936 metais telefonus turėjo tik turtingiau gyvenančios šeimos, o be to Didžiosios depresijos periodo pabaigoje šalyje buvo apie 9 mln. bedarbių, ir jų bei kitų vargingiau gyvenančių žmonių pavardės vargu ar galėjo būti klubų ir žurnalų prenumeratorių sąrašuose. Tie sąrašai negalėjo pakankamai gerai reprezentuoti tuometinės JAV visuomenės. Taip atsirado **ėmimo (atrankos) iškreiptis**. Antroji priežastis – **neatsakymo iškreiptis**: iš 10 mln. apklausiamųjų atsakė tik apie 2,4 mln. **Atsakymo lygmuo** buvo tik 0,24, o laikoma, kad atsakymo lygmuo, mažesnis už 0,5, yra per mažas. Juk atsakiusiųjų nuomonės galėjo žymiai skirtis nuo mažiau aktyvių neatsakiusiųjų į anketą žmonių nuomonių.

Panašus fiasko nutiko ir 1992 m. Lietuvoje, organizuojant visuomenės apklausą ir prognozuojant rinkimų į Lietuvos Respublikos Seimą rezultatus. To fiasko priežastys buvo plačiai aptartos spaudoje, todėl jų čia nebevardinsiu.

Dž.Gelapas, tiksliau nuspėjęs 1936 m. JAV prezidento rinkimų rezultatus, naudojo taip vadinamos **kvotinės imties** metodu. Sudarant kvotinę imtį, populiacija pagal įvairius požymius, kurie laikomi svarbiais, skaidoma į grupes, nustatomi grupių dydžiai, o po to į imtį imami grupių atstovai taip, kad atstovų skaičiai būtų proporcingi grupių dydžiui. Tačiau **sudaryti gerą kvotinę imtį nėra paprasta, nes nėra paprasta nustatyti, kokios populiacijos narių charakteristikos yra svarbios ir į kokias grupes reikia skaidyti populiaciją**. Be to, kyla klausimas, kaip parinkti grupių atstovus į imtį. Juk, juos parenkant bei apklausiant, gali pasireikšti taip vadinamoji **klausėjų įtaka**. Būtent dėl kvotinės imties metodo trūkumų Dž.Gelapui, sėkmingai nuspėjusiam 1936, 1940 ir 1944 metų JAV prezidento rinkimų rezultatus, 1948 m. nepavyko nuspėti, kad rinkimus laimės H.Trumenas.

Norint išvengti aukščiau išvardintųjų imties iškreipčių, naudojamosi **sluoksnine atsitiktine imtimi**, kuri skiriasi nuo kvotinės imties tuo, kad kvotinė imtis yra **ekspertinė**, o sluoksninė – **tikimybinė**. Tai reiškia, kad naudojant kvotinę imtį individai į imtį parenkami apklausėjo (eksperto) nuožiūra, o naudojant sluoksninę atsitiktinę imtį individai į ją parenkami taip, kad jų patekimo į imtį tikimybės kiekviename sluoksnyje būtų vienodos. Naudojantis sluoksninės imties sudarymo metodu, **heterogeninė** populiacija suskirstoma į **homogeninius sluoksnius (stratas)**, o po to iš tų sluoksnių, proporcingai jų dydžiui imamos atsitiktinės imtys. **Skirstymas į sluoksnius vykdomas pagal požymius, kurie laikomi esminiais**, t.y., pagal tuos požymius, kurie gali turėti poveikį tyrimo rezultatams. Pavyzdžiui, jei tiriamoji populiacija yra žmonės, tai tą populiaciją galima stratifikuoti, taikant ir derinant geografinius ir demografinius bei socialinius kriterijus, t.y. išskiriant regionus (pavyzdžiui: Aukštaitiją, Žemaitiją, Suvalkiją ir t.t.), juose gyventojus skirstant pagal gyvenvietės dydį (pavyzdžiui: didieji miestai, vidutiniai miestai, miesteliai, kaimai, vienkiemiai) ir t.t. Sluoksniai gali būti išskiriami ir pagal lytį, amžių, tautybę, profesiją ir pan. Jei sluoksniai pasirodo per dideli arba jų būna per daug, sudaroma **daugiapakopė atsitiktinė imtis**. Ji gali būti sluoksninė (**daugiasluoksnė** hierarchinė), lizdinė ir mišrioji. **Lizdinė (klasterinė)** imtis skiriasi nuo sluoksninės tuo, kad, taikant sluoksninę imtį, heterogeninė populiacija skaidoma į tokius poaibius, vadinamus sluoksniais, kurie savo viduje yra homogeniški atitinkamų esminių požymių atžvilgiu, o taikant lizdinę imtį, heterogeninė populiacija pagal tam tikrus požymius homogeniškai skaidoma į poaibius, vadinamus **lizdais (klasteriais)**, kurie, būdami heterogeniški savo viduje esminių požymių atžvilgiu, yra tarpusavyje ir visos heterogeninės populiacijos atžvilgiu panašūs (homogeniniai). Todėl, taikant lizdinę imtį, nėra būtina imti imtis iš kiekvieno lizdo, o pakanka atsitiktinai pasirinkti tik kai kuriuos lizdus ir imti atsitiktines imtis iš jų. Taikant daugiapakopės atsitiktinės imtis, gana dažnai taikomos taip

vadinamos **mišrios daugiapakopės** imtys. Jas taikant heterogeninė populiacija pradžioje suskaidoma į sluoksnius, po to kiekvienas sluoksnius dalijamas į **lizdus (klasterius)** ir atsitiktinai iš kiekvieno sluoksnio lizdų imama atsitiktinė imtis (t.y. palyginti nedaug lizdų). Jei ir patys lizdai yra per dideli, tai jie vėl dalijami į lizdelius, vėl atsitiktinai imama atsitiktinė lizdelių imtis ir t.t. Paskutiniame etape taikoma **paprastoji atsitiktinė imtis**, t.y. tokia imtis, kuria naudojantis, paskutiniojo etapo lizdelio nariai turi vienodą tikimybę patekti į imtį. Iš to lizdelio narių sąrašo į imtį nariai parenkami atsitiktinai, pavyzdžiui, generuojant kompiuteriu atsitiktinius (ar pseudoatsitiktinius) numerius. Imant atsitiktinę imtį iš mažiausiųjų lizdelių, galima naudotis ir taip vadinama **sisteminė atsitiktinė imtimi**. Jos ėmimas organizuojamas taip: 1) lizdelio nariai kokiu nors būdu išrikiuojami ir surašomi į sąrašą, ir tokiu būdu kiekvienas lizdelio narys gauna kokį nors numerį nuo 1 iki N , kur N – lizdelio narių skaičius; 2) pagal pasirinktą **imties koeficientą** n/N , užtikrinantį, kad atsitiktinė imties paklaida bus pakankamai maža, apskaičiuojamas imties dydis n ; 3) iš intervalo $[1; N/n]$ atsitiktinai išsirenkamas sveikas skaičius (numeris), tarkime s_0 ; 4) sudaroma numerių seka: $s_0, s_0+k, s_0+2k, \dots, s_0+(n-1)k$, kurioje k yra sveikoji skaičiaus N/n dalis, vadinama sisteminės atsitiktinės imties **žingsnio dydžiu**; 5) pagal tos sekos numerius s_0+ik , kur $i=0, 1, 2, \dots, n-1$, išrenkami lizdelio nariai į atsitiktinę sistemine imtį. Atsitiktinė sisteminė imtis turi keletą teigiamų savybių: 1) ją imti gana paprasta; 2) jos ėmimo rezultatus lengva patikrinti, nes pakanka žinoti lizdelio elementų sąrašą ir pradinį numerį s_0 ; 3) jei tas sąrašas buvo sudarytas, išrikiuojant lizdelio narius kokio nors esminio požymio didėjimo arba mažėjimo tvarka, tai sisteminės imties ėmimas sluoksniuoja lizdelio populiaciją ir tokiu būdu gali pagerinti imties ėmimo rezultatą, lyginant su paprastąja atsitiktine imtimi, kurioje lizdelio kiekvieno nario, imamo į imtį, numeris parenkamas visiškai atsitiktinai. Žinoma, jei lizdelio narių sąrašas sudarytas taip, kad jame koks nors esminis požymis periodiškai kinta su periodu, kartotinu skaičiui k , tai sisteminės imties ėmimas gali duoti iškreiptą rezultatą, tačiau tokie atvejai gali pasitaikyti tik labai retai ir jų galima išvengti, jei tokią galimybę turėti omenyje. Todėl atsitiktine sisteminė imtimi paskutiniame daugiapakopės sluoksninės atsitiktinės imties ėmimo etape naudojama gana dažnai.

Daugiapakopės mišriosios atsitiktinės imties ėmimą galima pailustruoti tokiu Lietuvos gyventojų apklausos pavyzdžiu. Naudodamiesi geografiniais ir demografiniais kriterijais, Lietuvos gyventojus suskirstome į sluoksnius. Kiekvieno sluoksnio gyventojus, pavyzdžiui, šiaurės Aukštaitijos vidutinių miestų gyventojus, suskirstome į lizdus – kiekvieno miesto gyventojus. Iš tų miestų atsitiktinai išrenkame kelis. Išrinktuose miestuose atsitiktinai išsirenkame po kelis kvartalus. Šie kvartalai dalijami į apylinkes, ir kiekviename tų išsirinktų kvartalų atsitiktinai pasirenkamos kelios apylinkės. Apylinkėse gyvenančių žmonių sąrašai jau nėra labai dideli (jei jie būtų per dideli, juos būtų galima toliau dalinti), todėl iš jų atsitiktiniu būdu (pavyzdžiui, naudojantis atsitiktine sisteminė imtimi) pasirenkami konkretūs gyventojai.

Žinoma, ir daugiapakopės atsitiktinės imties atveju gali pasireikšti apklausėjo įtaka, tačiau ir pačius apklausėjus galima rinktis, naudojantis atsitiktinės imties metodais, bei po to juos dar papildomai apmokyti, kaip atlikti apklausą, nesudarant kryptingo psichologinio spaudimo apklausiamiesiems. Labai svarbu taip pat, kad apklausos anketos būtų sudarytos tinkamai ir neprovokuotų kryptingų atsakymų, t.y. kad būtų išvengta **klausimo iškreipties**, o taip pat **neatsakymų iškreipties**.

Daugiapakopė atsitiktinė imtis duoda palyginti mažą iškreiptį ir, jei imtys pakankamai didelės, – palyginti mažą atsitiktinę imties paklaidą (dažniausiai mažesnę, negu 3 %). Tuo pačiu metu tokios imties ėmimo kaina ir trukmė, palyginti, nėra labai didelės, nes imties dydis taip ir pasirenkamas, kad imties ėmimo išlaidos, užtikrinant, jog imties paklaida būtų ne didesnė už leistinąją, būtų mažiausios.

Čia išvardintos ir trumpai aprašytos tik kai kurios imčių ėmimo metodų grupės. Praktikoje vartojamos ir įvairios tų metodų modifikacijos. Pavyzdžiui, daugiapakopėje atsitiktinėje imtyje iš sluoksnių bei lizdų gali būti imamos atsitiktinės imtys taip, kad tų imčių

apimtys būtų proporcingos ne tik pačių sluoksnių (ar lizdų) dydžiams, bet ir tiriamojo dydžio kintamumui. Gali būti ir taip, kad atsitiktinės ar subjektyvios (t.y. netikimybinės, besiskiriančios nuo atsitiktinių, vadinamų dar ir tikimybinėmis, tuo kad, sudarant atsitiktines imtis, stengiamasi siekti individų patekimo į imtį tikimybių vienodumo, o sudarant subjektyviąją imtį, individai parenkami subjektyviai, pavyzdžiui, vadovaujantis ekspertų nuomone) imtys būtų imamos iš sluoksnių, lizdų bei lizdelių nebūtinai proporcingai sluoksnių lizdų bei lizdelių dydžiams, bet vadovaujantis kitais principais, pavyzdžiui, pasinaudojant jau turima informacija apie tiriamąją situaciją ir apskaičiuojant optimalius imčių bei kvotų dydžius. Taip gali būti daroma, imant taip vadinamas **kartotines imtis**, t.y. stebint tuos pačius objektus skirtingais laiko momentais, pavyzdžiui, gydomus pacientus iki gydymo ir po gydymo. Taip gaunamos ir taip vadinamos **porinės imtys**, kuriose kiekvienas pirmosios imties elementas turi į jį panašų (nuo jo priklausantį) „porininką“ antroje imtyje. Taikant kartotines imtis, neretai stengiamasi organizuoti tokių imčių ėmimo procesą, kurio metu imčių ėmėjai mokosi ir tobulina **imčių ėmimo metodiką**.

Kartais imtis tenka imti ir norint nustatyti populiacijos ar jos sluoksnio dydį. Tokiais atvejais gali būti naudojama taip vadinamu **ženklavimo** metodu. Pavyzdžiui, biologai, norėdami nustatyti kokios nors laukinės populiacijos dydį, sugauna tiriamos populiacijos gyvūnų (paukščių, žuvų, žvėrių) imtį, sudarytą, tarkim, iš n_1 gyvūnų, paženkliną tuos gyvūnus, nepadarydami jiems žalos, ir paleidžia, o po tam tikro laiko, pakankamo, kad paženklintieji gyvūnai gerai susimaišytų su kitais tos populiacijos gyvūnais, beveik tolygiai pasiskirstydami visoje populiacijoje, sugauna naują tų gyvūnų imtį, sudarytą, tarkim, iš n_2 gyvūnų, ir suskaičiuoja joje paženklintų gyvūnų skaičių k . Jei paženklintieji gyvūnai visoje populiacijoje pasiskirstę vienodai kaip ir imtyje n_2 , tai $N/n_1 = n_2/k$, iš kur gauname, kad populiacijos dydis N yra lygus $n_1 \cdot n_2 / k$.

Panašiais bei dar sudėtingesniais metodais tenka naudotis ir tada, kada renkama informacija apie reiškinius, kuriuos kai kurie žmonės suinteresuoti nuslėpti. Pavyzdžiui, ne taip jau paprasta surinkti pakankamai tikslią informaciją apie alkoholizmo ir narkomanijos paplitimą, apie šešėlinės ekonomikos charakterį bei jos mastus ir pan. Gauti tikslią tiesioginę informaciją apie tokius reiškinius dažniausiai neįmanoma, ir apie juos tenka spręsti, naudojantis netiesiogine informacija. Pavyzdžiui, pravedus statistinius tyrimus ir nustačius, kad egzistuoja gana stabilus statistinis ryšys tarp alkoholikų ir savižudžių skaičių, vėliau tuo galima pasinaudoti, netiesiogiai vertinant alkoholizmo paplitimą. Įvertinti šešėlinės ekonomikos mastus taip pat galima tik netiesiogiai, vadovaujantis dėsniu, kad niekas neatsiranda iš nieko ir negali dingti, į ką nors nepavirsdamas, ir analizuojant įvairių srautų (pavyzdžiui, pajamų ir išlaidų) balansus ir disbalansus.

Kartais, sprendžiant panašias problemas, tenka kurti hipotetinius modelius ir po to tikrinti jų tinkamumą stebimų reiškinių aiškinimui. Norint gauti atsakymus į tuos klausimus, į kuriuos žmonės nesugeba ar nenori duoti tikslių atsakymų, neretai tenka ir eksperimentuoti. **Statistinis eksperimentas** yra viena iš priemonių, plačiai naudojamų, tikrinant įvairias hipotezes.

Pavyzdžiui, norint patikrinti, ar egzistuoja reikšmingas ryšys tarp dviejų reiškinių, kurių vieną manoma esantį priežastimi, o kitą – jos pasekmę, naudojama taip vadinamu **kontroliuojamuoju eksperimentu**. Tokio eksperimento metu populiacijoje imamos dvi vienodos ir kiek galima labiau atitinkančios visą populiaciją imtys ir sudaromos dvi grupės: **bandomoji** ir **kontrolinė**. Jei pačios imtys imamos kaip atsitiktinės, tai tokių eksperimentą vadina **atsitiktiniu**. Abiem grupėms sudaromos vienodos sąlygos, išskyrus tai, kad bandomoji grupė yra veikiamą tiriamąją priežastimi, o kontrolinė – ne. Po to žiūrima, kaip pasiskirsčiusi tiriamoji pasekmė bandomojoje ir kontrolinėje grupėse, ir daroma iš to išplaukianti išvada. Norint kiekybiškai įvertinti statistinio ryšio tarp dviejų reiškinių stiprumą, apskaičiuojami **koreliacijos koeficientai** bei **koreliaciniai santykiai**. Jų apskaičiavimo metodus galima rasti bet kuriame matematinės statistikos vadovėlyje, taip pat ir kintamųjų **regresinės priklausomybės** įvertinimo metodus, todėl aš čia jų neaprašinėsiu, o apsiribosiu

tik pastaba, kad Statistikos departamento darbuotojai, diegiant SIS, turėtų būti apmokyti naudotis matematinės statistikos metodais, o SIS programinės įrangos sudėtyje turėtų būti programos, realizuojančios **tipinius matematinius statistinės informacijos apdorojimo algoritmus**.

Kad kontrolinėje ir bandomojoje grupėse visos kitos sąlygos, išskyrus tiriamąjį poveikį būtų visiškai vienodos, gana dažnai naudojamosi **kontroliuojamuoju akluoju placebo eksperimentu**. Tokio eksperimento metu kontrolinei grupei daromas „netikras poveikis“ (tiksliau, daromas tik poveikio įvaizdis, nedarantis jokio fizinio poveikio), imituojantis tą poveikį, kuris daromas bandomajai grupei. Tas netikras poveikis, imituojantis tikrąjį poveikį, vadinamas **placebu**, iš kur ir tokio kontroliuojamojo eksperimento pavadinimas – **placebo eksperimentas**. Placebo eksperimentai gana plačiai taikomi medicinoje, tiriant kokių nors vaistų ar gydymo metodų efektyvumą, kad būtų išlygintos psichologinės eksperimento sąlygos kontrolinėje ir bandomojoje grupėse, nes buvo pastebėta, jog vien žinojimas, kad esi gydomas, ir, bendrai, psichologinis nusiteikimas (tikėjimas ar netikėjimas naudojamos priemonės efektyvumu) gali turėti tą ar kitą poveikį organizmui. Eksperimentas vadinamas akluoju todėl, kad stengiamasi, jog kontrolinės ir bandomosios grupių nariai nežinotų, kuriai iš grupių jie priklauso ir kurie iš jų gauna tikrąjį poveikį, o kurie – placebo. Gali būti tiriama dar viena, atstovaujanti visą populiaciją, kontrolinė grupė, kurios nariai jokio poveikio – nei tikro, nei placebo – negauna. Bandomosios ir kontrolinės grupių nariai nebūtinai turi būti suburti į kokias tai atskiras grupes ir gali net nežinoti vienas kito. Pačios grupės gali egzistuoti tik kaip sąrašai, o tikrinamasis poveikis ir placebo gali būti vykdomi atskirai su kiekvienu tų grupių individų (arba net niekaip nevykdomi, kaip su antrosios kontrolinės grupės individais, kurie net iš viso gali nežinoti, jog priklauso kažkokiai „kontrolinei grupei“).

Siekiant, kad eksperimento rezultatai būtų dar objektyvesni, naudojamosi **kontroliuojamuoju dvigubai aklu placebo eksperimentu**, kurio metu ne tik tiriamieji, bet ir patys tyrėjai bei tyrimo rezultatų apdorotojai nežino, kurie individai priklauso bandomajai, o kurie – kontrolinei grupei, t.y. kurie individai gauna tikrą poveikį, o kurie – placebo. Ir tik pasibaigus eksperimentui ir užfiksavus jo rezultatus, paskelbiami individų, kurie gavo tikrą poveikį, kodai ir įvertinamas ryšys tarp tiriamojo poveikio ir tiriamosios pasekmės.

Dvigubai aklo kontroliuojamojo atsitiktinio placebo eksperimento pavyzdžiu galėtų būti eksperimentas, įvykdytas 1954 m., tiriant 1953 m. sukurtos vakcinos nuo poliomiellito efektyvumą. Buvo atsitiktinai pasirinkta 750 000 vaikų, kuriems buvo pasiūlyta dalyvauti vakcinos tyrime. Iš jų apie 340 000 vaikų atsisakė, o dar apie 8 500 iškrito bandymo metu. Likusieji vaikai buvo padalyti į dvi grupes – bandomąją ir kontrolinę – maždaug po 200 000 vaikų kiekvienoje grupėje. Kiekviena iš grupių buvo sudaryta atsitiktinio ėmimo metodu taip, kad būtų kuo labiau panašios. Bandomosios grupės vaikai gaudavo tikrą vakciną, o kontrolinės grupės vaikai – panašų į ją nekenksmingą fiziologinį tirpalą. Eksperimentas buvo dvigubai aklas, ir gydytojai, nustatinėdami, ar vaikas serga poliomiellitu ar ne, nežinojo, ar tas vaikas buvo gavęs vakciną, ar tik placebo. Taip buvo daroma todėl, kad gydytojų diagnozės būtų objektyvesnės. Tai buvo svarbu, kadangi poliomiellito ligą ne visada lengva tiksliai diagnozuoti. Įvykdyto eksperimento rezultatai įtikinamai parodė, kad vakcina yra efektyvi.

Čia aprašytas eksperimentas priklauso **aktyviųjų eksperimentų** klasei. Aktyviaisiais eksperimentais (o dažniausiai – tiesiog eksperimentais) vadinama veikla, kuri specialiai vykdoma tam, kad būtų gautas atsakymas į iškilusius klausimus. Vykdam aktyviuosius eksperimentus, gana dažnai sudaromos specialios sąlygos, eliminuojančios pašalinius poveikius ir užtikrinančios eksperimento rezultatų objektyvumą (pavyzdžiui, placebo, dvigubai aklas placebo eksperimentas ir pan.).

Tačiau kartais atsakymus į klausimus galima gauti ir nevykdam specialių eksperimentų, o stebint natūralioje aplinkoje vykstančius reiškinius, pavyzdžiui, nagrinėjant savo ir kitų žmonių reguliarią veiklą ir jos rezultatus. Tokie **stebėjimai** ir jų rezultatų analizė bei iš jos išplaukiančių išvadų formulavimas kartais vadinami **pasyviaisiais eksperimentais** (arba tiesiog – **stebėjimais**).

Pavyzdžiui, apie tų ar kitų gydymo metodų bei naudojamų medikamentų efektyvumą vertingos informacijos galima gauti ne tik vykdant aktyviuosius eksperimentus su gyvūnais ar žmonėmis, bet ir apdorojant statistinę informaciją, sukaupią besigydančių (ar kada nors gydytų) pacientų poliklinikinėse bei klinikinėse ligų istorijose, sanatorinėse kortelėse ir t.t. Naudojantis atitinkamais matematiniais metodais ir kompiuterine technika, būtų galima „atfiltruoti“ nemažai objektyvios informacijos, kurią būtų galima panaudoti ir moksliniams tikslams, ir konkrečių pacientų gydymui. Nesakyčiau, kad šiuo metu minėtoji informacija, sukaupia tūkstančiuose (ir net milijonuose) ligų istorijų, bent kiek efektyviai panaudojama, nors sąlygos tokiam panaudojimui jau atsirado. Įvairias apklausas irgi galima priskirti prie pasyviųjų eksperimentų, nors, žinoma, kai kurias apklausas galima praveisti ir taip, kad jos bus labiau panašios į aktyvius, o ne į pasyviuosius eksperimentus.

Informaciją, kurią renka ir kaupia valstybinės statistikos sistemos darbuotojai, taip pat galima laikyti pasyviųjų eksperimentų rezultatais ir analizuoti ne tik esamos padėties vertinimo bei momentinių sprendimų priėmimo tikslais, bet ir kaip eksperimentinę mokslinių tyrimų medžiagą. Nesakyčiau, kad statistinė informacija, sukaupia statistikos tarnybose, šiuo atžvilgiu būtų efektyviai naudojama. Norėtųsi tikėti, kad ateityje bus suvokta, kokią didelį turtą sudaro surinkta, naudojantis patikimais metodais, ir gerai struktūrizuota statistinė informacija, ir kad tas turtas nebus tik renkamas, bet ir pakankamai efektyviai naudojamas.

Statistinės informacijos rinkimo problemų, procesų ir metodų trumpi aprašymai čia pateikti tam, kad būtų atkreiptas SIS projektuotojų dėmesys į tai, jog statistinės informacijos rinkimas – ne tik didelis „juodas“, bet ir gana sudėtingas bei svarbus darbas, reikalaujantis specialių žinių ir įgūdžių, specialaus pasiruošimo ir specialaus informacijos rinkimo procesų projektavimo, specialių metodikų paruošimo. Reikia žinoti ne tik **ką rinkti**, bet ir **kaip, kur ir kada (kaip dažnai) rinkti**.*

Apie pirmąjį klausimą mes jau kalbėjome ir priėjome išvadą, kad, kuriant SIS, pakankamai pagrįstą atsakymą į šį klausimą galima gauti, tik projektuojant SIS kaip VADIS sudėtinę dalį kartu su VADIS. Tik tokiu būdu galima susidaryti pakankamai pilnus ir pakankamai pagrįstus rodiklių – indikatorių, rodiklių – valdomųjų kintamųjų ir rodiklių – reikšmingų nevaldomųjų kintamųjų sąrašus, nustatyti rodiklių – indikatorių priklausomybes nuo valdomųjų ir nevaldomųjų kintamųjų, sudaryti valstybės valdymui ir visuomenės informavimui reikalingų rodiklių reikšmių ir jų apskaičiavimui reikalingų duomenų bankus.

Rodikliais – indikatoriais vadinami tokie rodikliai, kurių žinojimas leidžia organizacijų ir jų padalinių vadovams spręsti apie organizacijos, jos padalinių ir jos aplinkos būsenas bei tų būsenų kitimo tendencijas. Rodiklių – indikatorių pavyzdžiais, tiriant žmogaus organizmo būseną, galėtų būti kūno temperatūra, kraujospūdis, kraujo bei šlapimo analizių rezultatai ir t.t. Panašios paskirties rodiklių – indikatorių sąrašai turi būti sudaromi ir organizacijoms, tame skaičiuje – ir valstybėms. Sudarant tokius sąrašus, susiduriama su dilema: iš vienos pusės tie sąrašai turėtų būti pakankamai pilni, kad atspindėtų įvairius organizacijų egzistavimo, jų veiklos ir jų vystymosi aspektus; iš kitos pusės tie sąrašai neturėtų būti per daug dideli, nes kitaip juos būtų sunku aprėpti ir juose susigaudyti. Sprendžiant šią dilemą, matyt, būtų naudinga kiekvienai organizacijai sudaryti **hierarchinį rodiklių – indikatorių sąrašą, turintį tris lygius, atitinkančius tris informavimo režimus: reglamentinį, signalinį ir užklausimų**.

Į pirmąjį lygį turėtų patekti nedidelis pačių svarbiausių ir pačių bendriausių rodiklių – indikatorių kiekis. Tų rodiklių reikšmės turėtų būti galima pateikti vartotojams visuose trijuose režimuose, tačiau pagrindinis jų pateikimo režimas turėtų būti reglamentinis. Žinoma, tais atvejais, kai dėl kokių nors priežasčių kokio nors rodiklio deskriptyvinė reikšmė staiga nukrypsta nuo jo normatyvinės reikšmės daugiau, negu buvo numatyta normatyviniame modelyje, apie tai turėtų būti pranešta signaliniame režime, nelaukiant eilinio reglamentinio pranešimo. Tačiau tokie atvejai neturėtų būti dažni. Suprantama, kad pirmojo lygio rodiklių –

* Besidomintiems šia tema SIS projektuotojams būtų naudinga susipažinti su SD darbuotojams A.Plikuso paruošta metodika „Imčių metodai ir jų taikymai“ ir toje metodikoje nurodyta papildoma literatūra.

indikatorių (kaip ir visų kitų rodiklių) deskriptyvines ir normatyvines reikšmes turėtų būti galima sužinoti ir užklauso režime, tačiau pagrindinis šių rodiklių pateikimo režimas normaliomis organizacijos funkcionavimo sąlygomis, kaip jau buvo minėta, turėtų būti vis dėlto reglamentinis.

Antrąjį rodiklių – indikatorių sąrašo lygį turėtų sudaryti rodikliai, kurių pateikimo pagrindinis režimas būtų signalinis. Tų rodiklių deskriptyvinių reikšmių nuokrypiai nuo normatyvinių reikšmių turėtų būti pakankamai dažnai apskaičiuojami ir, tiems nuokrypiams pasiekus iš anksto nustatytas leistinas ribas, apie tai turėtų būti automatiškai signalizuojama atitinkamiems iš anksto numatytiems valdymo sistemos atitinkamų ešelonų pareigūnams. Apie šio lygio rodiklių reikšmes, kaip ir apie pirmojo lygio rodiklių reikšmes, turėtų būti galima sužinoti ir užklauso režime, tačiau šio lygio rodikliai turėtų skirtis nuo pirmojo lygio rodiklių tuo, kad apie jų reikšmes neinformuojama reglamentiniame režime. Signalinis informavimo būdas yra būdingas operatyvaus valdymo sistemoms ir nėra įprastas statistinės informacijos tiekimui, nes statistinė informacija dabartiniu metu retai kada būna pakankamai operatyvi. Tačiau, kol dar nesukurta VADIS, kai kurias Vyriausybės signalinio informavimo funkcijas, nors ir nelabai operatyviai, galėtų vykdyti ir Statistikos departamentas. Vėliau, diegiant VADIS, didžioji dalis signalinio informavimo funkcijų turėtų pereiti operatyvaus administravimo padaliniais, tačiau ir tuo atveju signalizavimas apie kai kurių rodiklių kitimo tendencijas, artėjančias prie normatyvinių ribų, galėtų ir toliau likti SIS funkcija, nes rodiklių kitimo tendencijų nustatymui reikalingą informaciją būtų tikslinga rinkti, kaupti ir apdoroti ne kur kitur, o statistinės informacijos sistemoje.

Trečiąjį rodiklių – indikatorių sąrašo lygį turėtų sudaryti rodikliai, apie kurių reikšmes būtų informuojama tik užklauso režime. Tam lygiui turėtų priklausyti didžiausias rodiklių – indikatorių skaičius.

Svarbesniųjų trečiojo ir antrojo lygio ir visų pirmojo lygio rodiklių – indikatorių reikšmės turėtų būti spausdinamos reguliariai leidžiamuose (metiniuose, ketvirtiniuose ar mėnesiniuose) žinynuose bei biuleteniuose, panašiai, kaip tai daroma dabar. Toks informavimo būdas irgi galėtų būti laikomas reglamentiniu, tačiau tokiu būdu teikiama informacija būtų naudojama „rankinės užklauso“ režime.

Dabartiniu metu Statistikos departamente dar nėra specialiai sudarytų, oficialiai patvirtintų ir praktikoje aprobuotų rodiklių – indikatorių, valdomųjų kintamųjų ir svarbių nevaldomųjų kintamųjų sąrašų, o yra tik tų sąrašų metmenys. Tuose metmenyse rodikliai – indikatoriai dar nėra struktūrizuoti į ką tik aprašytus lygius, o yra skirstomi į grupes pagal tai, kokios srities kokius reiškinius jie charakterizuoja (pavyzdžiui, į pagrindinius ekonominės ir socialinės raidos rodiklius, demografinius, finansinius, užimtumo, darbo apmokėjimo, namų ūkio, kainų, nusikalstamumo, švietimo, atskirų ūkio šakų rodiklius ir pan.). Toks rodiklių grupavimas yra naudingas, nes pagreitina reikalingo rodiklio reikšmės paiešką informaciniuose masyvuose bei žinynuose, todėl ir ateityje juo reikėtų naudotis, papildant jį aukščiau aprašytu struktūrizavimu į lygius pagal informavimo būdus. Dabartinis statistinės informacijos tiekimo būdas (o ji dabar dažniausiai pateikiama reglamentiniame režime leidžiamuose leidiniuose) yra pritaikytas tik neskubiam analitiniam darbui ir nėra pakankamai efektyvus ir patogus operatyviam valstybės ir jos regionų bei ūkio šakų ir firmų operatyviam administravimui bei valdymui. Iš vienos pusės, tuose leidiniuose spausdinamos statistinės informacijos yra tiek daug, jog ne visuomet lengva pastebėti ir atsirinkti tą informaciją, į kurią pirmoje eilėje reikėtų atkreipti dėmesį. Iš kitos pusės, apie tos informacijos pakitimus pranešama nepakankamai operatyviai, o be to tuose leidiniuose neretai pasigendama kai kurių reikalingų rodiklių reikšmių.

Iki šiol dar nėra ir vieningo (bendro visam Statistikos departamentui) rodiklių reikšmių banko, ir kai kurie departamento skyriai kuria ir eksploatuoja nuosavus lokalius informacinius bankus, nors tai, turint bendrą kompiuterių tinklą, nėra racionalu. Nuosavus **lokalius specializuotus informacinius bankus** galima ir kartais net tikslinga turėti, tačiau reikalingas ir centralizuotas **bendras informacinis bankas** bei galimybė keistis informacija tarp

lokalijų bankų, sujungtų į dar platesnį ir detalesnį **decentralizuotą informacinį banką – antrąjį bendrojo banko lygį.**

Į visus aukščiau išvardintus reikalingus ir galimus statistinės sistemos patobulinimus reikėtų atsižvelgti, projektuojant ir kuriant SIS. Žodžiu, ir šioje srityje SIS projektuotojų, Koordinacinės tarybos narių ir visų kitų SD darbuotojų laukia didelis ir sudėtingas darbas, kurio nepadarius, būtų ne visai logiška imtis statistinėje sistemoje vykdomų darbų gilaus kompiuterizavimo, SIS informacinės, matematinės ir techninės įrangos kūrimo.

Dar mažiau padaryta Statistikos departamente **valdomųjų kintamųjų**, nuo kurių priklauso rodiklių – indikatorių reikšmės, **sąrašų ir jų reikšmių informacinių bankų** sudarymo srityje.

Kol kas Statistikos departamente dar neskiriama pakankamai dėmesio sukauptos ir kaupiamos informacijos gilesnei analizei, išsiaiškinimui, kas nuo ko ir kaip priklauso, kokius kintamuosius, nuo kurių priklauso rodiklių – indikatorių reikšmės, kas, kaip ir kokiose ribose galėtų ir privalėtų keisti. Kai kuriems Statistikos departamento darbuotojams kol kas dar nepakankamai aiški ir pati valdomojo kintamojo sąvoka ir, kas svarbiausia, nėra aiškaus supratimo, kad tokios informacijos iš viso reikia. Tuo labiau, kad ir užsakovai (Vyriausybė, ministerijos ir t.t.) tokios informacijos taip pat kol kas neprašo. Neteko girdėti, kad Statistikos departamentas būtų gavęs užsakymą pateikti tam ar kitam vadovui informaciją apie tai, nuo kokių valdomųjų kintamųjų ir kaip priklauso to ar kito svarbaus rodiklio – indikatoriaus reikšmė, arba užklausimą apie tai, kaip turėtų (ar galėtų) pasikeisti rodiklių – indikatorių reikšmės, priėmus nutarimą apie kai kurių valdomųjų kintamųjų (pavyzdžiui, mokesčių normatyvų, palūkanų normų, lengvatų bei subsidijų kai kuriems ūkio subjektams ir pan.) reikšmių pakeitimą. Mažai kas susimąsto, kad būtų galima ir reikėtų efektyviau panaudoti statistinės sistemos darbuotojų reguliariai renkamą ir kaupiamą gana gausią informaciją, ją giliau apdorojant ir darant naudingas ir svarbias valstybei, jos regionams, jos ūkio šakoms bei atskiriems ūkio subjektams išvadas. Stebint visa tai ir įvairiuose valstybės valdymo lygiuose priimamų ne visada vykusių ir dažnai keičiamų sprendimų kaleidoskopą, kartais kyla drastiška mintis, jog **Statistikos departamentas (o taip pat ir daugybė kitų institucijų) egzistuoja ir atlieka gana didelį darbą ne tiek todėl, kad kam nors jo darbo rezultatai būtų gyvybiškai reikalingi ar labai naudingi, kiek todėl, jog taip jau priimta, kad valstybėje turi egzistuoti valstybinė statistikos sistema, kuri privalo rinkti, kaupti, apdoroti, saugoti ir skelbti statistinę informaciją.** Manychiau, jog viena svarbių Vyriausybės pareigų ir svarbiausia Statistikos departamento vadovų ir darbuotojų pareiga – padaryti Statistikos departamento veiklą tikrai efektyvia ir tikrai labai naudinga mūsų jaunai valstybei ir jos visuomenei. Šis kilnus tikslas, mano nuomone, yra visiškai realus ir **turėtų būti pagrindiniu SIS kūrimo tikslu.** Dabartiniais dinaminiais sudėtingų tarpusavio ryšių laikais **tikslios (patikimos), pakankamai efektyvios ir pakankamai pilnos informacijos turėjimas reikiamu momentu ir mokėjimas ja efektyviai pasinaudoti, priimant svarbius sprendimus, yra didžiulis turtas, leidžiantis išvengti stambių klaidų ir nuostolių bei atskleidžiantis daug naujų naudingų galimybių.** Reikėtų tik ryžtis peržengti tradicinio mąstymo inercijos kiautą, pamatyti tas naujas galimybes, kurias galėtų bei privalėtų suteikti SIS valstybės valdyme bei visuomenės gyvenime, ir imtis tų galimybių įgyvendinimo veiksmų – kurti SIS kaip sudėtinę ir labai svarbią VADIS dalį, kaip veiksmingą valstybės valdymo ir visuomenės informavimo bei toje visuomenėje vykstančių procesų analizės ir prognozavimo instrumentą.

Priėmus tokią nuostatą, nebeliktų abejonių, ar reikia Statistikos departamente žinoti, kokie rodikliai nuo ko ir kaip priklauso, ar turi būti Statistikos departamente valdomųjų kintamųjų sąrašai, ar reikia kaupti informaciją apie tų kintamųjų reikšmes, jų galimo kitimo

ribas ir apie faktišką kitimą, ar reikia tokią informaciją įvairiuose režimuose (reglamentiniame, signaliniame, užklauso) tiekti valstybės ir jos regionų valdymo bei jos ūkio šakų reguliavimo ir kontrolės institucijoms, visuomeninėms masinio informavimo organizacijoms.

Tuomet taip pat paaiškėtų, jog būtina turėti ir tų **nevaldomųjų kintamųjų, nuo kurių reikšmingai priklauso svarbių rodiklių – indikatorių reikšmės, sąrašus**, kad reikia rinkti, kaupti ir analizuoti informaciją apie tokių nevaldomųjų kintamųjų reikšmes, nagrinėti jų poveikį rodikliams – indikatoriums bei **prognozuoti nevaldomųjų kintamųjų kitimo tendencijas ir jų pasekmes**. Toks darbas Statistikos departamente dabar nėra reguliariai daromas, ir jis turėtų būti įtrauktas į SIS kūrimo planą.

Čia išsakyta mintis, jog SIS turi būti kuriama kaip VADIS sudėtinė dalis, nereiškia, kad SIS projektuotojai turi pasyviai laukti, kol prasidės intensyvūs VADIS kūrimo darbai. SIS projektuotojai ir SD vadovai turėtų tapti aktyviais tų darbų iniciatoriais, **pradėdami ir vykdydami juos SIS kūrimo procese bei palaipsniui įtraukdami į VADIS kūrimo darbą kitas įstaigas bei organizacijas**. Kaip sakoma: „jei kalnas neina prie Mahometo, tai Mahometas eina prie kalno“. Nuo to išloštų visi – ir VADIS, ir SIS, ir Vyriausybė, ir visuomenė.

Iš to, kas buvo pasakyta apie rinktinės statistinės informacijos turinio nustatymą, išplaukia išvada, kad **pakankamai pagrįstus atsakymus į klausimus, kokią informaciją reikia rinkti ir saugoti, o taip pat – kaip ją apdoroti, galima gauti, tik nustačius, kam ta informacija ir kada bus reikalinga ir kaip bus panaudojama, t.y. tik analizuojant bei projektuojant organizacijų ir jų subjektų, naudosiančių statistinę informaciją, veiklą, tik projektuojant organizacijų SNOD, TNOD ir PNOD**. O tai reiškia, kad pakankamai tobulą SIS projektą galima sukurti, tik projektuojant racionalią VADIS struktūrą ir pakankamai racionalius VADIS funkcijų vykdymo procesus, pakankamai racionalią VADIS organizacinę, informacinę, matematinę ir techninę įrangą.

Tokiame projektavime tikslinga naudotis tokiu sisteminės analizės ir sudėtingose organizacijose vykdomų procesų technologijos projektavimo bei optimizavimo instrumentu kaip operacijų tyrimas.

Operacijų tyrimas – tai taikomasis mokslas ir organizacijose vykstančių procesų technologijos bei jų valdymo analizės ir tobulinimo praktika, užsiimanti optimalių ar pakankamai artimų jiems sprendimų paieška, formavimu ir realizavimu organizacijų veikloje. Operacijų tyrimas plačiai naudojasi sisteminės analizės ir sisteminio modeliavimo metodais ir todėl yra sudėtinė sisteminio sąjūdžio dalis, aktyviai prisidėjusi prie tokio judėjimo atsiradimo ir aktyviai stimuliuojanti sisteminės metodologijos ir jos taikymų kokybinį ir kiekybinį vystymąsi.*

Būdamas sudėtingose organizacijose vykstančių ir vykdomų procesų bei jų valdymo technologijų projektavimo ir optimizavimo instrumentu, operacijų tyrimas savo eile pats jau kaip savo instrumentais intensyviai ir plačiai naudojasi matematiniu modeliavimu, ypač – optimizaciniais modeliais (žiūr. 13 psl.) ir matematiniais optimizavimo metodais (tiesiniu ir netiesiniu, dinaminio ir stochastiniu programavimu, variaciniu skaičiavimu ir t.t.). Dėl šių priežasčių, o taip pat dėl to, kad operacijų tyrimo specialistai stengiasi sprendžiamas problemas formuluoti griežta matematine kalba ir spręsti jas, naudodamiesi matematikos aparatu ir kompiuteriais, operacijų tyrimą galima būtų priskirti prie tikslųjų mokslų. Tačiau operacijų tyrimo negalima suvesti vien tik į juos, nes operacijų tyrimas – ne vien tik matematikos taikymas.

* žiūr. šios metodinės priemonės 40-41 puslį.

Operacijų tyrimo specialistams turi būti būdingas ne tik sisteminis mąstymas ir mokėjimas naudotis sisteminės analizės metodais, ne tik geras matematinio aparato žinojimas ir sugebėjimas kūrybiškai juo naudotis, bet ir pakankamas tiriamos srities žinojimas, jos specifikos suvokimas, gebėjimas bendrauti su organizacijos darbuotojais ir įtraukti juos į operacijų tyrimo kompleksinių kolektyvų veiklą, tą veiklą efektyviai organizuoti.

Operacijų tyrimo metodai modeliai ir optimalių sprendinių paieškos algoritmai taikomi įvairaus tipo organizacijų įvairios veiklos srityse: racionalizuojant ir optimizuojant ribotų resursų paskirstymą; nustatant optimalius atsargų lygius ir pasirenkant optimalias atsargų papildymo bei jų optimalaus lygio palaikymo strategijas; nustatant optimalius įrengimų keitimo, jų remonto ir profilaktinės kontrolės terminus; racionalizuojant ir optimizuojant masinio aptarnavimo sistemų struktūrą ir funkcionavimą, darbų vykdymo jose tvarkaraščius; projektuojant ir optimizuojant sudėtingus įvairių ryšių (transporto, komunalinius, informacinių ryšių ir pan.) tinklus ir pasirenkant optimalius maršrutus juose; derinant darbų, vykdomų skirtingų nepriklausomų vykdytojų, sudėtinguose kompleksuose vykdymo terminus; pasirenkant optimalias ar bent jau racionalias elgesio strategijas rungtyniavimo ir kooperavimosi procesuose; optimizuojant ar bent jau racionalizuojant kontrolės ir paieškos procesus ir t.t.*

Operacijų tyrimo metodus ir modelius būtų tikslinga panaudoti, projektuojant ir diegiant SIS: projektuojant ir tobulinant SNOD ir TNOD, informacijos rinkimo, perdavimo, kaupimo, apdorojimo, saugojimo ir panaudojimo procesus, informacinių srautų ir informacinių bankų turinį ir struktūrą, reikalingos informacijos paiešką informaciniuose bankuose, dialogų „žmogus – kompiuteris“ scenarijus, SIS techninių priemonių (pavyzdžiui, ADV ir kitokių terminalų) tinklus, jų remonto ir profilaktinės kontrolės procesus bei tvarkaraščius ir t.t.

Operacijų tyrimo metodų taikymas, projektuojant TNOD ir SIS informacinę įrangą, padeda gauti pakankamai pagrįstus atsakymus ne tik į 107 puslapyje minėtąją pirmąją, bet ir į antrąją grupę klausimų, kokią informaciją, kur, kada ir kaip rinkti, kaip ir kur ją kaupti, apdoroti ir saugoti, kada, kur ir kaip panaudoti.

Todėl būtų labai pageidautina, kad SIS projektuotojų kolektyvo sudėtyje būtų nors du – trys operacijų tyrimo specialistai, sugebantys kurti optimizacinius modelius ir spręsti operacijų tyrimo uždavinius, o visi kiti projektuotojai būtų susipažinę su operacijų tyrimo metodais ir mokėtų jais naudotis. Būtų taip pat tikslinga, kad SIS projektuotojai, ypatingai tie, kurie užsiims SIS organizacinės ir informacinės įrangos projektavimu bei diegimu ir todėl turės nuolat bendrauti su statistinės sistemos ir įvairių kitų įstaigų bei organizacijų darbuotojais, būtų supažindinti su šiuolaikinėmis **biheviorizmo (mokslo apie žmonių bei kitų gyvūnų elgesį ir jų santykius), konfliktų išvengimo ir konfliktinių situacijų sprendimo teorijos, socialinės kognicijos (pažinimo) bei socialinės psichologijos idėjomis** ir būtų perskaitę Deilo Karnegio (D.Carnegie) knygą „Kaip įsigyti draugų ir daryti įtaką žmonėms“.**

Baigiant šį skyrelį, tikslinga dar kartą pabrėžti, jog **tik projektuojant funkcijų vykdymo procesus ir tų procesų TNOD bei PNOD (maršrutines technologines kortas, operacines kortas bei operacines instrukcijas, funkcijų vykdymo planus bei tvarkaraščius), galutinai užbaigiami lygiagrečiai vykstantys funkcinio ir objektinio projektavimo bei jų sintezės loginiai etapai ir SNOD projektavimas, t.y. užbaigiamas SIS organizacinės**

* Plačiau su operacijų tyrimu galima susipažinti šių eilučių autoriaus paruoštose mokymo priemonėse.

** D.Carnegie. How to win friends and influence people

įrangos projektavimas.

Tuo pat metu išsiaiškinami organizacijos, jos subjektų ir jos klientų informaciniai poreikiai, reikalingų informacinių srautų turinys, maršrutai, apimtys ir struktūra, informacinių pranešimų turinys ir formos, informacijos gavimo, apdorojimo ir panaudojimo algoritmai, t.y. **pradedamas SIS informacinės ir matematinės įrangos projektavimas.**

Kaip rodo SD Informatikos tarnybos pateikta priešprojektinė dokumentacija („Statistikos informacinės sistemos kūrimo koncepcija“, „Statistikos teleinformacinio tinklo projektas“, „SIS techninės dokumentacijos formavimo standartas“, „Projekto „Statistinių duomenų bazė“ kūrimo koncepcija“, „Duomenų srautų schema“, „Priemonės duomenų apsaugai ir slaptumui užtikrinti“, „Paraiška statistikos darbuotojų paruošimui darbui su personaliniu kompiuteriu“ ir kiti), dabartinis SIS projektuotojų kolektyvas ir jo vadovybė SIS informacinės, matematinės ir techninės įrangos projektavimo darbų specifiką supranta ir žino gana neblogai, todėl šioje metodinėje priemonėje, skirtoje SIS organizacinės įrangos sisteminiam projektavimui, nėra reikalo kalbėti apie specifinius informacinės, matematinės ir techninės įrangos projektavimo klausimus. Pakanka apsiriboti nuoroda į antrosios kartos AVS informacinės, matematinės ir techninės įrangos projektavimo ir diegimo principus, aprašytus jau ne kartą minėtoje šių eilučių autoriaus knygutėje. Šitų principų žinojimo pilnai pakaks tam, kad SIS Koordinacinės tarybos nariai galėtų pakankamai apibrėžtai suformuluoti tuos reikalavimus, kuriuos turėtų tenkinti SIS informacinės, matematinės ir techninės įrangos projektai, o taip pat tam, kad tų įrangų projektavimo specialistai Koordinacinės tarybos narių reikalavimus teisingai suprastų.

Daugiau kaip trisdešimties metų darbo praktika automatizuotų organizacinių sistemų kūrimo srityje įtikino šių eilučių autorių, kad **pagrindiniai tokių sistemų projektavimo ir diegimo sunkumai bei pagrindinės klaidos kyla ne informacinės, matematinės ir techninės įrangos projektavimo metu, o kuriant ir diegiant sistemos organizacinę įrangą, organizuojant organizacijos kolektyvo darbą naujomis sąlygomis.**

Turint gerai suprojektuotas ir veikiančias SNOD, TNOD ir PNOD, informacinės, matematinės ir techninės įrangos projektavimas tampa beveik grynai inžineriniu darbu, reikalaujančiu gana daug specialių žinių ir įgūdžių, tačiau neiššaukiančiu būtinybės spręsti tokias problemas, kurių sprendimas ir jo kokybė reikalauja iš projektuotojų labai gilaus organizacijos specifikos žinojimo ir mažai priklausytų nuo projektuotojų pastangų ir jų profesinių sugebėjimų šioje srityje.

Teisingai suprojektuota automatizuota organizacinė sistema (visa sistema, o ne tik kompiuterinė jos dalis) gana greitai pradeda veikti ir veikia būtent taip (ar beveik taip), kaip numatyta projekte.

Ir atvirkščiai: kompiuterizuotos organizacinės sistemos informacinės, matematinės ir techninės įrangos projektavimo ir diegimo procesai, nesiremiantys gerai suprojektuota organizacine įranga, neduoda to efekto, kurį turėtų duoti, o neretai gana didelė dalis to, kas buvo numatyta projektuose, taip ir lieka popieriuje, netapdama organizacijos realios veiklos dalimi, nes, diegiant naujas technologijas, nepakanka instaliuoti technines priemones bei aprūpinti jas atitinkama programine įranga, bet reikia išspręsti ir daug organizacinių klausimų, suprojektuoti ir įdiegti naujas „rankines“ procedūras, apmokyti organizacijos darbuotojus dirbti naujomis sąlygomis, padėti jiems įveikti psichologinį barjerą, kuris neišvengiamai atsiranda, kai pareikalaujama keisti įprastus jau gerai žinomus savo darbo metodus į kažkieno sugalvotus naujus neįprastus ir sunkiai suprantamus metodus. Visa tai pasiekti pavyksta ne iškart, jei iš viso pavyksta.

Nepasiekus planuotų rezultatų, užsakovai lieka nepatenkinti projektuotojais,

nepateisinusiais didelės dalies vilčių, o projektuotojai – savimi ir užsakovais, neįgyvendinusiais projektų, į kuriuos buvo įdėta daug kvalifikuoto darbo ir net išradingumo.

Taip, pavyzdžiui, atsitiko netolimoje praeityje, kai daugelyje organizacijų buvo projektuojamos ir diegiamos AVS. Didelėje dalyje organizacijų AVS kūrimas buvo pradėtas „ne nuo to galo“: ne nuo organizacinės įrangos kūrimo ir, remiantis ja, informacinės bei matematinės įrangos projektavimo, o nuo brangios techninės įrangos įsigijimo. Viso to rezultatas – neracionaliai iššvaistytos gana didelės lėšos, užsakovų ir projektuotojų nusivylimas ir progresyvios idėjos diskreditavimas.

Tik nedidelėje dalyje organizacijų, kuriose AVS kūrime dalyvavo ne tik kompiuterinės technikos ir programavimo specialistai, bet ir visas organizacijos kolektyvas su savo vadovais, tapusiais pagrindiniais automatizuotos sistemos kūrimo ideologais ir „generaliniais konstruktoriais“, pavyko sukurti pakankamai efektyviai veikiančias AVS.

Remdamasis savo patirtimi ir siekdamas, kad SIS projektuotojai ir Statistikos departamento vadovai nekartotų kadaise AVS projektuotojų padarytų klaidų, šių eilučių autorius norėtų dar kartą atkreipti Statistikos departamento vadovų ir SIS projektuotojų dėmesį į SNOD, TNOD ir PNOD projektavimo ir diegimo, kuriant SIS, tikslingumą, o taip pat į statistinės sistemos darbuotojų ir jos vadovų dalyvavimo SIS kūrime būtinumą.

Tik projektuojant ir racionalizuojant organizacijos veiklą bei ją reglamentuojančias SNOD, TNOD ir PNOD vadovaujantis šioje metodinėje priemonėje aprašyta sisteminė metodologija ir ką tik minėtais AVS projektavimo principais, galima pakankamai tiksliai išsiaiškinti organizacijos bei jos padalinių informacinius poreikius bei konkrečius reikalavimus, kuriuos turi tenkinti kompiuterizuotos sistemos informacinė, matematinė ir techninė įranga. Tik diegiant SNOD, TNOD ir PNOD suprojektuotas atsižvelgiant į kompiuterizuotos sistemos funkcionavimo specifiką, gali būti sudarytos sąlygos efektyviam informacinės, matematinės ir techninės įrangos, o taip pat ir visos kompiuterizuotos sistemos funkcionavimui. Net ir geriausiai suprojektuota informacinė, matematinė ir techninė įranga negalės efektyviai funkcionuoti, jei sudėtingos organizacinės sistemos „žmogiškoji dalis“ nebus pakankamai gerai organizuota bei apmokyta ir neveiks suderintai kaip sudėtingos sistemos dalis.

Todėl SIS projektavimą derėtų pradėti nuo visos statistinės sistemos veiklos valstybės ir jos valdymo sistemos sudėtyje sisteminės analizės, nuo tos veiklos racionalizavimo, naudojantis operacijų tyrimo metodais, nuo tos veiklos derinimo ir inžinerinio projektavimo, naudojantis šioje metodinėje priemonėje aprašytais ir CASE priemonėmis dalinai automatizuotais funkcinio ir objektinio projektavimo bei jų sintezės metodais, nuo tos veiklos rutininės dalies reglamentavimo bei reglamentų dokumentavimo, projektuojant ir diegiant TNOD, SNOD ir PNOD, numatant, ką ir kaip toje veikloje derėtų kompiuterizuoti.

Tik išsiaiškinus vykdytinų funkcijų sąrašus ir tų funkcijų vykdymo algoritmus, tik nustačius visos sistemos, jos padalinių ir jos darbuotojų bei jos klientų informacinius poreikius, galima pradėti jau ne tik fragmentinį, bet ir sisteminį SIS informacinės, matematinės ir techninės įrangos projektavimą. Jo metu statistinės sistemos vadovai ir darbuotojai jau galėtų apsiriboti tik užsakovo vaidmeniu, o pagrindines SIS informacinės, matematinės ir techninės įrangos projektavimo funkcijas vykdytų specialiai paskirta SIS projektuotojų grupė, sudaryta iš kompiuterinės technikos ir jos taikymų specialistų, turinčių patirtį informacinių bankų ir jų funkcionavimui reikalingos programinės įrangos kūrime bei techninės įrangos diegime.

Iki tol, tobulinant statistinės sistemos struktūrą ir veiklą bei kuriant SIS organizacinę

įrangą, reikalingas bendras viso statistinės sistemos kolektyvo darbas, **kuriame vadovaujanti vaidmenį turėtų atlikti statistinės sistemos vadovai ir juos konsultuojantys sisteminės analizės ir sisteminio projektavimo specialistai ir dženeralistai**, susibūrę į SIS kūrimo darbų koordinavimo grupę – **SIS kūrimo Koordinacinę tarybą (KT).***

SIS kūrimo Koordinacinė taryba bent jau pradiniuose SIS kūrimo etapuose, kol vyksta statistinės sistemos struktūros ir veiklos sisteminė analizė ir kol projektuojama SIS organizacinė įranga, turėtų planuoti ir svarstyti bei koreguoti SIS kūrimo darbų eigą ne rečiau vieno karto į mėnesį, priimdama strateginius sprendimus bei organizuodama jų vykdymą.

Stebint dabartinį Koordinacinės tarybos darbą Statistikos departamente, sunku apsiginti nuo nuogastavimų, kad departamento vadovybė pasuko ar bent jau gali pasukti klaidingu keliu, tapusiu beveik tipiniu daugeliui stambių organizacijų.

Tokių organizacijų vadovai, būdami labai užimti kitais darbais ir nenorėdami per daug kvaršinti sau galvos neįprastomis sisteminio projektavimo problemomis, gana dažnai pasistengia kuo greičiau nusiimti šią našą nuo savęs ir perkrauna ją ant specialiai išskirtos projektuotojų grupės pečių. Ši, neturėdama organizacijos vadovų teisių bei patirties ir būdama nepajėgi ką nors radikaliai keisti organizacijos struktūroje ir veikloje, **užsiima tik tuo, ką ji sugeba daryti – grynai kompiuterinės sistemos projektavimu ir diegimu.**

Tačiau toks automatizuotų sistemų organizacijose kūrimo būdas, kaip parodė Japonijos specialistų tyrimai, atlikti JAV, niekur nedavė teigiamo efekto. Tai patvirtino ir AVS kūrimo buvusioje Sovietų Sąjungoje patirtis. Akademikas V.M.Gluškovas tokį AVS kūrimo būdą buvo gana taikliai pavadinęs „aviacinio motoro montavimu į senas vežėčias, tikintis, jog tai leis joms išvystyti lėktuvo greitį“.

Matyt, kad kažko panašaus tikimasi ir Statistikos departamente, diegiant kompiuterinę informacinę sistemą tik nedidelės grupelės projektuotojų iš Informatikos tarnybos jėgomis, beveik nieko nekeičiant statistinės sistemos struktūroje ir veikloje.

Taip elgiantis gali atsitikti, kad SIS taps tik madingu brangiai kainuojančiu statistinės sistemos papuošalu, gal būt, ir maskuojančiu kai kurias tos sistemos negalias, kenkiančias jos imidžui, tačiau nei pridedančiu jai naujų jėgų, nei gydančiu nuo senų ligų.

Vienas pagrindinių šios metodinės priemonės parašymo tikslų kaip tik ir yra tikslas padėti Statistikos departamento vadovams ir SIS projektuotojams išvengti „aviacinio motoro montavimo į senas vežėčias“ klaidos, atkreipiant jų dėmesį į SIS organizacinės įrangos projektavimo viso statistinės sistemos kolektyvo jėgomis būtinumą ir supažindinant su to projektavimo specifika, jo loginiais etapais bei metodais.

Nors į SNOD, TNOD ir PNOD projektavimą kai kurie Statistikos departamento vadovai kol kas žiūri gana skeptiškai ir net priešiška, tačiau šių eilučių autorius, remdamasis savo patirtimi ir žinodamas, jog Statistikos departamento vadovai daugumoje atvejų sugeba pakankamai sistemiškai ir savikritiškai mąstyti, yra tvirtai įsitikinęs, kad jie gana greit pakeis savo nepakankamai apgalvotą požiūrį į normatyvinės organizacinės dokumentacijos projektavimą ir jo vietą SIS kūrimo procese.

Todėl prie šios metodinės priemonės, kaip labai svarbūs jos priedai, pridedami metodiniai nurodymai, kaip tokią dokumentaciją reikėtų projektuoti.**

Su tais metodiniais nurodymais turėtų susipažinti Koordinacinės tarybos nariai ir SIS projektuotojai, ypač tie, kurie bendraus su statistinės sistemos darbuotojais, imdami iš jų interviu, aiškindamiesi darbuotojų vykdomas funkcijas, jų informacinius poreikius ir t.t. Su tais metodiniais nurodymais taip pat turėtų būti supažindinti visi kiti statistinės sistemos darbuotojai. Pastarieji jau dabar privalėtų turėti pareiginius pasus ir vykdytinių funkcijų TNOD bei PNOD. Bendradarbiaudami su SIS projektuotojais ir vadovaudamiesi minėtais

* žiūr. šios metodinės priemonės dvylikąjį skyrelį, kuriame apibrėžiamos KT, o taip pat PK, PDG ir SDG funkcijos.

** žiūr. pirmąjį priedą „Pareigybės paso projektavimo metodiniai nurodymai“, o taip pat trečiąjį priedą „Technologinės NOD projektavimo metodiniai nurodymai“.

metodiniais nurodymais, jie turėtų sudaryti naujas PP, SNOD, TNOD ir PNOD redakcijas, atitinkančias kuriamos SIS funkcionavimo sąlygas.

SIS projektuotojai savo ruožtu, padėdami statistinės sistemos darbuotojams kurti naujas SNOD, TNOD ir PNOD redakcijas, gautų SIS informacinės ir matematinės įrangos projektavimui reikalingą informaciją ir kartu su statistinės sistemos vadovais ir darbuotojais suprojektuotų SIS funkcinę ir objektinę struktūrą, jos organizacinę įrangą, racionalizuojančią ir reglamentuojančią statistinės sistemos veiklą.

Šio skyrelio, skirto organizacijos **veiklos procesų projektavimui**, pabaigoje reikėtų pažymėti, kad **SNOD, TNOD ir PNOD, atitinkančių SIS funkcionavimo sąlygas, diegimas turi vykti kartu su SIS techninės, matematinės ir informacinės įrangos diegimu, kartu su statistinės sistemos personalo apmokymu ta įrangą naudoti**. Diegimo metu gali paaiškėti tų ar kitų pakeitimų normatyvinėje organizacinėje dokumentacijoje ir SIS informacinėje bei matematinėje įrangoje reikalingumas, todėl galima sakyti, jog **organizacijos veiklos procesų tobulinimas ir projektavimas tęsiasi per visus chronologinius SIS kūrimo etapus**.

11. ORGPROJEKTAVIMO AUTOMATIZAVIMAS. CASE METODAS IR AUTOMATIZUOTA INFORMACINIŲ KOMPIUTERIZUOTŲ SISTEMŲ PROJEKTAVIMO SISTEMA „COMPUTER – AIDED SYSTEMS ENGINEERING“ (CASE).

Sudėtingų organizacinių kompiuterizuotų sistemų projektavimas – sudėtingas procesas, reikalaujantis iš projektuotojų gana plačių ir gana gilių žinių, sisteminio mąstymo, kūrybinių sugebėjimų, kolektyvinio darbo įgūdžių, organizuotumo, atsakingumo, kruopštumo ir didelių laiko sąnaudų. Natūralu, kad projektavimo darbe buvo pabandyta pasinaudoti kompiuterine technika, jos programine įranga, intelektualaus darbo automatizavimo metodais.

Pradžioje tie bandymai nebuvo ypač sėkmingi, tačiau atsiradus kompiuterinio laiko paskirstymo automatinėms sistemoms ir dialogo su kompiuteriu režimui, kompiuterių taikymas projektavimo darbuose tapo įprastu dalyku.

Gana greitai gimė **automatizuotų projektavimo sistemų (APS)** idėja. Techninių sistemų ir technologinių procesų projektavimo srityje tokios sistemos buvo sukurtos gana greitai ir jų taikymas davė neblogų rezultatų. Dabar tokių APS yra jau gana daug ir jų panaudojimo spektras gana platus: nuo sudėtingų radioelektroninių sistemų, kosminių laivų ir avialainerių projektavimo iki APS taikymų architektūroje ir projektuojant technologinius procesus cheminėje pramonėje bei audinių raštus ir įvairų dekorą tekstilėje, poligrafijoje ir t.t.

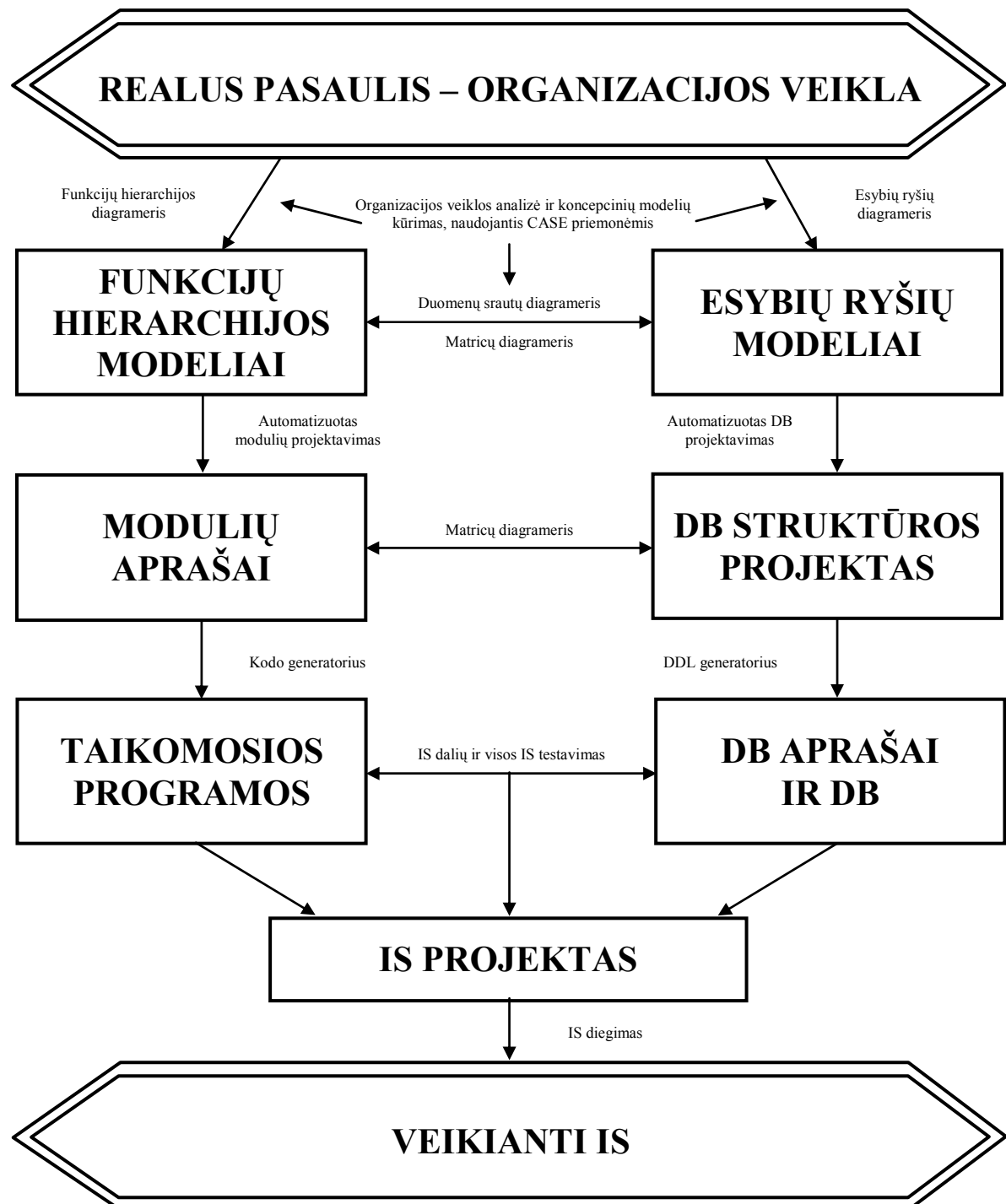
Žymiai sunkiau vyko APS kūrimas ir jų taikymas orgprojektavime, nes organizacijų projektavimo procesą buvo žymiai sunkiau struktūrizuoti ir formalizuoti.

Tačiau paskutiniaisiais XX a. dešimtmečiais **buvo sukurtos ir pradėtos naudoti APS, skirtos informacinių sistemų (IS) ir automatizuotų valdymo sistemų (AVS) organizacijose projektavimui bei to projektavimo ir diegimo procesų valdymui**. Tokiose APS kompiuterinė technika taikoma lyg ir „kvadratai“: naudojantis kompiuterine technika projektuojamos tokios sistemos, kuriose cirkuliuojančios informacijos apdorojimui vėlgi taikoma kompiuterinė technika, t.y. **automatizuojamas automatizuotų sistemų projektavimas**. Tokių sistemų pavyzdžiai: Information Engineering (IE), Structured Systems Analysis and Design Methodology (SSDM), Computer – Aided Systems Engineering (CASE).

Pastarąją APS (ir IS kūrimo metodą – CASE Method) aprašysiu detaliau, nes šia APS ir CASE metodu naudojamas projektuojant ir diegiant SIS.

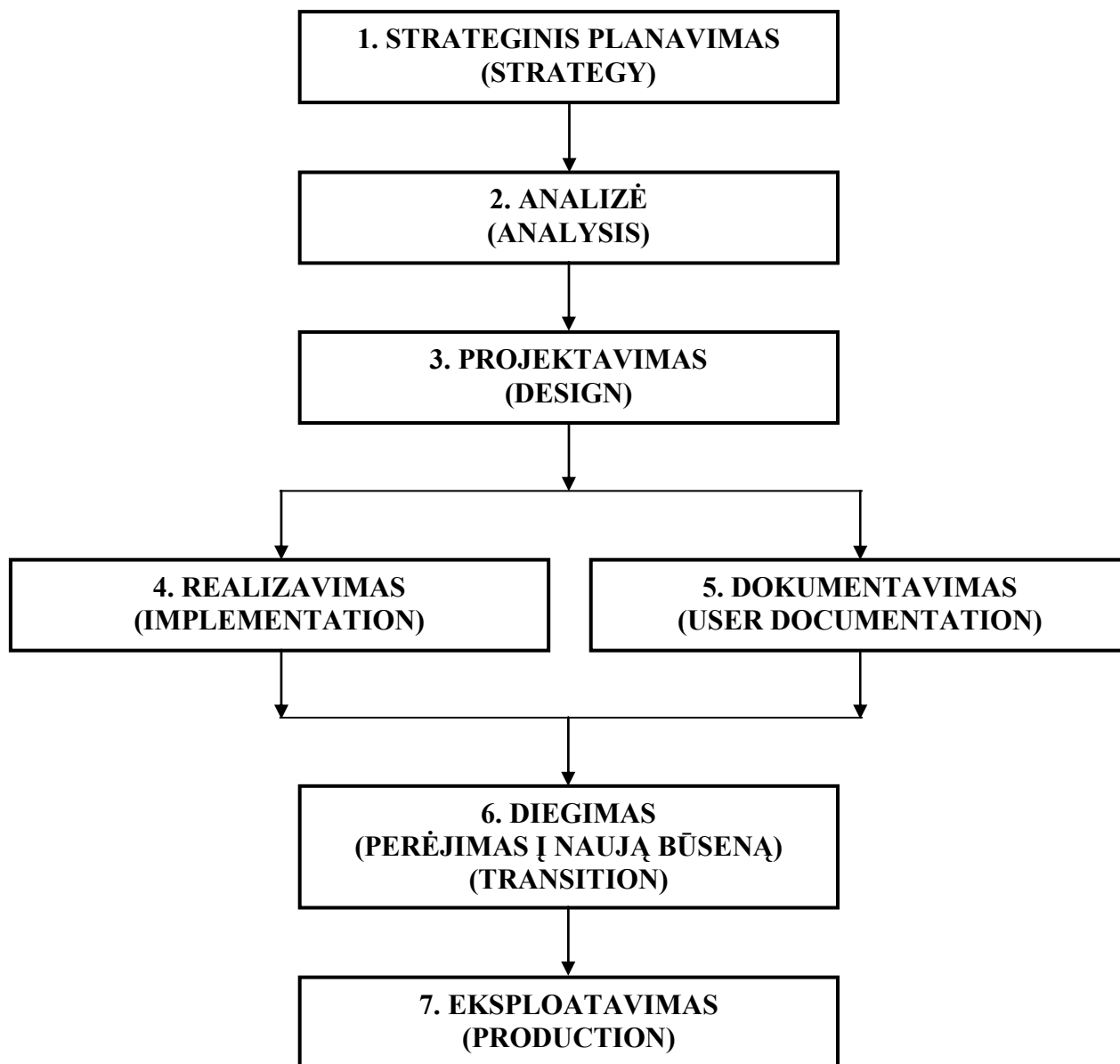
CASE metodas – tai struktūrinis IS kūrimo metodas, realizuojantis sisteminius funkcinio ir objektinio IS projektavimo ir jų sintezės metodus, naudojantis programinėmis APS CASE priemonėmis: CASE* Dictionary (CASE žodynu), CASE* Designer (CASE dizaineriu), CASE* Generator (CASE generatoriumi) ir kt. Naudojantis CASE metodu ir CASE priemonėmis pradžioje sudaromi **organizacijos veiklos**

konceptiniai modeliai, vaizduojantys veiklos funkcijų hierarchiją (**funkciniai modeliai**) ir organizacijoje egzistuojančius informacinius ryšius (**esybių ryšių objektiniai modeliai**), kurie sintezuojami į **veiklos funkcijų – esybių sąveikos matricas**. Po to tie konceptiniai modeliai, juos detalizuojant ir formalizuojant, **transformuojami į taikomųjų programų modulius ir DB aprašus**, pagal kuriuos **kuriami IS duomenų bankai ir IS programinė įranga** (žiūr. 8 pav.):



8 pav.

Taikant CASE metodą, išskiriamos **šešios chronologinės fazės**, kurias sudaro **septyni IS kūrimo etapai** (žiūr. 9 pav.):



9 pav.

Pradiniuose IS kūrimo, naudojantis CASE metodu, chronologiniuose etapuose (žiūr. 9 pav. pirmuosius du blokus) išsiaiškinami užsakovo poreikiai, analizuojama organizacijos, kurioje bus kuriama IS, veikla, **sukuriama būsimos IS bei jos architektūros vizija**, nustatoma, **ką reikia daryti**, projektuojant ir diegiant IS, **sudaromi IS projektavimo ir diegimo plano metmenys**.

Tai atitinka penktajame šios metodinės priemonės skyrelyje aprašytus **priešprojektinių tyrimų (PT) ir techninės užduoties (TU) formavimo etapus**, o taip pat **techninio projektavimo (TP) pradžią** (žiūr. šios metodinės priemonės 21 psl. 6 pav. pirmąjį, antrąjį, trečiąjį, ketvirtąjį ir penktąjį blokus bei jų aprašymą 20-jame puslapyje).

Sekančiuose CASE metodo taikymo etapuose (žiūr. 9 pav. trečiąjį, ketvirtąjį ir penktąjį blokus) projektuojama, **kaip reikėtų daryti** tai, kas buvo numatyta pradiniuose etapuose, **IS vizija transformuojama į IS projektą ir kuriamos IS sudėtinės dalys**, reikalingos IS vizijos realizavimui.

Tai atitinka šioje metodinėje priemonėje aprašyto **sisteminio orgprojektavimo (techninio ir darbinio projektų sudarymo) chronologinius etapus** (žiūr. 6 pav. penktąjį, šeštąjį, septintąjį, devintąjį, vienuoliktąjį, trylikąjį ir penkioliktąjį blokus).

Po to CASE metodas rekomenduoja įvykdyti sudarytų projektų **testavimą** ir **diegimą**,

pereinant į IS eksploataciją (žiūr. 8 ir 9 pav.).

Projektų testavimas atitinka tuos sisteminio orgprojektavimo etapus, kurių metu tikrinama techniniame ir darbiname projektuose priimtų sprendimų kokybė (žiūr. 6 pav. aštuntąjį, dešimtąjį, dvyliktąjį, keturioliktąjį, šešioliktąjį ir septynioliktąjį blokus).

IS diegimo ir eksploataavimo etapai (žiūr. 9 pav. paskutinius du blokus) atitinka sisteminio orgprojektavimo metodologijoje numatytos trečiosios sudėtingų sistemų kūrimo stadijos tų pačių pavadinimų etapus (žiūr. 6 pav. aštuonioliktąjį, devynioliktąjį ir dvidešimtąjį blokus).

CASE metodas kiekvienam IS kūrimo chronologiniam etapui nustato **standartinius** to etapo **lokaliuosius tikslus**, kurie, projektuojant konkrečią IS, gali būti konkretizuoti bei papildyti **specifiniais lokaliaisiais tikslais**. Lokalieji chronologinių ir loginių etapų tikslai parenkami taip, kad jų pasiekimas (etapas po etapo) vestų prie **sisteminių (globaliųjų) IS sukūrimo ir organizacijos veiklos efektyvumo didinimo tikslų** pasiekimo.

Siekiant užsibrėžtųjų globaliųjų tikslų, CASE metodas rekomenduoja kiekviename chronologiniame IS kūrimo etape atlikti iš lokaliųjų to etapo tikslų išplaukiančias **standartines užduotis**, kurių sąrašas, kuriant konkrečią IS, gali būti modifikuotas ir papildytas **specifinėmis užduotimis**, išplaukiančiomis iš kuriamos IS specifikos.

CASE metodas taip pat nustato **standartinius rezultatus**, kurie turi būti gauti, atliekant standartinę užduotį, ir reikalauja, kad kiekvienam IS kūrimo etapui būtų sudaromas **darbų planas**, kuriame būtų išvardinamos to etapo užduotys, jų vykdytojai, įvykdymo terminai ir užduočių įvykdymo rezultatai. Tai padeda **organizuoti ir disciplinuoti IS kūrimo procesą, priartina organizacinių sistemų projektavimą ir diegimą prie techninių sistemų projektavimo ir realizavimo procesų organizuotumo ir „inžineriškumo“ lygio**.

Viena būtinųjų kiekvieno etapo standartinių užduočių yra **ataskaitos** apie to etapo užduočių vykdymą ir pasiektus rezultatus **sudarymas** bei **atsiskaitymas** užsakovui, užsakovo **supratimo ir pritarimo pasiekimas**.

Kita būtina kiekvieno chronologinio etapo (išskyrus paskutinįjį) užduotis – **sekančio etapo darbų plano sudarymas** ir IS kūrimo **bendrojo plano patikslinimas**.

Kaip matome, taikant CASE metodą, realizuojamos **izomorfiškumo, sudėtingos sistemos kūrimo palaipsniškumo, orgprojektavimo inžineriškumo, slenkančio planavimo (slenkančio horizonto) ir iteracijų** principų rekomendacijos (žiūr. dešimtąjį, keturioliktąjį ir dvidešimtąjį sisteminės metodologijos principus šios metodinės priemonės 27, 29 ir 34 puslapiuose, o taip pat IAVS projektavimo metodologijos inžineriškumo ir slenkančių persidengiančių planavimo ir apskaitos periodų principus šių eilučių autoriaus knygutės „Pirmosios kartos IAVS neefektyvumo priežastys. Antrosios kartos IAVS projektavimo principai“ 86 ir 208÷211 puslapiuose).

Galima pasakyti dar daugiau: **CASE metodas, pasitelkdamas projektavimo ir projektavimo proceso valdymo automatizavimo priemones, realizuoja tikrai sisteminę sudėtingų informacinių sistemų kūrimo sudėtingose organizacijose metodologiją**, aprašytą šios metodinės priemonės penktajame skyrelyje ir ką tik minėtoje šių eilučių autoriaus knygutėje.

IS projektavimo chronologinių etapų užduotys, taikant CASE metodą, gali būti dekomponuojamos (ir dažniausiai yra dekomponuojamos) į **veiksnius**. Tai atitinka TNOD projektavimo idėją, jau išsakytą ankstesniuose šios metodinės priemonės skyreliuose, o taip pat TNOD projektavimo technologiją, aprašytą trečiajame šios metodinės priemonės priede.

Dalis veiksmų, taikant CASE metodą, skiriama IS projektavimo ir projektų realizavimo procesų **kontrolei** ir, apytiksliai, **valdymui**, o visi kiti veiksmai – **IS projektavimo užduotyse nurodytų praktinių rezultatų pasiekimui**. Norėčiau dar kartą atkreipti skaitytojo dėmesį į tai, jog CASE programinė įranga leidžia **dalinau automatizuoti ne tik IS projektavimo, bet ir to projektavimo procesų valdymo procedūras, t.y. leidžia padidinti ne tik projektuotojų, bet ir projekto vadovų darbo efektyvumą**. Todėl manyčiau, jog CASE sistema (ir metodą, ir jo taikymo automatizavimo priemones) **galima ir tikslinga**

taikyti ne tik IS, bet ir AVS bei integruotų automatizuotų sistemų (IAS) kūrimui.

Pažymėtina taip pat tai, jog CASE metodą aprašančioje literatūroje (pavyzdžiui, Richard Barker. CASE* METHOD TASKS AND DELIVERABLES, Richard Barker. CASE* METHOD ENTITY RELATIONSHIP MODELLING, Richard Barker. Cliff Longman. CASE* METHOD FUNCTION AND PROCESS MODELLING, Dai Clegg, Richard Barker. CASE* METHOD FAST – TRACK, Linda Hickman, Cliff Longman. CASE* METHOD BUSINESS INTERVIEWING) nurodoma, jog, kuriant IS, turi būti analizuojamos ir projektuojamos **ne tik kompiuterizuojamos, bet ir „rankinės“ procedūros**, t.y. ir tos procedūros, kurios ir po IS įdiegimo liks „rankinės“.

Tai dar kartą patvirtina šių eilučių autoriaus pozicijos ginče su kai kuriais Statistikos departamento ir jo tarnybų vadovais teisingumą ir pagrįstumą: **jei labai skirtingose šalyse (JAV ir buvusioje Sovietų Sąjungoje) visiškai vienas kito nepažįstantys žmonės, dirbdami panašų darbą, gavo tas pačias išvadas, tai reiškia, jog tose išvadose glūdi tam tikra dalis objektyvios tiesos, su kuria nesiskaityti bei atmesti iš tų išvadų išplaukiančias rekomendacijas, neįrodžius jų nepagrįstumo, būtų neišmintinga.**

Kiekvienam IS kūrimo etapui CASE metodas reikalauja nustatyti **kritinius sėkmės faktorius**, atkreipiančius projektuotojų dėmesį į tas aplinkybes, nuo kurių pirmoje eilėje priklauso IS kūrimo užduočių įvykdymo sėkmė ir IS projekto kokybė.

CASE metodo taikymo chronologinių etapų aprašymuose, apie kuriuos šioje metodinėje priemonėje bus kalbama šiek tiek vėliau, pateikiami **tipinių lokaliųjų tikslų, tipinių užduočių, tipinių jų įvykdymo rezultatų ir tipinių kritinių sėkmės faktorių sąrašai** kiekvienam etapui.

Kai kurie tipiniai kritiniai sėkmės faktoriai yra bendri daugumai chronologinių etapų ir yra ypač svarbūs. Vienu svarbiausių bendrųjų kritinių sėkmės faktorių laikomas **vartotojų ir vartotojų organizacijų vadovų dalyvavimas IS kūrime**. Tai irgi dar kartą patvirtina šių eilučių autoriaus poziciją šiuo klausimu ir atitinka sisteminio orgprojektavimo principų reikalavimus (žiūr. dvidešimt penktąjį sisteminės metodologijos principą šios metodinės priemonės 35–37 puslapiuose).

Manau, kad vartotojų ir jų vadovų dalyvavimo IS kūrime vaidmens ir reikšmės supratimas CASE metodo autorių ir taikytojų gretose ateityje dar labiau padidės, nes, kaupiantis suprojektuotų IS diegimo patirčiai, vis labiau darysis aišku, **jog reikalingas ne tik informacinės, matematinės ir techninės įrangos, bet ir organizacinės įrangos projektavimas, kompiuterizuotos ir nekompiuterizuotos veiklos organizacijoje suderinimas.**

Jau dabar pripažįstama, jog glaudus IS projektuotojų (sisteminių analitikų ir projektuotojų, DB administratorių, programinės įrangos kūrėjų ir t.t.) ir organizacijos vadovų ir darbuotojų bendradarbiavimas gali žymiai pagreitinti IS sukūrimą. Tuo, pavyzdžiui, ir remiasi **greitojo taikomosios sistemos kūrimo metodas (CASE* METHOD FAST – TRACK).**

Jei organizacijose su racionalia nusistovėjusia struktūra ir gerai organizuota veikla dar ir galima būtų apsiriboti tik interviu bei prezentacijų technologija, kaip tai rekomenduojama jau minėtoje knygoje „CASE* METHOD BUSINESS INTERVIEWING“, tai Lietuvoje dabar, kai pereinamojo laikotarpio sąlygomis vyksta principiniai ekonominių bei juridinių santykių ir pačios subjektų veiklos pasikeitimai, reikalingas ne tik pakankamai tikslus esamų struktūrų, veiklos funkcijų ir esybių bei jų ryšių „fotografavimas“ ir modeliavimas, bet ir viso to **perprojektavimas (reinžinerija)**, o to padaryti be organizacijos vadovų ir kitų jos darbuotojų aktyvaus dalyvavimo organizacinės įrangos projektavime ir diegime praktiškai neįmanoma. **Reinžinerijos reikalingumą jau pripažįsta ir CASE sistemos kūrėjai.**

Manau, kad, vystantis automatizuoto orgprojektavimo technologijai, **normatyvinių metodų taikymo lyginamasis svoris CASE technologijoje turės didėti** ir taps ne mažiau svarus, kaip deskriptyvinių metodų taikymo lyginamasis svoris. To pasekoje **orgprojektavimo technologija artės prie inžinerinės sudėtingų techninių automatizuotų**

ir automatinųjų sistemų automatizuoto projektavimo technologijos.

Atsiręs ir supratimas, jog projektuojant automatizuotas organizacines sistemas **tikslinga optimizuoti ne tik organizacijų struktūrą, bet ir jose vykstančią veiklą.** Tai privertė APS CASE kūrėjus taip **modernizuoti CASE programines priemones, kad projektuotojams būtų patogų naudotis operacijų tyrimo metodais, optimizaciniais modeliais ir optimizavimo algoritmais bei matematinės statistikos aparatu.** SQL funkcinių galimybių tam jau nebepakanka, o CASE programinių priemonių „integravimas“ su kitomis sistemomis nėra lengvas bei patogus. Atrodo, kad Oracle kompanijoje tai pradedama suprasti ir imama vis daugiau dėmesio skirti ne tik RDBVS, bet ir taikomųjų sistemų programinės įrangos projektavimo progresyvių technologijų ir priemonių kūrimui.

Kitas svarbus bendrasis kritinis IS kūrimo sėkmės faktorius – **IS kūrimo darbų įvykdymo greitumas.** Jei greitai besikeičiančioje aplinkoje IS nebus kuriama greitai, tai jau kūrimo metu ji gali dalinai pasenti ir nebeatitikti vartotojų ir jų aplinkos reikalavimų. IS kūrimo greitį, kaip jau buvo minėta, gali žymiai padidinti aktyvus vartotojų ir jų vadovų dalyvavimas IS projektavimo ir diegimo procesuose, o taip pat projektavimo automatizavimo priemonių intensyvus naudojimas. Todėl svarbu, kad Statistikos departamentas, kurdamas SIS, naudotųsi pačiomis naujausiomis ir pačiomis galingiausiomis CASE ir kitomis Oracle priemonėmis (pavyzdžiui, Designer/2000, Developer/2000), ir gerai apmokytų SIS projektuotojus naudotis jomis.

Tačiau net kuriant IS greitais tempais, projektavimo metu gali įvykti pasikeitimų, dėl kurių iškiltų reikiamybė IS projekte kai ką keisti. Tuo labiau taip gali atsitikti IS eksploatavimo metu. **Todėl kuriamoji IS turi būti pakankamai lanksti.** Tai reiškia, jog ją turi būti nesunku adaptuoti, o reikalui esant – ir perprojektuoti. CASE priemonės leidžia tai daryti žymiai greičiau ir lengviau, negu „rankinio“ projektavimo laikais.

Dar vienas labai svarbus IS diegimo ir jos funkcionavimo sėkmės bendrasis kritinis faktorius – **IS patikimumas.** Jei IS nebus pakankamai patikima – ji nefunkcionuos. Jos net įdiegti nepavyks. Pakankamo patikimumo pasiekimui reikalinga ne tik aukšta IS projekto kokybė, ne tik patikima kompiuterinė ir kitokia technika, bet ir **patikimas IS „žmogiškosios dalies“ veikimas.** „Žmogiškoji“ IS dalis **veiks pakankamai patikimai tik tada, kai žmonės bus suinteresuoti ir mokės patikimai vykdyti savo funkcijas bei gerai žinos jas ir jų vykdymo technologijas.** Iš čia išplaukia IS „žmogiškosios dalies“ veiklos projektavimo ir vartotojų apmokymo būtinybė, o taip pat vartotojų dalyvavimo IS kūrime, kaip efektyviausio jų apmokymo ir jų suinteresuotumo patikimu IS funkcionavimu būdo, tikslingumas.

Svarbiais IS kūrimo sėkmės bendraisiais kritiniais faktoriais, naudojantis CASE metodu ir priemonėmis, laikomi dar ir kruopštus bei nuoseklus IS kūrimo **darbų planavimas,** užtikrinantis **sisteminį užduočių suderinamumą** bei gautų viename chronologiniame etape **rezultatų perimamumą** kitame etape, o taip pat **aukšta IS projektavimo specialistų kvalifikacija,** jų teorinis pasiruošimas ir praktinė IS kūrimo patirtis.

Analizuodami **CASE metodo taikymo loginius etapus** (žiūr. 8 pav.), matome, jog **CASE metodas remiasi sisteminiais funkcinio ir objektinio projektavimo metodais bei jų sinteze ir savo programinėmis priemonėmis padeda realizuoti sisteminės metodologijos principų reikalavimus** (žiūr., pavyzdžiui, dvidešimtąjį, dvidešimt penktąjį ir dvidešimt šeštąjį tos metodologijos principus šios metodinės priemonės 33 ir 35-38 puslapiuose). Skaitytojas, analizuodamas anksčiau išvardintą CASE metodą aprašančią literatūrą, nesunkiai pastebės, kad CASE metodas stipriai koreliuoja su tuo, kas parašyta šios metodinės priemonės penktajame ir dešimtajame skyreliuose. Panašią koreliaciją rasime ir lygindami CASE metodą, šią metodinę priemonę ir G.Kolinso (G.Collins) knygą „Structured System Development“.

Toks čia išvardintų požiūrių ir metodologijų, susiformavusių skirtingomis sąlygomis, artumas, rodo, jog tie požiūriai ir metodologijos remiasi objektyviais faktais ir dėsniais, bendrais sudėtingų kompiuterizuotų organizacinių sistemų kūrimo procesams, nepriklausomai nuo to, kurioje šalyje ir kokioje organizacijoje jie bevyktų.

Tas faktas, kad CASE metodas realizuoja mūsų jau aprašytą sisteminio orgprojecktavimo metodologiją, sintezuojančią funkcinį ir objektinį projektavimą, leidžia čia, aprašant CASE metodą, nebekartoti to, kas buvo parašyta ankstesniuose šios metodinės priemonės skyreliuose, o iškart pereiti prie žadėtojo detalesnio CASE metodo taikymo chronologinių etapų aprašymo (žiūr. 9 pav.).

Pirmojo etapo, vadinamo **strateginio planavimo (Strategy)** etapu, **pagrindiniai lokalieji tikslai** yra šie:

- 1) bendraujant su būsimais kuriamos IS vartotojais, sukurti **organizacijos veiklos, jos struktūros ir jos sudėties bendrųjų koncepcinių modelių (funkcijų hierarchijos ir esybių ryšių diagramų) rinkinį**;
- 2) kartu su užsakovu sukurti **būsimos IS architektūros viziją**;
- 3) sudaryti **užsakovo reikalavimų ir rekomendacijų** kuriamam IS projektui sąvadą;
- 4) sudaryti **suderintą IS kūrimo planą**, atitinkantį sisteminio orgprojecktavimo metodologiją ir atsižvelgiantį į esamus organizacinius, finansinius ir techninius apribojimus.

Šis – pirmasis – IS kūrimo etapas yra labai svarbus, nes jo metu priimamas pagrindinis (strateginis) sprendimas – **kokią sistemą kursime**.

IS sukūrimo **sėkmė priklauso nuo to, kaip pilnai, tiksliai ir giliai IS kūrėjai supranta organizacijos veiklą ir jos poreikius**. Todėl **pagrindinis kritinis sėkmės faktorius šiame etape** (o taip pat ir visuose kituose IS kūrimo etapuose, ypač – antrajame, trečiajame, šeštajame ir septintajame) **yra glaudus bendradarbiavimas tarp visų projektu suinteresuotų pusių**: tarp IS projektuotojų ir užsakovo organizacijos vadovų bei tos organizacijos darbuotojų –būsimųjų IS vartotojų. SIS atveju šis ratas dar labiau išsiplečia, nes atsiranda reikiamybė bendradarbiauti ir su statistinės informacijos vartotojais, esančiais už Statistikos departamento ribų.

Neatrodė, kad dabartiniu metu Statistikos departamente pakankamai suvokiama šio svarbaus kritinio sėkmės faktoriaus reikšmė ir kad šis faktorius dabartiniu metu pakankamai išnaudojamas.

Antras svarbus kritinis sėkmės faktorius šiame etape yra **projektuotojų patirtis IS kūrimo srityje** ir, ypač, nagrinėjamos (ar panašios į ją) **organizacijos veiklos, struktūros ir informacinio ūkio tobulinimo** srityje.

Manychiau, jog būtų tikslinga, kad SIS kūrimo Koordinacinės tarybos nariai ir SIS projektavimo grupių vadovai pasistengtų plačiau susipažinti su šia patirtimi. Juk išmintingi ir atsakingi žmonės stengiasi mokytis iš svetimų klaidų, kad mažiau darytų savų, o taip pat ir iš svetimų pasiekimų, kad jie greičiau taptų ir savais pasiekimais.

Pavyzdžiui, SIS projektuotojams būtų naudinga detaliau susipažinti su CASE metodo ir jo priemonių taikymo patirtimi, projektuojant ir diegiant IS, sukaupia Bendroje Lietuvos – Austrijos įmonėje „Informacinės technologijos“.

Šios įmonės darbuotojai kūrė ir dabar palaiko bei tobulina kompiuterizuotas informacines sistemas Lietuvos Respublikos Seime, Vidaus reikalų ministerijoje, Finansų ministerijoje, Teisingumo ministerijos Teisinės informacijos centre, Vilniaus m. Griežto režimo pataisos darbų kolonijoje, Valstybiniame socialinio draudimo fonde, Lietuvos investicijų agentūroje, Lietuvos geologijos tarnyboje ir kitose organizacijose. Jie turi sukaupę nemažai projektinės ir ataskaitinės dokumentacijos, su kuria SIS kūrėjams būtų pravartu susipažinti. Tai padaryti nesunku, nes Statistikos departamentas jau gana ilgą laiką bendradarbiauja su įmone „Informacinės technologijos“. Ši įmonė yra perdavusi Statistikos departamentui ir palaiko ORACLE DBVS bei CASE programinę įrangą, pateikdama vis naujesnes jos versijas. Dar glaudesniam SIS kūrėjų ir „Informacinės technologijų“ darbuotojų bendradarbiavimui galėtų padėti ir šių eilučių autorius, nes tos įmonės vadovų ir jos darbuotojų tarpe gana daug buvusių jo studentų, kurie sutiktų padėti SIS kūrėjams ne tik apmokamų sutarčių pagrindu, bet ir iš pareigos jausmo. Dalį projektinės ir ataskaitinės dokumentacijos, sudarytos įmonės „Informacinės technologijos“ darbuotojų, kuriant

konkrečias IS, naudojantis CASE metodu ir priemonėmis, galima jau dabar gauti pas šių eilučių autorių. Būtų neprotinga visu tuo nepasinaudoti.

IS projektuotojams natūraliai gali kilti noras jau pirmajame IS kūrimo etape pilnai ir detalai išnagrinėti visą organizacijos veiklą. Toks noras galėtų būti įgyvendintas, jei organizacija būtų nesudėtinga, o taip pat tuo atveju, kai IS projektavimo darbuose aktyviai dalyvautų organizacijos vadovai ir visi jos darbuotojai. Taip ir daroma, naudojantis anksčiau minėtu **greituoju IS kūrimo metodu (CASE Method Fast – Track)**, kai IS kūrimui tiesiogiai vadovauja organizacijos vadovai ir didžiąją dalį IS kūrimo darbų vykdo patys tos organizacijos darbuotojai, gerai žinantys organizacijos struktūrą, jos funkcijas, tų funkcijų vykdymo technologijas, vykdytojų asmenines savybes ir galimybes. **Tuomet pirmieji du IS kūrimo etapai (žiūr. 9 pav.) jungiami i vieną.**

Tačiau tais atvejais, kai organizacija sudėtinga ir jos vadovai apsiriboja tik IS užsakovo vaidmeniu, o organizacijos darbuotojai IS kūrime beveik nedalyvauja, pradėti IS kūrimą nuo detalaus organizacijos tyrimo, neturint jos bendro vaizdo, būtų neracionalu, nes projektuotojai paskeštų detalėse ir, besiaiškindami jas, sugaištų neleistinai daug laiko.

Todėl CASE metode **sudėtingos organizacijos analizė, vykdančią IS projektuotojų jėgomis, skaidoma į du etapus:** strateginio planavimo (Strategy) ir analizės (Analysis) (žiūr. 9 pav.).

Pirmajame etape vykdomas pakankamai pilnas, bet nedetalus organizacijos veiklos nagrinėjimas ir sudaromas apibendrintas (sustambintas) koncepcinis organizacijos modelis, susidedantis iš organizacijos veiklos funkcijų hierarchijos ir esybių ryšių modelių (žiūr. 8 pav.).

Antrajame etape, remiantis pirmajame etape sukurtais sustambintais koncepciniais modeliais ir atsižvelgiant į pirmajame etape iškilusius klausimus, organizuojami jau pakankamai detalūs tyrimai ir sudarytų koncepcinių modelių detalizavimas bei patikslinimas. Tų modelių sudarymo metodika, taikant CASE metodą, iš esmės sutampa su metodika, aprašyta šios metodinės priemonės 5. ir 10. skyreliuose, ir skiriasi nuo pastarosios tik tuo, kad realizuojama, naudojantis projektavimo automatizavimo priemonėmis, kurios bus aprašytos šiame skyrelyje šiek tiek vėliau.

Vykdančią pirmąjį CASE metodo taikymo etapą, turi būti pasiekti šie pagrindiniai rezultatai:

- 1) nustatytos organizacijos veiklos kryptys, t.y. nustatyti ir pakankamai tiksliai bei pilnai apibrėžti organizacijos tikslai, prioritetai, apribojimai;
- 2) sudarytas bendras veiklos funkcijų hierarchijos modelis;
- 3) sudarytas bendras esybių ryšių modelis (esybių ryšių modelių rinkinys – esybių ryšių diagramos);
- 4) sudarytas rekomendacijų bei reikalavimų projektui rinkinys;
- 5) išskirti ir aprašyti svarbiausi organizaciniai bei technologiniai faktoriai;
- 6) apibrėžtos kuriamos IS ribos;
- 7) aprašyta galima kuriamos sistemos koncepcinė architektūra ir infrastruktūra;
- 8) sudarytas ir suderintas IS kūrimo planas;
- 9) išvardinti projekto resursai.

Šio etapo (Strategy) turinį sudaro šios **užduotys:**

- 1) projekto darbų vadovavimas ir administravimas;
- 2) projekto apžvalga, bendrų terminų nustatymas, projekto turinio, tikslų, prioritetų ir rezultatų nustatymas ir suderinimas, t.y. projektavimo techninės užduoties sudarymas ir suderinimas;
- 3) etapo darbų planavimas (pavyzdžiui, interviu ėmimo grafikų sudarymas ir pan.);
- 4) brifingai, interviu ir kitoks informacijos rinkimas, jos įrašymas į žodyną (Dictionary);

- 5) organizacijos koncepcinio modelio sudarymas (organizacijos vystymosi kryptių nustatymas, funkcijų hierarchijos išsiaiškinimas ir aprašymas, esybių ryšių diagramų sudarymas, duomenų srautų aprašymas);
- 6) grįžtamojo ryšio tyrimo ir projektavimo procese paruošimas ir vykdymas, rezultatų patvirtinimas;
- 7) apibendrinto organizacijos modelio dokumentavimo užbaigimas;
- 8) IS architektūros koncepcijos sukūrimas, sistemos dydžio prognozė ir rekomendacijų jos projektuotojams formulavimas;
- 9) informacinės sistemos sukūrimo apriorinio plano sudarymas;
- 10) žodinių ataskaitų apie atliktus darbus paruošimas ir aptarimas;
- 11) ataskaita užsakovo organizacijos vadovybei;
- 12) rašytinių ataskaitų projekto vadovybei paruošimas ir pateikimas.

Čia pateiktas pirmojo etapo užduočių sąrašas nurodo pagrindines ir gana sustambintas užduotis. Šios užduotys konkrečiomis sąlygomis gali būti detalizuojamos ir modifikuojamos. Kaip jau buvo minėta, formuluojant užduotis, gali būti nurodomi ir konkretūs **veiksmai**, kuriuos reikia padaryti, vykdant užduotis. Pavyzdžiui, į pirmosios užduoties sudėtį įeina:

- 1.1. etapo darbų eigos monitoringas, lyginimas su plane numatyta eiga, faktinės eigos koregavimas ir, reikalui esant, plano keitimas;
- 1.2. bendravimas su projektuotojų grupėmis, jų veiklos derinimas;
- 1.3. bendravimas su užsakovu ir užsakovo organizacijos darbuotojais, periodiškų susitikimų tarp užsakovo vadovybės ir projektuotojų rengimas;
- 1.4. reagavimas į projektuotojų ir užsakovo poreikius;
- 1.5. projektuotojų aprūpinimo CASE ir kitomis reikalingomis projektavimo automatizavimui priemonėmis organizavimas;
- 1.6. medžiagos, kuri galėtų būti reikalinga vėlesniuose projektavimo etapuose, archyvavimas;
- 1.7. ataskaitų apie padarytą darbą pateikimas.

Vykdant antrąją užduotį, nustatomas konkretus strateginio planavimo etapo darbų turinys, o taip pat būsimos ataskaitos turinys, suderinami projekto tikslai, prioritetai, užduotys ir laukiami rezultatai, paruošiama bendra projektavimo techninė užduotis, įvertinamas būsimų interviu skaičius, apibrėžiamos projekto vadovų atsakomybės sritys.

Vykdant trečiąją užduotį, apibrėžiamos organizacijos ir jos veiklos nagrinėjimo sritys, išsiaiškinama, iš ko bus imami interviu, apibrėžiamas tų interviu turinys, suplanuojamas interviu ėmimo grafikas, projekto aptarimų grafikas, išsiaiškinama, kokios srities specialistai bus reikalingi projektinių sprendimų patikrinimui, sudaromas etapo darbų planas ir jis aptariamas su projekto vadovais ir užsakovais.

Ketvirtosios užduoties turinį nusako šie darbai:

- 4.1. pasiruošimas interviu pravedimui: susipažinimas su turimais dokumentais, metodinės literatūros skaitymas, klausimynų sudarymas, interviu eigos scenarijų projektų sudarymas;
- 4.2. interviu pravedimas ir vartotojų pasiūlytos projektuotojams naudingos dokumentacijos surinkimas;
- 4.3. interviu metu ir susipažįstant su organizacijoje egzistuojančia dokumentacija (pavyzdžiui, su SNOD, TNOD ir PNOD) gautos informacijos apdorojimas ir susisteminimas;
- 4.4. pradinio organizacinio modelio, atspindinčio organizacijos ir kuriamos IS tikslus, organizacijos funkcionavimo ir vadovavimo jai poreikius, tų tikslų pasiekimo ir tų poreikių tenkinimo priemones ir būdus, sudarymas;
- 4.5. funkcijų hierarchijos pradinių diagramų paruošimas ir patikslinimas papildomuose interviu;

- 4.6. esybių ir jų ryšių pradinių diagramų paruošimas ir patikslinimas papildomuose interviu;
- 4.7. esybių, jų atributų, sinonimų, funkcijų, svarbių teiginių ir jų apibrėžimų, kilusių idėjų, kurios galėtų būti naudingos įvairiuose IS kūrimo etapuose, ir t.t. įrašymas į žodyną (Dictionary);
- 4.8. svarbiausių esybių ir jų atributų apibrėžimo sričių nustatymas ir užfiksavimas žodyne;
- 4.9. būdingų konkrečių pavyzdžių ir kitos projektuotojams naudingos informacijos užfiksavimas žodyne;
- 4.10. egzistuojančių galimybių, apribojimų, prioritetų ir kritinių sėkmės faktorių išsiaiškinimas ir užfiksavimas žodyne.

Penktosios užduoties turinį sudaro:

- 5.1. organizacijos veiklos ir vystymosi kryptių nustatymas ir užfiksavimas projektinėje dokumentacijoje;
- 5.2. organizacijos tikslų, prioritetų, apribojimų, galimybių, kritinių sėkmės faktorių projektinis formulavimas;
- 5.3. sustambinto funkcinio organizacijos modelio (funkcijų hierarchijos diagramos) sudarymas;
- 5.4. sustambinto objektinio organizacijos modelio (ERD) sudarymas ir duomenų srautų aprašymas.

Šeštąją užduotį sudaro:

- 6.1. pirmajame etape nagrinėjamų dalykų, kuriuos reikėtų peržiūrėti, atsižvelgiant į etapo vykdymo metu gaunamą informaciją, sąrašo ir peržiūros (grįžtamojo ryšio) grafiko sudarymas;
- 6.2. peržiūrai reikiamos dokumentacijos paruošimas;
- 6.3. atliktų darbų apžvalgos sudarymas;
- 6.4. atliktų darbų apsvartymas;
- 6.5. visų pakeitimų bei priimtų pasiūlymų dokumentavimas;
- 6.6. priimtų pakeitimų įgyvendinimas, atitinkamai pataisant funkcijų hierarchijos ir esybių ryšių diagramas.

Vykdam septintąją užduotį, įsitikinama, jog surinkta informacija yra pakankamai pilna ir dokumentuota tokia forma, kad ja bus galima naudotis ir vėliau – kituose IS kūrimo etapuose.

Pavyzdžiui, įsitikinama, jog visos esminės esybės, jų atributai ir ryšiai yra įtraukti į žodyną, jog paruoštos to žodyno ataskaitos, leidžiančios gauti įvairius deskriptyvinio organizacijos modelio pjūvius, t.y. nagrinėti tą modelį įvairiais projektuotojus dominančiais aspektais.

Aštuntąją užduotį, kurios tikslas yra, naudojantis iki tol sukaupta informacija, rekomenduoti būsimos IS architektūrą, sudaro:

- 8.1. organizacijos poreikių ir prioritetų bei jų loginės priklausomybės galutinis nustatymas;
- 8.2. būsimos sistemos ribų, naudojantis funkcijų modeliu, nustatymas;
- 8.3. IS architektūros koncepcinio modelio sudarymas;
- 8.4. galimų technologijų (techninė įranga, programų paketai bei kita programinė įranga ir t.t.), kurios bus panaudojamos būsimoje IS, nustatymas;
- 8.5. galimų alternatyvių sprendimų visose IS taikymo srityse nustatymas;
- 8.6. rekomendacijų organizacijos posistemiams pertvarkyti pasiūlymas.

Vykdam devintąją užduotį, sudaromas IS sukūrimo pagrindinių darbų sąrašas, išvardinami tų darbų vykdytojai bei to vykdymo kokybės kontrolieriai, nurodomi apytikriai tų darbų įvykdymo terminai ir visa tai įrašoma į „žodyną“ (Dictionary). Apriorinis IS sukūrimo

planas vėlesniuose etapuose, pereinant nuo vieno IS kūrimo etapo prie kito, yra tikslinamas, atsižvelgiant į darbų eigą ir iškilusius naujus poreikius bei pasikeitusias aplinkybes.

Dešimtos, vienuoliktos ir dvyliktos užduočių turinio čia neaprašinėsiu, nes jį pakankamai pilnai nusako tų užduočių pavadinimai.

Tikrinant užduočių vykdymo eigą (žiūr. pirmąją užduotį) ir atsižvelgiant į susiklosčiusias aplinkybes, užduočių turinys gali būti šiek tiek pakeistas bei papildytas, tačiau pagrindiniai užduočių tikslai išlieka. Išlieka ir reikalavimai pasiekti aukščiau išvardintus standartinius etapo rezultatus.

Įvykdžius visas pirmojo etapo (Strategy) užduotis ir ataskaitų metu gavus organizacijos vadovų pritarimą, išreišskiantį pasitenkinimą pasiektais rezultatais, pereinama prie antrojo etapo (Analysis) (žiūr. 9 pav.)

Analizės etapas perima, analizuoja ir patikrina pirmojo etapo rezultatus, juos detalizuoja ir papildo, sukurdamas pagrindą trečiajam – projektavimo (Design) – etapui.

Analizės etape turi būti pasiekti šie pagrindiniai **rezultatai**:

- 1) turi būti detalizuotos ir suderintos esybių ryšių diagramos, t.y. turi būti pilnai išsiaiškintos visos svarbios IS funkcionavimui esybės, pilnai apibrėžti visi jų esminiai atributai, nustatyti ir aprašyti visi ryšiai tarp esybių, tame tarpe ir ryšiai „daug su daug“; organizacijos vadovybė turi būti supažindinta su sudarytomis ERD ir patvirtinusi jas;
- 2) turi būti detalizuotos ir suderintos funkcijų hierarchijos diagramos, t.y. turi būti pilnai išsiaiškintos visos organizacijos efektyviai veiklai reikalingos funkcijos (iki pačių elementariųjų), o taip pat turi būti nustatyta, kaip, vykdant tą ar kitą funkciją, naudojami duomenys esybių pavidalu; organizacijos vadovybė turi būti supažindinta su sudarytomis FHD ir patvirtinusi jas;
- 3) naudojantis gautomis FHD ir ERD, o taip pat CASE programine įranga, turi būti sudaryta funkcijų ir esybių ryšių matricų sistema (žiūr. devintąją šios metodinės priemonės skyrelį), aprašanti duomenų naudojimą funkcijų vykdymo procesuose;
- 4) turi būti preliminarieji įvertintos kuriamoje informacinėje sistemoje naudosimų duomenų apimtys ir jų panaudojimo dažniai (tikimybės), o taip pat preliminarieji įvertinti reikalingų programinių modulių skaičius;
- 5) turi būti priimti, aptarti, suderinti ir patvirtinti sprendimai, kokiomis priemonėmis bus vykdomas IS projektavimas ir projektų realizavimas;
- 6) atsižvelgiant į analizės etapo rezultatus, atsirandančias naujas galimybes ir apribojimus bei kitas išaiškėjusias aplinkybes, turi būti pakoreguotas bei patikslintas IS kūrimo planas.

Analizės etape pagrindiniai **kritiniai sėkmės faktoriai** yra šie:

- 1) kuo didesnis vartotojų sudominimas vykdomais IS kūrimo darbais ir kuo didesnis jų įtraukimas į tuos darbus;
- 2) kruopštus etape vykdomų darbų kokybės ir jų įvykdymo pilnumo patikrinimas;
- 3) pakankamai pilnas ir pakankamai tikslus projektavimo (Design) ir realizavimo (Implementation) etapų veiksmų ir prielaidų identifikavimas ir išsiaiškinimas, kokią informaciją, reikalingą šiems etapams, reikia surinkti analizės etape;
- 4) kuriamos IS informacinio banko ir programinės įrangos sudėties, struktūros ir apimčių nustatymo kuo didesnis tikslumas;
- 5) pakankamai griežta ir tiksli etapo darbų vykdymo grafiko kontrolė, visų numatytų darbų atlikimas grafiku numatytuose laiko rėmuose.

Šiame etape turi būti įvykdytos šios pagrindinės **užduotys**:

- 1) analizės etapo darbų detalus suplanavimas, susidedantis iš:
 - a) darbų, kuriuos reikės atlikti, charakterio, apimčių, vykdytojų ir vykdymo terminų nustatymo;

- b) organizacijos veiklos ir struktūros trūkumų, aptiktų organizacijos veiklos analizės metu pirmajame IS kūrimo etape, pašalinimo terminų nustatymo ir atsakingų už tai asmenų paskyrimo;
 - c) organizacijos darbuotojų, iš kurių reikės imti interviu analizės etapo metu, o taip pat kitų IS projektavimui reikalingos informacijos gavimo šaltinių nustatymo, interviu ėmimo grafiko sudarymo;
 - d) organizacijos vadovų grupės, su kuria bus derinamas analizės etapo metu atliekamas darbas, sudėties nustatymo, tokios grupės organizavimo ir analizės etapo darbų plano su ta grupe aptarimo;
- 2) vadovavimas etapo darbų vykdymui, darbų vykdymo kokybės ir jų įvykdymo terminų kontrolė, ataskaitų apie atliktus darbus paruošimas ir kiti administraciniai darbai, susiję su IS kūrimo antruoju etapu;
 - 3) IS kūrimui ir kūrimo darbų dokumentavimui reikalingų standartų bei terminologijos, IS priimtino vartotojams kriterijų, apribojimų, į kuriuos būtina atsižvelgti, kuriant sistemą, ir kitų faktorių, kurie gali įtakoti IS kūrimo procesą ir jos funkcionavimą (pavyzdžiui, vartotojų saugumo poreikių, sistemos patikimo funkcionavimo techninių ir technologinių reikalavimų ir pan.) nustatymas, peržiūrėjimas ir patikslinimas;
 - 4) ERD ir FHD detalizavimas ir patikslinimas, gautų papildytų ERD ir FHD analizė, duomenų ir duomenų srautų apimčių bei jų vartojimo dažnių nustatymas;
 - 5) funkcijų ir esybių ryšių nustatymas, tų ryšių matricų sudarymas, detalių specifikacijų, dokumentuojančių esybių atributus, funkcijų tarpusavio priklausomybę ir interfeisus, sukūrimas;
 - 6) konsensuso apie etapo pabaigą pasiekimas, peržiūrint bei aptariant kartu su būsima IS vartotojais analizės etapo metu gautus rezultatus ir įsitikinant, jog jie tenkina vartotojus.

Analizės etapo metodologijai būdinga, kad sisteminiai analitikai privalo dirbti kartu su organizacijos vadovais ir jos darbuotojais – būsima IS vartotojais, – detaliai išsiaiškindami ir patikrindami visus reikalavimus, kuriuos turėtų tenkinti kuriamoji IS. Turi būti siekiama, kad viskas, kas yra ir bus daroma, kuriant IS, atitiktų organizacijos tikslus ir duotą naudą organizacijai.

Siekiant analizės pilnumo ir visapusiškumo, tikslinga apžvelgti kiek galima daugiau alternatyvių požiūrių į organizacijoje sprendžiamas problemas ir joje vykstančius procesus. Į nagrinėjamas problemas reikia pažvelgti ir iš funkcijų pusės, o taip pat ir naudojamos informacijos atžvilgiu, jungiant funkcinę ir objektinę analizę, derinant tai, kas buvo sužinota interviu metu, su tuo, kas randama analizuojant organizacijoje esančią dokumentaciją (normatyvinę ir ataskaitinę). Labai naudingi bandymai „pažaisti“ dalykinius žaidimus, imituojant dabartinių ir būsimų vartotojų vaidmenis, įsigilinant į galimas problemines situacijas, įsivaizduojant ir išanalizuojant perėjimo procesus nuo senos prie naujos sistemos. Visa tai padeda pastebėti ir išryškinti tai, kas turėtų būti pakeista dabartinėje organizacijos struktūroje ir veikloje, kas turėtų būti naujai suprojektuota ir įdiegta. Turi būti kruopščiai išnagrinėti ir aprašyti teisiniai, politiniai, biudžetiniai, žmonių saugumo, darbo metodų, psichologiniai ir kiti apribojimai, kad, projektuojant IS, būtų į juos atsižvelgta ir kad suprojektuotoji sistema galėtų būti realizuota.

Sudėtingos organizacijos analizė – iteracinis procesas, turintis keletą lygmenų. Pereinant nuo vieno lygmens prie sekančio, didėja analizės detalumas ir patikslinami ankstesniuose lygmenyse gauti rezultatai. Todėl tarp analizės etapo lygmenų turi egzistuoti grįžtamieji ryšiai. Svarbu taip pat suvokti, kad nors didžioji dalis analizės yra dalinimo (dekompozicijos) procesas, skaidantis analizuojamas sudėtingas esybes į vis smulkesnes ir paprastesnes dalis iki nedalomų komponentų, tačiau vis dėlto analizė, kurios paskirtis – sukaupti informaciją, reikalingą naujos sistemos projektavimui, negali būti kokybiškai įvykdyta be atvirkštinio proceso – būsimos sistemos bent jau įsivaizduojamos preliminaros

sintezės iš analizės metu išskiriamų ir projektavimo metu sukuriamų komponentų.

Trečiasis IS kūrimo, naudojantis CASE metodu ir priemonėmis, chronologinis etapas – **projektavimo (Design) etapas**.

Jo pagrindinis tikslas – surasti geriausią įmanomą IS variantą, tenkinantį analizės etapo metu išsiaiškintus ir patikslintus vartotojų poreikius, atsižvelgiantį į egzistuojančius apribojimus ir atitinkantį pageidautiną organizacijos veiklos ir jos valdymo automatizavimo bei šiuolaikinių informacinių technologijų panaudojimo lygį.

Šio etapo metu, naudojantis objektiniais modeliais (ERD), projektuojama duomenų bazė (DB) bei specifikuojamos toje bazėje būsiančios duomenų saugyklos, o naudojantis funkciniais modeliais (FHD), projektuojami taikomųjų programų moduliai ir rankinės procedūros (kitais atvejais – TNOD), ekranų, ataskaitų ir modulių ryšiai, paruošiamos programų specifikacijos ir suprojektuotos sistemos testavimo planas. Kartu projektuojama ir IS techninė įranga, reikalinga IS DB ir programinės įrangos funkcionavimui.

Informacija, sukaupta projektavimo etape, vėliau bus naudojama projekto realizavimo (Implementation) ir dokumentavimo (User Dokumentation) etapuose. Pagrindiniai standartiniai projektavimo etapo **rezultatai**:

- 1) IS architektūros projektas ir pakankamai tikslus DB apimtį ir IS programinės įrangos modulių skaičiaus įvertinimas;
- 2) DB projektas (DB sudėties ir struktūros aprašymas),
- 3) loginės ir fizinės schemos,
- 4) programinės įrangos modulių projektai,
- 5) programų modulių specifikacijos,
- 6) rankinių procedūrų specifikacijos,
- 7) IS techninės įrangos projektas,
- 8) vartotojo vadovo projektas,
- 9) suderinta perėjimo etapo strategija (IS diegimo strategija ir planas),
- 10) sistemos testavimo planas,
- 11) vykdymo dokumentacijos paruošimo planas,
- 12) peržiūrėtas ir patikslintas IS kūrimo planas.

Pagrindiniais **kritiniais sėkmės faktoriais** šiame etape laikomi šie:

- 1) pakankami gilus ir pakankamai detalus organizacijos poreikių supratimas,
- 2) geras techninės įrangos ir šiuolaikinių informacinių technologijų galimybių žinojimas,
- 3) savalaikis iškylančių problemų identifikavimas ir sprendimas.

Siekiant aukščiau išvardintų rezultatų, projektavimo etape vykdomos šios pagrindinės **užduotys**:

- 1) taikomosios programinės įrangos projektavimas,
- 2) IS duomenų bazės ir jos sukūrimo proceso projektavimas,
- 3) IS techninės įrangos projektavimas,
- 4) IS projekto sudėtinių dalių derinimas bendrame IS projekte ir programų specifikacijų sukūrimas,
- 5) sistemos testavimo plano sudarymas,
- 6) projektavimo rezultatų aptarimas su vartotojais ir etapo pabaigos patvirtinimo gavimas.

Vykdam pirmąją užduotį, nustatoma, kurios veiklos funkcijos, įeinančios į FHD bei į funkcijų ir esybių ryšių matricas, yra automatizuotinos, sudaromi tas funkcijas realizuojantys programiniai moduliai ir tų modulių specifikacijos.

Tai atliekama:

- a) išsiaiškinant elementarias veiklos funkcijas ir sudarant funkcijų modulių matricą,
- b) parenkant kiekvienam moduliiui tinkamas realizavimo priemones ir nustatant, kurios funkcijos yra automatizuotinos,

- c) nustatant taikomųjų programų sistemos sudėtį ir architektūrą bei suprojektuojant tos sistemos sudėtines dalis ir reikalingas priemones: meniu struktūras, dialoginius ekranus, automatizuotas ir rankines procedūras, įvairių klasių vartotojų interfeisus, ekranų, ataskaitų ir kitų dokumentų formas,
- d) peržiūrint ir koreguojant kartu su būsimaisiais vartotojais, DB administratoriais bei techninės įrangos projekto vadovais programinės įrangos sistemos architektūrą ir modulius.

Vykdam antrąją užduotį, objektiniuose moduluose aprašytos esybės paverčiamos DB lentelėmis ir failais bei kitais saugojamais IS informacinio banko elementais (pavyzdžiui, mikrofilmais ir pan.), sukuriama reikalingi atvaizdžiai ir indeksai. Tai atliekama:

- a) nustatant esybių realizavimo būdus (DB lentelės, failai ar kiti specifiniai duomenų saugojimo būdai),
- b) paverčiant esybės lentelėmis, failais bei kitais saugojamais IS informacijos banko elementais,
- c) nustatant IS informacinio banko ir jo sudėtinių dalių apimtį,
- d) suprojektuojant DB architektūrą, atsižvelgiant į veikimo ir saugojimo poreikius,
- e) sukuriant vartotojų atvaizdžius,
- f) atliekant suprojektuotos DB auditą, išsiaiškinant, ar projektas atitinka kuriamos IS reikalavimus, ir atitinkamai koreguojant DB projektą,
- g) naudojantis CASE programinėmis priemonėmis, sukuriant DB projektą kompiuterių atmintyje.

Vykdam trečiąją užduotį, sudaromas IS techninės įrangos projektas.

Tai atliekama:

- a) nustatant IS automatizuotų darbo vietų (ADV) „adresus“, reikiamus jų pajėgumus ir kitas charakteristikas,
- b) projektuojant ADV tinklo architektūrą ir nustatant to tinklo kanalų technines charakteristikas (praleidžiamąją galią, patikimumą ir t.t.),
- c) parenkant IS techninės įrangos projekto reikalavimus tenkinančių reikalingų įrenginių markes ir sudarant jų instaliavimo projektą bei techninės įrangos projekto realizavimo sąmatą,
- d) sudarant IS techninės įrangos funkcionavimo aprašymą (projektą).

Vykdam ketvirtąją užduotį, peržiūrimos ir suderinamos IS projekto sudėtinės dalys ir sudaromos programų specifikacijos.

Tai atliekama:

- a) peržiūrint ir derinant failų ir DB projektus, vartotojų interfeisų projektus, pradines modulių specifikacijas,
- b) nustatant konkrečius vykdytojus, kurie turės realizuoti programų modulius,
- c) paruošiant kiekvieno modulio detalią specifikaciją,
- d) sudarant suderintą visos IS ir jos programinės įrangos projektą.

Vykdam penktąją užduotį, nustatomos reikiamos sąlygos, duomenys ir tikėtini testavimo rezultatai, reikalingi visos sistemos pilno testavimo plano sudarymui. Tai atliekama:

- a) nustatant projektavimo etapo testavimo kriterijus, tokius kaip duomenų korektiškumo, jų prieinamumo ir kt.;
- b) peržiūrint ir koreguojant analizės etape apibrėžtas ir aprašytas testavimo sąlygas;
- c) nustatant sistemos dalių ir visos sistemos testavimo planą (testavimų seką):
 - c₁) atskirų modulių ir rankinių procedūrų,
 - c₂) susijusių modulių grupių,
 - c₃) sistemos sąveikavimo su aplinka ir jos koegzistavimo su aplinkinėmis sistemomis,

- c₄) sistemos ribinių galimybių,
- c₅) sistemos atstatymo galimybių, įvykus sutrikimams sistemos veikloje,
- c₆) duomenų perėmimo iš senų sistemų į naują testavimą.

Vykdamas šeštąją užduotį, įsitikinama, jog visi kiti projektavimo etapo darbai yra pabaigti, gauti to etapo vykdymo plane numatyti rezultatai, sudaryti realizavimo ir vartotojų dokumentacijos sukūrimo etapų planai.

Šios užduoties vykdymo metu:

- a) nustatomi resursų poreikiai,
- b) nustatomos ir aprašomos duomenų perkėlimo iš senos sistemos į naują procedūros,
- c) sudaromi detalūs realizavimo ir vartotojų dokumentacijos sukūrimo etapų planai,
- d) vartotojai supažindinami su projektavimo rezultatais bei IS kūrimo sekančių etapų planais, reikalui esant, sudaroma etapo pabaigos ataskaita ir atsiskaitoma projekto užsakovams,
- e) pasiekiamas projektuotojų ir projekto užsakovų susitarimas, jog projektavimo etapas yra pilnai užbaigtas ir galima pereiti prie projekto realizavimo bei vartotojų dokumentacijos sudarymo.

Projektavimo etapo metodologijai būdinga, jog pripažįstamas projektavimo proceso iteratyvumo būtinumas, kad visos kylančios projektavimo metu idėjos bei projektinių sprendimų galimos alternatyvos būtinai turi būti išbandytos ir bandomos tol, kol randamas tinkamas sprendimas, atitinkantis objektyvius kriterijus. Vartotojai ir jų vadovai, o taip pat sistemos techninio palaikymo darbuotojai turi kiek galima plačiau dalyvauti projektinių sprendimų priėmimo bei projektuotojų darbo vertinime. Toks projektuotojų, IS techninės įrangos eksploatuotojų ir IS vartotojų bendravimas, ieškant projektinių sprendinių, ypač reikalingas ir svarbus tais atvejais, kai nepavyksta rasti projektinių sprendinių, kurie tenkintų visus analizės metu išsiaiškintus reikalavimus ir būtų tuo pačiu metu pakankamai praktiški ir atsiperkantys. Tokių sprendinių neradimo priežastys gali būti labai įvairios: projektuotojų kvalifikacijos, laiko ar pinigų stoka, kokių nors kitų resursų ribotumas, organizacijos darbuotojų nepasiruošimas naudotis sudėtinga technika ir programine įranga, techninės įrangos galimybių ribotumas ir pan. Visais tokiais atvejais svarbu laiku aptarti iškilusias problemas ir nuspręsti, ar galima modifikuoti organizacijos ir atskirų vartotojų poreikius taip, kad jie galėtų būti patenkinti esamos technologijos bei technikos ir esamų resursų rėmuose.

Iteraciniai procesai gali vykti lygiagrečiai įvairiais projektavimo aspektais ir labai svarbu pasiekti jų suderinamumą, jų rezultatų sintezę bendrame IS projekte. Visa tai reikalauja nemažai laiko ir kitų resursų, tačiau reikia atsiminti, kad **iškilusios problemos, neišspręstos projektavimo metu, būtinai iškils vėlesniuose IS kūrimo ir funkcionavimo etapuose, ir tuomet jų sprendimas kainuos žymiai brangiau.**

Net ir tuo atveju, kai atrodo, jog visos projektavimo metu iškilusios problemos yra pakankamai gerai išspręstos, vėlesniuose IS kūrimo ir funkcionavimo etapuose gali paaiškėti, jog reikalingi projektinių sprendimų papildymai bei patobulinimai, t.y. galimi iteraciniai grįžimai iš vėlesniųjų etapų į ankstesnius, pavyzdžiui, iš realizacijos etapo į projektavimo ar net analizės etapą.

Ketvirtasis IS kūrimo, naudojantis CASE metodu ir priemonėmis, **etapas – realizavimo (Implementation) etapas.**

Jo metu vykdomas **programų kodavimas ir testavimas** (naudojantis atitinkamomis priemonėmis, pradedant įprastu programavimu ir baigiant „greitojo kūrimo“ būdu, naudojantis generatoriais ir kitomis CASE programinės įrangos priemonėmis), **sukuriama ir suderinama** programinės įrangos testavimui reikalinga **duomenų bazė.**

Realizavimo etapo sudėtyje galima išskirti planavimą, programinės įrangos ir programų struktūros projektavimą, programų kodavimą, testavimą iš apačios į viršų (blokų ir jų sąryšių testai) bei darbų vykdymo grafiko kontrolę.

Pagrindiniai standartiniai šio etapo **rezultatai:**

- 1) programų sukūrimas,
- 2) suderinta su programine įranga duomenų bazė,
- 3) veikiančios testuotos programos,
- 4) peržiūrėtas ir pakoreguotas organizacijos perėjimo į naują būseną projektas,
- 5) sisteminių testų rezultatai,
- 6) instaliuota ir veikianti IS kūrimo techninė ir programinė įranga,
- 7) pirmosios žinios apie IS efektyvumą.

Realizavimo etapui yra ypatingai svarbūs šie **kritiniai sėkmės faktoriai**:

- 1) kiekvieno darbo ir jo rezultatų kokybės užtikrinimas kiekviename trumpame intervale, nes dėl tamprios visų darbų sąsajos bet kokia klaida viename darbe gali atsilipti kitų darbų rezultatų kokybei,
- 2) detali kiekvienos programos veikimo analizė, atsižvelgiant į kompiuterių tinklo, procesoriaus bei įvedimo ir išvedimo įrenginių charakteristikas,
- 3) duomenų bazės ir programinės įrangos suderinimas, detali galimų situacijų analizė,
- 4) kruopštus sistemos ir jos sudėtinių dalių testavimas, tame skaičiuje ir apribojimų bei išimčių testai, konfliktinių situacijų testai, testų rezultatų analizė ir sistemos tobulinimas.

Realizavimo etapo darbų turinį sudaro šios **užduotys**:

- 1) pasiruošimas projektų analizavimui,
- 2) būsimų programų aptarimas,
- 3) programų kūrimas,
- 4) sisteminio testo paruošimas, įvykdymas, jo rezultatų analizė ir sistemos projekto koregavimas,
- 5) vadovavimas etapo darbų eigai ir jos administravimas,
- 6) etapo užbaigimas ir to užbaigimo patvirtinimo gavimas.

Vykdam pirmąją užduotį:

- a) peržiūros ir patikrinamos programų specifikacijos bei realizavimui numatyto laiko sąnaudos,
- b) kiekvienai funkcijų grupei galutinai pasirenkamos realizavimo priemonės, dokumentuojami visi techniniai sprendimai,
- c) atspausdinami detalūs blokų kodavimo ir testavimo tvarkaraščiai,
- d) paruošiama IS projekto realizavimui reikalinga techninė ir programinė įranga ir kitos priemonės, sukuriama projekto realizavimui reikalinga darbo aplinka.

Vykdam antrąją užduotį, siekiama, kad programuotojai, aptardami su projektuotojais programų specifikacijas, suprastų kuriamų programų paskirtį ir algoritmus ir būtų pakankamai pasiruošę sudaryti tas programas per realizavimo etapo darbų plane numatytą laikotarpį. Tuo tikslu:

- a) programuotojai kartu su projektuotojais peržiūri ir aptaria programų specifikacijas ir darbų apimties įverčius, suderina programų standartus;
- b) sudėtingesnėms programoms aptariamas ir sudaromas kodavimo ir testavimo planas;
- c) sudaromas visų programų testavimo planas ir nustatomi pradiniai testavimo duomenys bei galimi testavimo rezultatai;
- d) įsitikinama, jog bendrųjų procedūrų projektavimas yra atliktas nuosekliai ir programuotojai, kurie naudosis tomis procedūromis, yra gerai informuoti.

Vykdam trečiąją užduotį:

- a) atliekamas pradinis modulių kodavimas, testavimas ir koregavimas,
- b) testuojamos modulių grupės ir tikrinama, ar tinkamai funkcionuoja ryšiai tarp modulių ir programų valdymas,

- c) dokumentuojamos programos, atkreipiant ypatingą dėmesį į programų sąryšių, specialių paprogramių bei procedūrų naudojimą, palengvinančių sistemos priežiūrą.

Vykdam ketvirtąją užduotį, tikrinamas ir užtikrinamas sistemos patikimumas. Tos užduoties vykdymo metu:

- a) paruošiama sisteminiam testavimui reikalinga aplinka: duomenys, programų bibliotekos, operacinės sistemos, komunikacijos ir visa kita testavimui reikalinga įranga;
- b) atliekamas pats testavimas;
- c) dokumentuojami testavimo rezultatai, fiksuojamos visos aptiktos klaidos ir pastebėti trūkumai;
- d) koreguojamas IS projektas, jos programinė, informacinė ir techninė įranga, šalinamos testavimo metu pastebėtos klaidos bei trūkumai, sprendžiamos iškylusios testavimo metu problemos ir vėl iš naujo testuojama;
- e) kai pasiekama būseną, kada visa sistema reikiamai funkcionuoja, priimamas sprendimas apie testavimo pabaigą ir tas faktas dokumentaliai užfiksuojamas.

Vykdam penktąją užduotį:

- a) prižiūrima etapo darbų eiga, atsižvelgiant į planą, ir, reikalui esant, įvedamos korektyvios į darbų eigą, o kartais – ir į planą;
- b) peržiūrimi analizės ir projektavimo etapų rezultatai ir, reikalui esant, grįžtama prie kai kurių tų etapų darbų, jų rezultatų koregavimo bei papildymo;
- c) programavimo grupės darbai derinami su kitomis grupėmis (auditoriais, projektuotojais, techninės įrangos vadovybe, duomenų administratoriais, IS kūrimo vadovybe) ir jų darbais;
- d) periodiškai rengiami organizacijos – IS vartotojo – vadovybės ir IS projektavimo vadovų susitikimai, kurių metu aptariama IS kūrimo ir realizavimo etapo darbų eiga bei iškylančios problemos; į tuos pasitarimus, reikalui esant, kviečiami ir kai kurie eiliniai specialistai tiek iš vartotojų, tiek iš projektuotojų pusės.

Vykdam šeštąją užduotį, pasiekiamas visų kuriama sistema suinteresuotų žmonių konsensusas, t.y. bendra nuomonė apie tai, kad realizuojamoji sistema tenkina visus jos techninėje užduotyje iškeltus ir vėlesniuose IS kūrimo etapuose iškilusius reikalavimus ir tinka reguliariam eksploatavimui. Šios užduoties atlikimo metu:

- a) peržiūrimas sukurtos sistemos pilnumas, patikimumas ir tikslumas, įsitikinama, jog jie atitinka keliamus sistemai reikalavimus;
- b) įsitikinama, jog visos testavimo metu surastos klaidos ištaisytos ir pastebėti trūkumai pašalinti;
- c) peržiūrimas sukurtos sistemos tinkamumas visais kitais aspektais ir įsitikinama, kad organizacijos personalas galėtų šia sistema naudotis, žymiai padidindamas savo darbo efektyvumą.

Realizavimo etapo metodologijai būdinga, jog šis etapas vyksta lygiagrečiai su vartotojo dokumentacijos sukūrimo etapu (User Documentation). Todėl svarbu suderinti programuotojų ir dokumentacijos kūrėjų – analitikų – darbą.

Kadangi kompiuterinių programų kūrimas vyksta tam tikroje izoliacijoje nuo vartotojų, tai labai svarbu, kad bet kokie specifikacijų pakeitimai būtų priimti ne izoliuotai, bet bendru programuotojų, projektuotojų ir vartotojų susitarimu, kad tie pakeitimai būtų būtinai užfiksuoti atitinkamoje dokumentacijoje.

Norint, kad programuotojai, kurdami programas, priimtų kuo mažiau neteisingų sprendimų, siektina, jog programuotojai gerai suvoktų kontekstą, kuriame jie kuria programas.

Būtina programuotojus supažindinti su bendrais sistemos kontūrais bei su jos projektu, o taip pat su tuo, kaip jų kuriamos programos turi įsimontuoti į visą sistemą ir funkcionuoti

joje su kitomis tos sistemos komponentėmis. **Tam tikslui sugaištas laikas ir pastangos vėliau keleriopai atsipirks programinės įrangos kokybe.**

Vartotojai, savo eile, turi gauti galimybę aptarti su projektuotojais ir programuotojais visus jiems iškilančius klausimus ir pageidavimus, ko jie norėtų tikėtis iš kuriamos sistemos, ir turi būti būtinai informuojami apie tos sistemos veikimą, naudingumą, patikimumą ir kitas svarbias charakteristikas. Tuomet jie priiminės tikslesnius sprendimus ir ateityje efektyviau naudotis IS priemonėmis.

Programuotojai turi būti skatinami dokumentuoti programų kūrimo metu priimamus sprendimus ir juos periodiškai aptarinėti su vartotojais ir projektuotojais – analitikais. **Programuotojai turi stengtis neįvedinėti papildomų apribojimų duomenų apimtims ir formatams, mažinančių kuriamos sistemos lankstumą. Atvirkščiai, jie turėtų stengtis tą lankstumą didinti, kad sistema galėtų gerai funkcionuoti ne tik dabar, bet ir ateityje, pasikeitus aplinkybėms. Už tokias pastangas bei pasiekimus sistemos lankstumo didinimo srityje programuotojai turėtų būti tuo ar kitu būdu skatinami, palaikant programuotojų suinteresuotumą programinės įrangos kokybės gerinimu.** Visais tais (atvejais, kai kokių nors papildomų apribojimų įvedimas programose tampa neišvengiamas, turi būti sudarytos detalios darbo su programomis taisyklės, kurios po aptarimo su projektuotojais ir vartotojais turi būti įtrauktos į taisyklių rinkinį, specialiai saugomą duomenų bazėje kaip ir kiti metaduomenys, t.y. duomenys apie DB. Tuo rinkiniu turi būti galima naudotis ir jį keisti, naudojantis specialiais interfeisais.

Nemažai naudingų patarimų programuotojai galėtų rasti jau anksčiau minėtoje šių eilučių autoriaus knygutėje, jos skyreliuose „IAVS matematinės įrangos principai“ ir „IAVS informacinės įrangos principai“.

Penktasis IS kūrimo, naudojantis CASE metodu ir priemonėmis, etapas – vartotojo ir sistemos aptarnavimo operacijų dokumentacijos kūrimo (User Documentation) etapas.

Šiame etape **paruošiami vartotojų vadovai ir kita reikalinga personalo apmokymui, sistemos testavimui, įdiegimui ir eksploatavimui dokumentacija.** Dokumentai turi būti pakankamai detalūs, pilni, lengvai suprantami ir tikslūs, kad nekiltų neaiškumų testuojant, diegiant ir eksploatuojant IS. Dokumentacijos paruošimas turi būti užbaigtas prieš sukurtos sistemos priimtino testavimą diegimo etape.

Jau vykdant strateginio planavimo ir analizės etapus, buvo išskirtos ir dokumentuotos visos funkcijos, įskaitant ir tas, kurios buvo ir pasilieka grynai rankinėmis, bei tas, kuriose pasireišk automatizuotų bei vartotojo rankinių procedūrų sąveika. Ši sąveika projektavimo etape buvo giliau ištirta ir detaliau aprašyta – projektavimo etape jau buvo paruošti ir dokumentuoti sistemos perėjimo į naują būseną bendrieji bruožai ir planas. Realizavimo etape yra ruošama programinę įrangą aprašanti ir eksploatavimui reikalinga dokumentacija. Dokumentavimo (User Documentation) etape visa aukščiau išvardintuose etapuose ir jų dokumentuose sukaupta informacija sujungiama į vieningą visumą ir jos pagrindu paruošiami **vartotojų vadovai bei IS automatizuotosios dalies įdiegimo ir eksploatavimo (taip vadinamoji „vykdymo perdavimo“) dokumentacija.** Šiame etape kuriamos dokumentacijos tikslas – aprūpinti IS vartotojus – organizacijos vadovus ir darbuotojus, o taip pat automatizuotąją IS dalį diegiantį ir aptarnaujantį personalą (taip vadinamas „**palaikymo tarnybas**“) visa jų apmokymui ir efektyviam darbui reikalinga informacija suprantamoje ir patogioje jiems formoje.

Pagrindiniai dokumentavimo etapo **rezultatai:**

- 1) vartotojų dokumentacija (vartotojų vadovai),
- 2) IS automatizuotosios dalies diegimo ir eksploatavimo (vykdymo ir palaikymo) dokumentacija.

Šiame etape ypatingai svarbu – įtraukti vartotojus ir organizacijos vadovybę į aukščiau išvardintos dokumentacijos kūrimo ir aptarimo bei tobulinimo procesus, kad tokiu būdu būtų pasiektas kuo didesnis kuriamos dokumentacijos tinkamumas ir efektyvumas, o taip pat vartotojų ir organizacijos vadovybės pasitikėjimas savimi, diegiant ir eksploatuojant IS,

atliekant sistemos tinkamumo testus. Galima sakyti, jog vartotojų ir organizacijos vadovybės maksimalus įtraukimas į šio etapo darbus yra pagrindinis **kritinis dokumentavimo etapo sėkmės faktorius**.

Dokumentavimo etapo darbų turinį nusako šios **užduotys**:

- 1) vartotojo dokumentacijos paruošimas ir to paruošimo užbaigimas,
- 2) diegimo ir vykdymo dokumentacijos paruošimas ir to paruošimo užbaigimas,
- 3) etapo darbų vykdymo vadovavimas ir administravimas,
- 4) etapo darbų užbaigimas ir to užbaigimo patvirtinimo gavimas.

Vykdam pirmąją užduotį, sukuriama vartotojų vadovai, o ją užbaigiant, vartotojų dokumentacijos versija įtraukiama į programinės įrangos sistemą naudojimui „on–line“ režime.

Vykdam antrąją užduotį, sudaromi sistemos įdiegimui ir jos automatinės dalies funkcionavimo užtikrinimui reikalingi dokumentai – aprašymai (instrukcijos):

- a) techninės bei programinės įrangos instaliavimo,
- b) sistemos inicializavimo,
- c) sistemos paleidimo ir sustabdymo reguliarių procedūrų,
- d) sistemos konfigūravimo,
- e) duomenų archyvavimo ir atstatymo procedūrų,
- f) klaidų apdorojimo,
- g) tinklo saugumo užtikrinimo,
- h) bendravimo su kitomis sistemomis procedūrų.

Vykdam trečiąją užduotį:

- a) atliekamas darbų monitoringas, vadovaujantis etapo darbų planu – grafiku;
- b) kontroliuojama kuriamos dokumentacijos kokybė pilnumo, tikslumo, suprantamumo ir patogumo vartojime atžvilgiu;
- c) užtikrinama, kad dokumentacijos kūrėjai turėtų reikiamas CASE ir versijų kontrolės programines priemones;
- d) užtikrinama, kad sukurtoji dokumentacija atitiktų standartus.

Vykdam ketvirtąją užduotį, siekiama užbaigti dokumentavimo etapo darbus ir pasiruošti IS diegimo etapui. Tuo tikslu:

- a) peržiūrima ir aptariama su vartotojais, IS techninės ir programinės įrangos aptarnavimo personalu ir DB administratoriais visa šiame etape sukurta dokumentacija; reikalui esant, ši dokumentacija yra koreguojama, o po to gaunamas vartotojų bei automatizuotų IS posistemių palaikymo darbuotojų ir organizacijos vadovybės pritarimas bei sutikimas, kad dokumentacijos paruošimo darbai yra baigti;
- b) peržiūrimi, aptariami ir, reikalui esant, koreguojami diegimo etapo planai, paruošti realizavimo etapo eigoje.

Vykdam dokumentavimo etapo pirmąją ir antrąją užduotis, labai naudinga ir net **būtina niekada nepamiršti apie tai, kas, kada ir koku tikslu skaitys tą ar kitą dokumentą. Dokumentų kūrėjams būtina sugebėti įsivaizduoti save dokumento skaitytojo vaidmenyje ir pabandyti suvokti, kaip, kokioje formoje būtų geriausia gauti projektuojamo dokumento informaciją, ko galėtų toje informacijoje skaitytojui trūkti, o ko galėtų būti per daug ir tik trukdytų informacijos suvokimui ir darbui**. Gerai organizuotas sukurto dokumentacijos turinio aprašymas bei kitokia efektyvi **navigacijos informacijoje sistema** gali žymiai palengvinti ir pagreitinti naudojimąsi informacija.

Vartotojų dokumentacijos kūrimas, norint, kad ta dokumentacija būtų geriausiai tinkama vartojimui, matyt, yra daugiau menas, negu mokslas, nes reikalauja iš kūrėjo sugebėjimo įsijausti į situaciją, kurioje ta dokumentacija bus naudojama, o taip pat į jos vartotojo vaidmenį.

Todėl tokią dokumentaciją sukurti geriausiai pavyksta ne bandant detaliai surašyti visą tų ar kitų darbų vykdymo technologiją kitiems ir už kitus, o pavedant tuos aprašymus sukurti patiems vartotojams, suteikiant jiems visą reikalingą tokiam darbui informaciją, parodant atitinkamus pavyzdžius bei nurodant bendruosius standartus, kurių, derėtų laikytis*.

Vartotojų vadovai turi būti paruošiami, pateikiant funkcijų ir jų vykdymo aprašymus, ekranų, ataskaitų ir kitų įvairių formų turinį, struktūras ir išsidėstymą, patvirtinimų lenteles, klaidų kodus ir kitą informaciją taip, kad vartotojai galėtų suprasti suprojektuotą sistemą bei jos funkcionavimą ir patys pajėgtų susikurti sau individualius vartotojų vadovus, suprantamus ne tik jiems patiems, bet ir kitiems asmenims. Turi būti aiškiai išsakyta, ko reikalaujama ir ko tikimasi iš vartotojo, kaip jis gali pasinaudoti IS priemonėmis toje ar kitoje situacijoje.

Dažnai labai prasminga sukurti mokomąjį aprašymą („tutorial“), kuris galėtų padėti pradedančiam vartotojui suprasti, kaip suprojektuotoji sistema turėtų veikti, ir, naudojantis vaizdžiais pavyzdžiais, paaiškinti, kaip sistema gali padėti organizacijai ir pačiam vartotojui asmeniškai. Labai naudinga, kad dokumentacija būtų prieinama vartotojams ne tik „popierine forma“, bet ir kaip taip vadinamas „on-line-help“ su užuominomis, plačiais klaidų pranešimais ir kontekstine informacija. Tai žymiai palengvina vartotojų darbą.

Vykdymo dokumentacija turi būti trumpu ir tikslu kasdieninių ir kitų reguliariai atliekamų operacijų vadovu operatoriams. **Didelėse sistemose gyvybiškai svarbu turėti gerą sistemos palaikymo vadovą, sudarytą kaip specifinės informacijos iš visų sistemos dokumentų santrauką.** Šiame vadove turi būti aprašytos tos vidinės detalės, kurios leistų vykdymo vadovams ir patiems vykdytojams diagnozuoti problemines situacijas bei prognozuoti jų susidarymą. Tai ypatingai svarbu „dvidešimt keturių valandų per dieną“ darbo režime, nes leidžia vykdymo vadovybei spręsti iškilusias problemas, neskambinant projekto kūrėjams, pavyzdžiui, trečią valandą ryto.

Šeštasis IS kūrimo etapas – tai projekto diegimo etapas, kitaip vadinamas dar ir **perėjimo (Transition) etapu,** nes jo metu organizacija ir jos informacinė sistema **pervedama į naują būseną.**

Šiame etape atliekamos visos užduotys, **reikalingos tam, kad pradėtų veikti naujoji sistema: pradžioje – lygiagrečiai su senąja, o vėliau – ir viena naujoji.** Diegimo etape prasideda ir naujosios sistemos palaikymas (organizacinis, informacinis, matematinis, techninis). Perėjimas į naują būseną turi būti įvykdytas kuo mažiai nutraukinėjant organizacijos veiklą, vartotojai turi likti pasitikintys savimi ir pasiruošę dirbti naujos sistemos funkcionavimo sąlygomis. Tuo tikslu pravedamas vartotojų apmokymas ir padedama jiems atlikti naujos sistemos priėmimo testą. Šio etapo metu užbaigiamas techninės bei programinės įrangos instaliavimas ir pakraunami visi duomenys, reikalingi sistemos funkcionavimo pradžiai. Į pradinį naujos sistemos palaikymo periodą įeina pastovus jos ir jos veikimo tikrinimas ir atrandamų trūkumų šalinimas.

Vartotojų ir organizacijos vadovybės atžvilgiu diegimo etapas yra pats sunkiausias ir pats kritiškiausias, nes šio etapo metu organizacijos darbuotojams tenka dirbti lygiagrečiai su dviem informacinėmis sistemomis ir tuo pačiu metu atlikti visas organizacijos funkcijas bei spręsti organizacijai iškylančias problemas. Tai šiek tiek panašu į tai, kai bute vienu metu įrenginėjama virtuvė ir vonia ir tuo pačiu metu tame bute gyvenama, valgoma, maudomasi. Todėl svarbu, kad toks košmaras tęstųsi neilgai.

Papildomi sunkumai atsiranda ne tik dėl darbo padvigubėjimo, bet ir todėl, kad pasikeitimus pradžioje gana dažnai iš viso sunku priimti, nes nelabai kam norisi mokytis atlikinėti įprastus darbus nauju būdu, kai tu juos esi įpratęs atlikinėti senu būdu beveik automatiškai. Todėl nenuostabu, kad naujos sistemos diegimui atsiranda priešiškus. Toks spontaniškas priešiškus iš vartotojų pusės neturėtų būti interpretuojamas kaip sąmoninga neigiama vartotojų reakcija į naują sistemą, nors tai kartais ir pasitaiko, kai naujos sistemos

* Tai rodo, kad ir CASE metodologijos kūrėjai priėjo prie tos pačios išvados, kaip ir šių eilučių autorius: technologinę NOD turi kurti patys darbų vykdytojai, vadovaujant sistemos projektuotojams.

diegimas paliečia kai kurių darbuotojų interesus. **Į tą priešišumą naujos sistemos diegėjai turėtų stengtis žiūrėti kaip į eilinę beveik neišvengiamą naujų sistemų kūrimą lydinčią problemą, kurią (kaip ir daugelį kitų problemų) reikia ramiai ir protingai išspręsti, vengiant santykių aštrinio ir priešiško gilinimo.**

Diegimo metu atsiranda nemažai ir kitų problemų, susijusių su resursais, laiku, organizacijoje vykstančių procesų tęstinumu ir eile kitų iš anksto nenumatytų kliūčių. Todėl šiame etape labai svarbus gerai suderintas visų darbų planavimas ir jų eigos kokybiškas operatyvus vadovavimas, reguliari kontrolė. Kiekvienas tikslus išskylančios kliūties diagnozavimas ir išankstinio sprendimo turėjimas, kaip tokiais atvejais derėtų elgtis, gali labai padėti kuo greičiau pašalinti susidariusius nesklandumus. Tačiau didžiausia pagalba visais tokiais atvejais yra organizacijos vadovų ir jos darbuotojų kuo aktyvesnis dalyvavimas sistemos kūrimo darbuose: anksčiau, projektuojant naują sistemą, ir dabar, ją diegiant. Į naujos sistemos diegimo darbą vartotojai privalo būti įtraukti dar prieš keletą mėnesių iki diegimo etapo pradžios. Jie visi turi būti supažindinti su naujos sistemos ypatybėmis, priprasti prie griežtesnio duomenų tikrinimo, jų sutvarkymo, prie naujų darbų su kompiuteriais ir t.t. Tai galima padaryti specialaus organizacijos darbuotojų apmokymo metu, bet dar geriau, jei organizacijos darbuotojai prie viso to pripranta ir viso to išmoka glaudžiai bendradarbiaudami su IS projektuotojais projekto kūrimo metu ir tapdami to projekto bendraautorais, gerai suprantančiais ir projekto esmę, ir jo detales. Projekto kūrime aktyviai dalyvavę vartotojai po to gali apmokyti visus kitus mažiau tame darbe dalyvavusius vartotojus. Toks apmokymo būdas yra žymiai efektyvesnis už apmokymą, kurį organizuotų vartotojams vien tik IS projektuotojai: sisteminiai analitikai, dizaineriai, programuotojai, duomenų administratoriai, techninės įrangos specialistai.

Naudinga, kad jau IS diegimo metu vartotojai apčiuopiamai pajustų tą naudą, kurią jiems gali duoti nauja sistema. Todėl reikia stengtis taip suplanuoti diegimo procesą, kad būtų galima kuo anksčiau parodyti naujos sistemos privalumus.

Svarbu, kad sistemos priėmimo testas būtų atliktas kiek galima visapusiškiau pačių vartotojų ir jų vadovybės jėgomis aplinkoje, kiek galima artimesnėje organizacijos reguliaraus darbo aplinkai. Vartotojai turi taip patikrinti priimamą naują sistemą, kad gerai žinotų, ką jie gauna, įskaitant tiek naudą, tiek trūkumus.

Kai kurių nepatogumų, kurių anksčiau nebuvo, gali iškilti dėl to, kad naujoji sistema, būdama gana naudinga vartotojams, gali sudaryti jiems ir kai kurių problemų. Taip jau yra, **kad žmonės prie naujų naudingų dalykų pripranta gana greitai: jie juos priima kaip normalius ir užmiršta, kaip viskas buvo iki tol. Tačiau jie lieka labai nepatenkinti naujais netgi nežymiais nepatogumais.** Šios, atsirandančios dėl kai kurių nepatogumų problemos daug lengviau sprendžiasi, jei vartotojai dalyvavo projektavime, kartu ieškojo projektinių sprendinių ir asmeniškai įsitikino kai kurių nepatogumų neišvengiamumu, priimdami juos kaip nedidelę kainą už tą didelę naudą, kurią duoda naujoji sistema. Tai dar vienas svarbus argumentas, įrodinėjant, kodėl svarbu, kad organizacijos darbuotojai neperkeltų visų naujos sistemos kūrimo darbų vien tik ant specialiai išskirtų projektuotojų grupės pečių, bet ir patys kuo aktyviau dalyvautų tuose darbuose, projektuodami sau savo darbo aplinką ir savo darbo procesus, ieškodami geriausių sprendimų.

Diegimo etapo metu turi būti gauti ir laiku pateikti šie pagrindiniai standartiniai rezultatai:

- 1) vartotojų ir aptarnaujančio personalo mokymo medžiaga bei apmokymo planai,
- 2) apmokyti vartotojai ir aptarnaujantis personalas bei jų vadovybė,
- 3) instaliuota ir pilnai veikianti sistema,
- 4) pakrauti (gal būt, konvertuoti) duomenys,
- 5) IS palaikymo priemonės,
- 6) sukurtas ir reguliariai palaikomas sistemos bandomojo eksploatavimo metu pastebėtų trūkumų registravimo ir aprašymo žurnalas,

- 7) poinstaliacinio sistemos testavimo rezultatų ir jos funkcionavimo periodo apžvalginė ataskaita,
- 8) pilnai užbaigta IS dokumentacija.

Siekiant šių rezultatų, diegimo etapo metu vykdomos šios **užduotys**:

- 1) vadovavimas darbų eigai ir administracinių funkcijų vykdymas,
- 2) vartotojų ir aptarnaujančio personalo apmokymas,
- 3) priėmimo testo paruošimas,
- 4) techninės, programinės ir kitos konfigūracinės įrangos instaliavimas,
- 5) duomenų priėmimas ir pakrovimas,
- 6) sistemos perėjimo į naują būseną paruošimas ir įvykdymas,
- 7) sistemos palaikymas kritiniu periodu,
- 8) poimplementacinė sistemos projekto ir pačios veikiančios sistemos peržiūra ir koregavimas.

Vykdam pirmąją užduotį:

- a) atliekamas darbų monitoringas, vadovaujantis aukščiau sudarytu diegimo etapo planu;
- b) esant reikalui, koreguojamas anksčiau sudarytas darbų planas, derinant su vartotojais menkiausius to plano pakeitimus ir stengiantis, kad reguliaraus organizacijos darbo pertraukimai būtų minimalūs;
- c) organizuojami periodiški projektuotojų ir organizacijos vadovybių susitikimai;
- d) užtikrinamas visų CASE priemonių prieinamumas ir patikimas jų funkcionavimas.

Vykdam antrąją užduotį:

- a) sudaromas apmokymo tvarkaraštis, suderinamos datos ir paskiriami specialistai, kurie turės praveisti apmokymą;
- b) kruopščiai pasiruošiama apmokymams: paruošiami apmokymo vadovėliai ir kitos apmokymo priemonės (skaidrės, plakatai, pavyzdžiai, dokumentai, užduotys, techninės ir programinės priemonės);
- c) atliekamas apmokymas, kiekvienos apmokymo sesijos pradžioje pakartojant paskutinės sesijos medžiagą;
- d) atliekama reguliari mokymo efektyvumo kontrolė;
- e) numatomas, sukuriamas ir įdiegiamas reguliariai veikiantis naujai atsirandančių sistemos vartotojų ir jų aptarnaujančio personalo apmokymo mechanizmas.

Vykdam trečiąją užduotį:

- a) peržiūrimas ir, reikalui esant, koreguojamas anksčiau sudarytas sistemos priėmimo planas;
- b) paruošiamas priėmimo testas: įvertinami priimtino kriterijai, pateikiami visi reikiami duomenys, pateikiama reikiama įranga, paruošiami korektiški duomenų konvertavimo kriterijai.

Ketvirtos užduoties atlikimo metu užbaigiamas techninės ir programinės įrangos, reikalingos IS įdiegimui, funkcionavimui ir to funkcionavimo patikrinimui, diegimas, sukuriamas pilna ir gerai funkcionuojanti IS palaikymo tarnybų sistema.

Vykdam penktąją užduotį sukuriami duomenys, reikalingi naujos sistemos funkcionavimo pradžia. Jie gali atsirasti dviem būdais:

- a) įvedant juos tiesiogiai (naudojantis vartotojams skirtais duomenų įvedimo interfeisais),
- b) konvertuojant juos iš senojoje sistemoje esančių duomenų.

Dalis duomenų gali būti pakrauta į naują sistemą pirmuoju būdu, dalis – antruoju. Naudojantis antruoju būdu, reikia atkreipti ypatingą dėmesį į įvedamų duomenų patikimumą ir į jų atitikimą naujosios sistemos standartams ir kitiems reikalavimams, nes gana dažnai

senieji duomenys jų netenkina arba tenkina nepakankamai pilnai.

Vykdamas šeštąjį užduotį:

- a) patikrinamas ir, reikalui esant, koreguojamas perėjimo tvarkaraštis.
- b) peržiūrimi ir, reikalui esant, koreguojami sistemos funkcionavimo reikalavimai: saugumo dalykai (pavyzdžiui, slaptažodžiai), patikimumas ir t.t.;
- c) baigiamas pagalbinių dokumentų paruošimas, sukuriamas kasdieninio sistemos eksploatavimo grafikas;
- d) įvykdomas organizacijos perėjimas iš senos būsenos į naują;
- e) oficialiai paskelbiama, kad nauja sistema jau „gyva“, ir tas svarbus įvykis „atžymimas“ su atitinkamu iškilmingumu kaip dideliu ir sudėtingu darbu pasiektas svarbus viso kolektyvo laimėjimas, naujo organizacijos gyvavimo etapo pradžia.

Septintosios užduoties turinį sudaro kasdieninė pagalba vartotojams bei sistemos matematinę, informacinę ir techninę įrangą aptarnaujančiam personalui, o taip pat reguliarius sistemos funkcionavimo stebėjimas ir pastebėtų trūkumų šalinimas.

Atliekant aštuntąjį užduotį, surenkami atsiliepimai apie naujos sistemos funkcionavimą, įvykdoma galutinė jau funkcionuojančios sistemos peržiūra bei projektinės ir eksploatacinės dokumentacijos korekcija. Reikalui esant, koreguojamas jau veikiančios sistemos funkcionavimas.

Diegimo etapo svarbiausi **kritiniai sėkmės faktoriai** yra šie:

- 1) aktyvus vartotojų ir jų vadovybės dalyvavimas IS projektavimo ir diegimo procesuose,
- 2) gera apmokymo organizacija ir jo efektyvumas,
- 3) geras priėmimo testavimo organizavimas, sudarantis sąlygas vartotojams detaliai susipažinti su nauja sistema ir jos veikimu, savarankiškai įsitikinti sistemos veiksmumu, išbandant ją darbo sąlygomis, ir praktiškai pasimokyti ją naudoti įvairiose situacijose,
- 4) kokybiškas ir operatyvus sudėtingo ir kritiško laiko atžvilgiu naujos sistemos elementų ir visos tos sistemos įdiegimo proceso koordinavimas,
- 5) sąlygų sudarymas, kad vykdymo, techninio palaikymo ir aplikacijų palaikymo vadovybė bei personalas būtų pakankamai paruošti naujos sistemos diegimui ir aptarnavimui, turėtų visą reikalingą tokiam darbui dokumentaciją, kad galėtų giliai suprasti sistemą, žinotų, kaip diagnozuoti problemas ir kokių priemonių imtis, atsiradus techninės ar programinės įrangos darbo nesklandumams,
- 6) geras įdiegimo etapo planavimas, užtikrinantis normalų organizacijos darbą naujos informacinės sistemos diegimo metu ir leidžiantis išvengti didesnių to darbo sutrikimų, o taip pat sudarantis sąlygas aktyviam IS kūrėjų, jos vartotojų ir palaikymo sistemų aptarnaujančio personalo suderintam bendradarbiavimui IS diegimo procese,
- 7) diegiamos suprojektuotos sistemos integravimosi su kitomis egzistuojančiomis sistemomis, automatizavimo įrenginiais bei programų paketais užtikrinimas teorijoje (projekte) ir praktikoje (bandomojoje eksploatacijoje).

Septintasis (ir paskutinis) **IS kūrimo etapas** – tai **eksploatavimo etapas**, neretai vadinamas dar ir **gamybos (Production)** etapu.

Šio etapo metu prasideda ir tęsiasi normalus ir reguliarius naujos IS funkcionavimas, kurio metu vykdymo vadovybė su savo personalu teikia ir palaiko sutartą sistemos aptarnavimo lygį, o palaikymo vadovybė su savo personalu reaguoja į išsylančias problemas ir vartotojų poreikius, talkininkaudama šiam procesui. Vykdomas aptarnavimo lygio monitoringas, atliekami reikalingi papildomi techniniai patikrinimai, ištaisomos anksčiau nepastebėtos klaidos, atliekami eksploatacavimo eigoje išaiškėję būtini pakeitimai. Organizacijos vadovybė kartu su IS projektuotojais, remdamasi monitoringo rezultatais, nusprendžia, ar reikia tobulinti sistemą ir, jei reikia, tai kaip, kokiose kryptyse. Eksploatavimo etapas privalo

užtikrinti reikiamą visos sistemos funkcionavimą, reikalaujantį tik minimalaus palaikymo vadovybės įsikišimo. Šio etapo metu patikrinama, kaip sistema funkcionuoja įvairiomis ir besikeičiančiomis sąlygomis. Reikalui esant, daromi sistemos ir jos projekto pakeitimai, nepertraukiant sistemos funkcionavimo ir stengiantis pasiekti maksimalaus sistemos priimtinumą vartotojui.

Eksplotavimo etapo pagrindiniai standartiniai **rezultatai** turi būti šie:

- 1) sistemos veikimo statistiniai duomenys ir jų apdorojimo rezultatai,
- 2) patikrinimai, atstatymai, archyvų failai,
- 3) klaidų aprašymai ir jų ištaisymo ataskaitos,
- 4) kontrolės žurnalo įrašai ir pakeitimai,
- 5) sistemos patobulinimai,
- 6) sistemos audito rezultatai,
- 7) nauji reikalavimai.

Šio etapo pagrindinės **užduotys**:

- 1) vykdymo aptarnavimo užtikrinimas,
- 2) reagavimas į vartotojų poreikius,
- 3) sistemos funkcionavimo monitoringas.

Pirmosios užduoties vykdymas turi užtikrinti reikiamą aptarnavimo lygį ir reikiamą sistemos funkcionavimo palaikymą. Ją atliekant:

- a) pildomas sistemos sutrikimų žurnalas,
- b) pastebėti trūkumai diagnozuojami ir sprendžiama, kaip juos šalinti,
- c) palaikomas sistemos funkcionavimas, t.y. atliekamos nustatytos vykdymo procedūros,
- d) iškilus reikalui, patariama vartotojams, kaip reikėtų elgtis tose ar kitose situacijose (tai gali būti daroma tiesiogiai ir netiesiogiai – per „pagalbos stalelio“ dokumentaciją),
- e) po bet kokių sistemos funkcionavimo sutrikimų atstatomas sistemos funkcionavimas ir duomenys.

Atliekant antrąją užduotį:

- a) apibrėžiami ir kartu su vartotojais reguliariai peržiūrinėjami naujos sistemos funkcionavimo kokybės kriterijai: sistemos pajėgumas, techninis funkcionavimas, patikimumas, patogumas, aptarnavimo lygis ir pan.,
- b) renkami vartotojų pageidavimai, o taip pat sistemos funkcionavimo statistika, pateikiami ir aptariami su vartotojais jos apdorojimo ir analizės rezultatai,
- c) peržiūrimas vartojimo patogumas ir, reikalui esant, įvedami sistemos patobulinimai, sistemos funkcionavimui ir vartojimo patogumui pagerinti.

Vykdant trečiąją užduotį, t.y. sistemos funkcionavimo monitoringą, stebima, ar įdiegta naujoji sistema vis dar atitinka organizacijos tikslus, tikrinama, ar organizaciniai modeliai sistemoje yra dar nepasenę ir vis dar atitinka galbūt jau pasikeitusias aplinkybes, ir sukuriamos rekomendacijos, ką ir kaip organizacijoje, jos organizacinėje, informacinėje, matematinėje (programinėje) ir techninėje įrangoje derėtų tobulinti artimoje ar tolimesnėje ateityje.

Eksplotavimo etapo svarbiausieji **kritiniai sėkmės faktoriai** yra šie:

- 1) pasiryžimas pasiekti aukštą aptarnavimo lygį,
- 2) geranoriškas ir pastovus bendradarbiavimas su vartotojais ir savalaikis geranoriškas reagavimas į vartotojų poreikius,
- 3) gera sistemos veikimo ir tą veikimą gerinančių pakeitimų kontrolė.

Šiame etape už reikiamo aptarnavimo lygio suteikimą pirmoje eilėje yra atsakingi vykdymo ir palaikymo tarnybų vadovai, tačiau tuo pat metu IS projekto kūrėjų vadovai turėtų būti pasiruošę visokeriopai ir bet kada jiems padėti, ypatingai eksploatavimo etapo pradžioje,

kol dar nesukaupta pakankama darbo naujomis sąlygomis patirtis ir neišgaudytos „silpnosios“ naujosios sistemos ir jos funkcionavimo vietos.

Nors IS projektavimas šiame etape jau laikomas baigtu, tačiau automatizuoto projektavimo priemonės (CASE priemonės) ir projektavimo metu sukauptas CASE žodynas (Dictionary) dar gali būti naudingi.

Visų pirma, tos priemonės ir žodynas gali padėti, įvedant kokius nors IS projekto ir IS funkcionavimo pakeitimus. CASE žodynas, naudojantis CASE priemonėmis, tuomet gali būti patikrintas nuo pradžios iki galo, nustatant, kokį poveikį planuojami pakeitimai turės visoms kitoms IS dalims, ir tokiu būdu sužinant, ar tokių pakeitimų poveikis atitiks visos sistemos poreikius. Tai padės išvengti atvejų, kai, taisant vieną, gadinama kita.

Antra, planuojant pakeitimus sistemoje ir žodyne, jie gali būti atliekami taip, kad atitiktų sistemos struktūrą. Tai palengvina pakeitimų projektavimą ir įgyvendinimą.

CASE priemonės taip pat gali būti naudojamos sistemos konfigūracijos kontrolei pakeitimų metu, projekto kontrolei, o taip pat peržiūrėjimo ir monitoringo procesų palengvinimui. Realiai funkcionuojančios sistemos analizė ir tą funkcionavimą charakterizuojančių faktų registravimas žurnale gali padėti nustatyti realaus sistemos efektyvumo santykį su planuotu efektyvumu ir, reikalui esant, atkreipti organizacijos ir projekto vadovų dėmesį į tai, kad naujoji sistema funkcionuoja ne visai taip, kaip buvo numatyta projekte.

Reguliariai atliekamos sistemos veikimo objektyvios apžvalgos turi rodyti IS kūrėjų suinteresuotumą geru sistemos funkcionavimu, išsprendžia atsirandančias smulkias problemas ir gali padėti išvengti didelių problemų. Reikėtų stengtis, kad būtų kuo mažiau intervencijų į veikiančią sistemą iš šalies, kad vartotojai jaustųsi patys turi ir valdą sistemą, t.y., kad jie patys kontroliuotų sisteminius parametrus, vykdymo poreikius, papildomą spausdinimą, patys atlikinėtų vykdymo monitoringą.

Būtų per daug naivu manyti, jog IS kūrimo procesas jau pilnai užbaigtas, kai sistema jau reguliariai funkcionuoja organizacijoje.

Vartotojams labai svarbu, ypač pirmosiomis IS funkcionavimo savaitėmis, kad IS projekto kūrėjai greitai ir geranoriškai reaguotų į visas išskylančias problemas ir atsirandančius nesklandumus. Todėl suprojektuotai, įdiegtai ir reguliariai funkcionuojančiai naujai sistemai dar kurį laiką, kurį galėtume pavadinti **garantiniu**, turėtų nepristigti tos sistemos kūrėjų dėmesio ir globos. Tai reiškia, jog **septintasis IS kūrimo etapas gali tęstis gana ilgai ir skubėti šį etapą kuo greičiau baigti nėra tikslinga**.

Aprašę IS kūrimo, naudojantis CASE metodu ir priemonėmis, chronologinius etapus ir apibendrinami tai, kas buvo pasakyta apie jų tikslus, ryšius, užduotis, rezultatus, kritinius sėkmės faktorius, tyrimo, projektavimo ir diegimo metodiką, galima padaryti išvadą, kad tie etapai sudaro **metodiškai gerai organizuotą ir kompiuterizuotą nuoseklų organizacinės sistemos kūrimo procesą, atitinkantį sisteminės metodologijos reikalavimus ir darbų organizavimu gana artimą inžineriniam sudėtingų techninių sistemų kūrimui. CASE metodo ir jo priemonių taikymas gana miglotai apibrėžiamą sudėtingų organizacinių sistemų kūrimo meną priartina prie moksliskai pagrįsto ir gerai techniškai organizuoto inžinerinio darbo**.

Dar kartą priminsiu **pagrindinius CASE metodo ir jo taikymo bruožus**.

1. CASE metodas yra struktūrinis sudėtingų sistemų kūrimo metodas, sintezuojantis funkcinę ir objektinę sudėtingų sistemų modeliavimo, projektavimo, realizavimo ir diegimo metodologijas, deskriptyvinį ir normatyvinį sistemų aprašymo metodus.
2. CASE metodas ir CASE priemonės – tai automatizuotų organizacinių sistemų ir jų IS automatizuoto projektavimo ir realizavimo sistema, kuriai būdingas stengimasis maksimaliai panaudoti kompiuterinę techniką ir jos programinę įrangą sudėtingos organizacinės sistemos, kūrimo ir jos tobulinimo darbe, o taip pat ir pačios organizacijos funkcionavime.

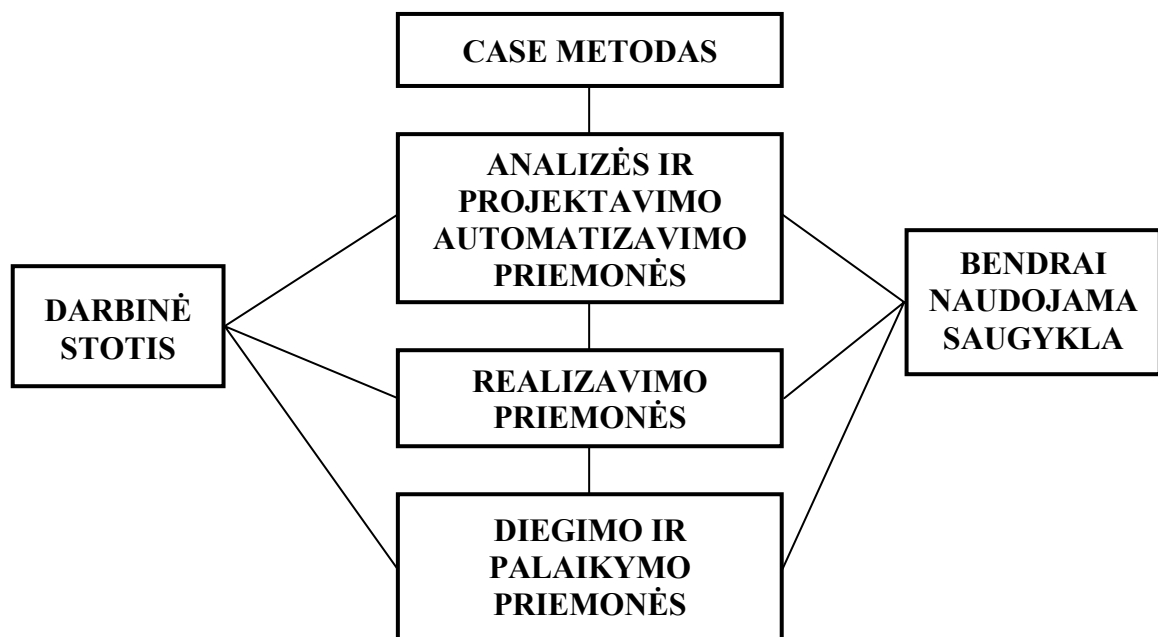
3. CASE metodo taikymui yra būdingas stengimasis maksimaliai įtraukti į organizacinės sistemos kūrimo ir tobulinimo procesą organizacijos darbuotojus ir organizacijos vadovybę. Ši tendencija, laikui bėgant, šių eilučių autoriaus nuomone, vis stiprės.
4. Visas IS kūrimo procesas, naudojantis CASE metodu ir priemonėmis, yra skaidomas į aiškiai apibrėžtus ir turinčius savo specifinius tikslus bei tų tikslų siekimo specifinę metodiką chronologinius etapus su aiškiai apibrėžtais kiekvieno etapo standartiniais rezultatais, kuriuos būtina ir tikrai galima pasiekti ir už kuriuos būtina atsiskaityti kiekvieno chronologinio etapo pabaigoje, prieš pereinant prie kito etapo.
5. Kiekvienas etapas, išskyrus patį paskutinįjį, nuosekliai ir natūraliai pereina į sekančius IS kūrimo etapus, kurie tęsia, detalizuoja, gilina ir konkretizuoja tuos darbus, kurie buvo pradėti ankstesniuose etapuose. Tokiu būdu, kiekvienas ankstesnis IS kūrimo etapas, siekdamas savo tikslų ir standartinių rezultatų, tuo pačiu metu sukuria savo rūšies trampliną (ar placdarmą) naujam kokybiniam šuoliui sekančiame IS kūrimo etape, kuriame, remiantis ankstesniuose etapuose gautais rezultatais, sprendžiami nauji uždaviniai ir taip palaipsniui vis labiau artėjama prie galutinio tikslo – organizacijos ir jos informacinės sistemos pervedimo į naują būseną. Tuo pačiu metu, pereinant nuo etapo prie etapo, palaipsniui vis labiau ryškėja kuriamos sistemos struktūra ir sudėtis, jos pagrindiniai bruožai, jos funkcionavimo algoritmai, ir visa tai paskutiniuose IS kūrimo etapuose materializuojasi į veikiančią sistemą.
6. Kiekvienam etapui formuluojamos aiškiai apibrėžtos užduotys su jų turinį nusakančiais konkrečiais veiksmais, kuriuos reikia atlikti, su tų užduočių vykdymo metu pasiektiniais rezultatais; nurodomi tam etapui būdingi kritiniai sėkmės faktoriai, į kuriuos pirmoje eilėje turėtų būti atkreiptas IS kūrėjų dėmesys.
7. Kiekviename IS kūrimo etape ypatingas dėmesys skiriamas projektų ir darbų kokybei ir jos kontrolei. **Kokybės kontrolė, taikant CASE metodiką ir priemones, – ne atsitiktinis darbų ir jų rezultatų peržiūrėjimas tais momentais, kai projekto ar darbo grupių vadovams atsiranda laisvo laiko, o iš anksto planuojamas ir reguliariai atliekamas atsakingas darbas.** Kiekvieno etapo darbų planas būtinai privalo turėti savo sudėtyje kokybės kontrolės planą, kuriame turi būti nurodyti kokybės tikrinimo momentai bei taškai, o taip pat tie veiksmai, kuriuos reikės įvykdyti priklausomai nuo kokybės tikrinimo rezultatų. Kai kuriems projektų kokybės kontrolės momentams suteikiamas netgi projekto bei jo dalies priėmimo statusas: nepasiekus reikiamos kokybės, projektas ar jo dalis nepriimami ir užduotis laikoma neįvykdyta. Todėl kokybės kontrolės momentai yra planuojami dar prieš kiekvienos konkrečios užduoties užbaigimą, kad dar liktų laiko padaryti pataisymus bei įdiegti naudingus pasiūlymus. Kai kurioms užduotims (pavyzdžiui, DB suprojektavimui ir sukūrimui) kokybės kontrolė atliekama du kartus: užduoties vykdymo pradžioje ir prieš baigiant atlikti užduotį. Tai padeda tam, kad nuo pat užduoties vykdymo pradžios būtų laikomasi nustatytų standartų ir kad užduotis būtų atlikta kokybiškai ir pilnai. Programinės įrangos kūrime kontrolės procesai organizuojami taip, kad programuotojai daugiau stengtųsi išvengti klaidų, negu jas taisyti.

Visa tai disciplinuoja IS kūrėjų darbai paverčia jį gerai organizuotu inžineriniu procesu, formalizuoja ir supaprastina didelę dalį sistemos kūrimo procedūrų, formalizuotų bei automatizuotų metodų taikymą jose ir tokiu būdu didina IS kūrėjų darbo efektyvumą. Tuo pačiu metu CASE metodo ir jo priemonių taikymas apsaugo sudėtingą ir kūrybinį organizacijos IS kūrimo procesą nuo pavojaus išsigimti į vien tik neaiškiai apibrėžtas miglotomis aplinkybėmis vykdomas paieškas, kurių metu tų paieškų dalyviai, neturėdami aiškių tikslų ir apibrėžtų būtinų užduočių, nebūdami

priversti siekti ir pasiekti konkrečių rezultatų, net būdami pilni gerų norų, pradeda neefektyviai leisti darbo laiką, praranda atsakomybės už galutinį viso darbo rezultatą jausmą ir to rezultato taip ir nepasiekia. Karti AVS kūrimo kai kuriose organizacijose patirtis rodo, jog toks pavojus yra gana realus ir kad, norint jo išvengti, reikia imtis rimtų organizacinių priemonių. Viena tokių priemonių ir yra CASE metodo ir jo programinės įrangos taikymas.

CASE metodą galima taikyti ir nesinaudojant kompiuteriais, tačiau to metodo panaudojimas tampa žymiai efektyvesnis, kai didžioji dalis sudėtingų ir dažnai klaidas iššaukiančių organizacijos analizės, IS projektavimo ir projekto realizavimo bei jo diegimo uždavinių yra atliekama, naudojantis kompiuterinėmis sistemomis ir jų programine įranga, specialiai sukurta CASE metodo taikymui ir IS kūrimo darbą automatizavimui.

Tipinio integruoto CASE priemonių rinkinio architektūrą vaizduoja 10 pav.



10 pav.

Naudodamiesi CASE programinėmis priemonėmis ir pakankamai galinga darbine stotimi (pakankamai galingu procesoriumi ir pakankamai didele ir gerai organizuota duomenų saugykla), IS projektuotojai gali kaupti ir saugoti saugykloje analizės ir projektavimo metu atsirandančią informaciją ir patogiai prieiti prie jos. Ši saugykla ir jos valdymo sistema vadinama CASE žodynu (CASE*Dictionary) arba tiesiog Saugykla (Repository).

Darbinė stotis – tai minikompiuteris ar pakankamai galingas personalinis kompiuteris bei jų vietinis tinklas, kurį turėtų sudaryti serveris (pavyzdžiui, Pentium–133, 48 RAM, 2GB HD) ir kelios ADV (pavyzdžiui, sukurtos personalinių kompiuterių Intel 486, turinčių ne mažiau 8 MB operatyvinę atmintį ir 200 MB HD bazėje).

IS kūrimo darbų automatizavimui reikalinga ne tik informacijos saugykla, bet ir įvairios modeliavimo ir projekcinės informacijos vaizdavimo, transformavimo bei apdorojimo priemonės, t.y. įvairios utilitos, tokios kaip funkcijų hierarchijos ir esybių ryšių diagrameriai, matricų diagrameriai, duomenų suderinamumo ir pilnumo tikrinimo utilitos, vieno etapo rezultatų konvertavimo į duomenis kitam etapui priemonės, formų, ataskaitų ir meniu generatoriai, DDL komandų generatorius, utilitos, generuojančios programų modulius ir DB lenteles iš funkcijų ir esybių ryšių aprašymų (pavyzdžiui, utilitos „Fastpath module mapping“ ir „Fastpath table mapping“) ir t.t. Visos tokios priemonės jungiamos į CASE programinių priemonių paketus ir sudaro **ORACLE CASE programinių priemonių rinkinį**, kurio vieno pirmųjų variantų pagrindą sudarė jau minėti **CASE*Dictionary**, **CASE*Designer** ir

CASE*Generator.

CASE programinės priemonės yra skirtos:

- 1) **užduočių vykdymo automatizavimui**, automatizuojant perėjimus nuo vieno (anksčiau gautų) rezultatų rinkinio prie kitų (naujų rezultatų) ir tokiu būdu žymiai pagreitinant bei palengvinant naujos sistemos sukūrimą;
- 2) **IS kūrimo proceso dokumentavimui**, kaupiant kompiuterių atmintyje ir dokumentuojant informaciją apie visus to proceso aspektus;
- 3) **padėti kitų – neautomatizuojamų – užduočių atlikimui**, pateikiant struktūrizuotą informaciją kompiuterių ekranuose bei printeriais spausdintose diagramose ir ataskaitose;
- 4) didelės žmonių grupės, kuriančios organizacinės sistemos projektą, darbo derinimui ir integravimui, sudarant sąlygas, kad žmonės galėtų naudotis visa visų sukaupta bendra informacija, ne tik nekliudydami, bet ir padėdami vienas kitam;
- 5) **naujos sistemos kūrimo proceso valdymo** (planavimo, kontrolės, ataskaitų sudarymo ir t.t.) **automatizavimui**.

Naujų sistemų kūrimą ir veikiančių sistemų tobulinimą labai palengvina CASE priemonėmis realizuojamos **įvairios modeliavimo technikos**:

- a) esybių ryšių modeliai (ERD) informacijos poreikių vaizdavimui,
- b) veiklos funkcijų dekompozicija ir tų funkcijų hierarchijos vaizdavimas FHD, fiksuojant ir aprašant organizacijoje vykdomas bei vykdytinas funkcijas,
- c) duomenų srautų modeliai, susiejantys veiklos funkcijas ir organizacijoje naudojamą bei cirkuliuojančią informaciją,
- d) įvairios matricos, skirtos ryšių vaizdavimui, pavyzdžiui, veiklos funkcijų ir esybių sąveikos matricos (FESM), esybių ir lentelių matricos, vaizduojančios koncepcinio ir fizinio modelio ryšius ir pan.

Kartu su modeliavimo technikomis CASE gali būti realizuojamos **automatizuotos projektavimo ir diegimo technikos**:

- a) realiacinės duomenų bazės projekto ir jos valdymo sistemos sukūrimas,
- b) taikomosios sistemos (modulių ir meniu sistemos) projektavimas,
- c) modulių generavimas,
- d) prototipų kūrimas vartotojų išspūdžiams, nuomonėms bei nuostatomis išsiaiškinti dar prieš projektavimo pabaigą,
- e) testų kūrimas ir sistemos testavimas.

CASE programinė įranga gana sparčiai vystosi ir jau galima kalbėti apie tos **įrangos kartas**.

Iki 1994 m. (o kai kuriose organizacijose – iki 1995 m. ir vėliau) buvo naudojamosi CASE priemonių rinkiniu **Oracle CASE*Tools**, turinčiu šias jau minėtas pagrindines sudėtinės dalis:

- 1) CASE*Dictionary (CASE žodyną),
- 2) CASE*Designer (CASE dizainerį),
- 3) CASE*Generator (CASE generatorių).

CASE*Dictionary – tai naujos sistemos kūrimo metu gaunamos bei sudaromos informacijos (analizės rezultatų, techninių užduočių, projektinės dokumentacijos ir t.t.) kaupimo bei saugojimo priemonė ir pati kaupiamų duomenų saugykla („žodynas“). Žodynas palaiko visus sistemos kūrimo ir gyvavimo etapus, naudodamas interaktyvias formas informacijos įvedimui, modifikavimui, peržiūrėjimui ir išvedimui. Jame galima saugoti net keletą projektų, o taip pat skirtingas projektų versijas. Visi žodyne sukaupti duomenys aprūpinami nuorodomis, leidžiančiomis juos jungti vienus su kitais įvairiais pjūviais, o taip pat patikrinti duomenų pilnumą bei suderinamumą, išvesti bei perduoti įvairių ataskaitų forma.

CASE*Dictionary turi utilitas, kurios, sąveikaudamos su duomenų valdymo bei taikomosios sistemos modulių generavimo programomis, padeda projektuotojams kaupti duomenų bazę ir IS programinę įrangą. CASE*Dictionary – daugiavartotojiška sistema, nes

su žodynu vienu metu gali dirbti daug vartotojų. Žodyne saugoma informacija prieinama jos vartotojams per meniu – formų interfeisus tiek personaliniuose, tiek mini, tiek mainframe kompiuteriuose.

CASE*Designer yra programinė priemonė automatizuotam darbui su grafine IS projekto informacija. CASE dizaineris organizuoja multi-tasking'inį windows'inį, naudojančią „pelę“, aukštos skiriamosios gebos grafinį interfeisą tarp žodyno ir jo informacijos vartotojo. Šis interfeisas leidžia, naudojantis diagramomis ir popup'iniais langais, interaktyviai modifikuoti žodyno duomenų bazę. Printeriu gali būti spausdinamos įvairios (tame tarpe ir alternatyvios, atskleidžiančios skirtingus požiūrius) diagramos: esybių sąryšių (ERD), funkcijų hierarchijos (FHD), duomenų srautų ir pan., o taip pat sudėtingos matricos, pavyzdžiui, matricos, vaizduojančios funkcijų ir esybių sąveiką (FESM). CASE*Designer sukuria patogią aplinką tokioms darbinėms stotims kaip SUN, HP, VAX bei galingiems PC, galintiems būti per tinklą įjungtiems į daugiavartotojišką paskirstytą CASE*Dictionary duomenų bazę.

CASE*Generator yra priemonių rinkinys ataskaitų (Report), formų (Forms), meniu (Menu) bei kitų interaktyvių ekraninių formų generavimui. Kitaip tariant, CASE*Generator yra įvairių generatorių šeima (formų generatorius, ataskaitų generatorius, meniu generatorius, DDL generatorius, kodo generatorius), kurioje specialūs generatoriai, naudodamiesi projektavimo metu suformuluotais poreikiais bei kita žodyne sukaupta informacija, aplinkos ir instaliavimo parametrais kaip duomenimis, generuoja atitinkamas pilnai funkcionuojančias kompiuterines programas su išeities tekstais. Be to, CASE*Generator gali pakrauti programų aprašus, struktūrą, duomenų bazės naudojimo informaciją į CASE*Dictionary saugomą projektą. Naudojantis CASE generatoriumi, taip pat galima regeneruoti jau esančią programą taip, kad ji atitiktų projekto pasikeitimus ir naujus poreikius, aplinkos ir instaliavimo parametrus bei išlaikytų programuotojų atliktus projektui neprieštaraujančius pakeitimus.

Iš taikomosios sistemos modulių aprašų, naudojantis kodo generatoriumi, gaunamos tų modulių programos, o naudojantis DDL generatoriumi (DDL – Data Definition Language – yra SQL komandų grupės), užbaigus duomenų bazės projektą, gaunami duomenų bazės aprašai, o po to sukuriami ir pati duomenų bazė (žiūr. 8 pav., kur pavaizduotas šių generatorių, o taip pat ir CASE dizainerio diagramerių vartojimas IS projektavimo procese).

Kartu su čia parašytais pagrindinėmis priemonėmis (žodynu, dizaineriu ir generatoriumi) į CASE*Tools rinkinį įeina dar ir pagalbinės priemonės. Viena tokių priemonių yra **CASE projektas** – daugiavartotojišką menedžmento sistema, palaikanti CASE metodo taikymą. Ši priemonė gali būti vartojama vertinant ir kontroliuojant užduotis ir rezultatus daugiaprojektinėje persidengiančioje aplinkoje, kur pagrindiniai resursai gali būti naudojami keliems projektams, o taip pat gali būti integruojama su žodyne esančiais elementais, pavyzdžiui, diagramomis.

Kita pagalbinė priemonė, įeinanti į CASE*Tools rinkinį, yra **CASE tiltas**. Jį sudaro priemonių šeima, kurių paskirtis – užtikrinti įvairių CASE priemonių ir jų produktų koegzistavimą. Naudojantis CASE tilto programomis, informacija iš vienos saugyklos gali būti konvertuota į standartinį saugyklų protokolą, pernešta į kitą saugyklą ar sujungta su CASE žodynu ir atvirkščiai – iš CASE žodyno perkelta į kokią nors kitą saugyklą.

Kuriant IS, gali būti naudojamos ne tik čia aprašytomis projektavimo automatizavimo priemonėmis, įeinančiomis į CASE*Tools rinkinį, bet ir kitomis programomis, reikalingomis IS funkcionavimui ir jos programinės įrangos sudarymui, pavyzdžiui, RDBVS programine įranga, duomenų apdorojimo programomis, programomis, formuojančiomis ataskaitas bei įvairių kitų dokumentų formas, programomis, skirtomis grafiniam informacijos ir jos analizės rezultatų vaizdavimui ir t.t. Todėl kartu su Oracle CASE*Tools naudojamos ir **Oracle Tools** bei **Oracle Data Access** rinkiniai, sudarančiais **Oracle Cooperative Development Environment (CDE) priemonių šeimą** (Oracle Forms, Oracle Reports, Oracle Book, Oracle Procedure Builder, Oracle Graphics, Oracle Data Browser, Oracle Data Query).

Apie CDE priemonių paskirtį galima spręsti iš jų pavadinimų todėl aš čia tų priemonių

neaprašinėsiu. Trumpai apsisistosiu tik ties viena iš jų – Oracle Graphics, kaip labai reikalinga ir naudinga statistinės informacinės sistemos darbuotojams ir statistinės informacijos vartotojams.

Naudojantis Oracle Graphics, galima generuoti daugiau nei 50 tipų diagramas (stulpelines, linijines, skritulines, išbarstytas, Ganto ir kt.) bei grafikus. Tam nereikalingas joks specialus programavimas. Kiekvienas grafikas ar diagrama gali būti pritaikyta konkrečiai situacijai, operuojant tokiais grafikų parametrais kaip ašys, tinklas, žymių charakteristikos. Programuotojai gali nurodyti, kaip nuo informacijos duomenų bazėje turėtų priklausyti diagramų spalvos bei jų užpildymo būdai, pridėti papildomus grafinius laukus ir tekstą. Pačios diagramos gali būti paverstos dinaminiais objektais bei, naudojantis „pele“, savo rūšies „klavišais“, leidžiančiais vartotojui greitai ir patogiai prieiti prie kitų informacijos vaizdavimo priemonių, pavyzdžiui, ekraninių formų, pateikiančių vartotojus dominančią papildomą informaciją. Tokiu būdu, naudojantis jau minėtais grafiniais klavišais, dinaminiais grafiniais objektais bei besikartojančiais aktyviais langais, galima patogiai jungti visas Oracle CDE dalis į vieną visumą. „Multimedia“ leidžia jungti dinامينius vaizdus ir garsus, o realaus laiko sistemos – kurti objektus, kurių vizualiniai atributai automatiškai keičiasi, reaguodami į besikeičiančius duomenis. Atvira Oracle Graphics architektūra leidžia šiai sistemai dalintis duomenimis ir procesais su Oracle Forms, Oracle Reports bei Oracle Book sistemomis. Yra įmanoma diagramą arba ekraną, sukurtą Oracle Graphics, įdėti į Oracle Forms priemonėmis parašytą formą. Taip grafinis duomenų vaizdas tiesiogiai susiejamas su tekstu. Programuotojai taip pat gali integruoti Oracle Graphics su kitais programų paketais ir jungti su paprogramėmis, parašytomis bet kuria iš 3GL kalbų. Su Oracle Graphics sukurtos taikomosios sistemos gali importuoti ir eksportuoti duomenis bei grafiką iš daugelio duomenų šaltinių. Naudojantis Oracle Open Gateway, atvaizdžiai gali tiesiogiai pasiekti ir keisti duomenis, saugomus ne Oracle tipo duomenų šaltiniuose, pavyzdžiui, IBM SQL/DS arba DB2. Naudodamiesi prie Oracle Graphics prijungta PL/SQL kūrimo terpe, programuotojai gali rašyti, kompiliuoti ir taisyti PL/SQL kodą, t.y. naudotis PL/SQL – procedūriniu SQL kalbos tęsiniu, turinčiu sąlygas, ciklus ir kitus operatorius, būdingus klasikinėms procedūrinėms kalboms.

Iš to, kas čia buvo pasakyta apie Oracle Graphics, akivaizdu, kad tokios priemonės buvimas SIS įrangos sudėtyje padėtų žymiai praturtinti SIS galimybes ir palengvintų statistinės sistemos darbuotojams jų darbą su statistine informacija.

Nuo 1994 – 1995 m.m. CASE*Tools rinkinys buvo pradėtas keisti nauju paketu, pavadintu **Oracle CDE CASE Bundle** arba **CDE2 CASE**.

CDE2 CASE išplėtė Oracle CASE technologiją, įvesdamas veiklos reinžinerijos priemones, naujus interfeisus į projekto duomenų saugyklą, naujas tos saugyklos tvarkymo priemones, diagramas visam sistemos gyvavimo ciklui, Oracle Forms 4.5 ir Oracle Reports 2.5 programų generatorius.

CDE2 CASE sudaro šios sudėtinės dalys:

- 1) **Oracle Process Modeller 1.0** – organizacijos veiklos modeliavimo ir reinžinerijos priemonė, naudojanti srautų diagramas ir video bei audio medžiagą svarbių įvykių iliustravimui;
- 2) **Oracle Systems Modeller 1.0** – organizacijos, kaip sistemos, modeliavimo priemonė, turinti visą rinkinį diagramų ryšių, esančių organizacijoje, vaizdavimui;
- 3) **Oracle Systems Designer 1.0** – priemonė, skirta duomenų bazės projektavimui, vaizdavimui ir programų modulių bei procedūrų logikos modeliavimui, naudojantis tiek veiklos, tiek PL/SQL programavimo kalbos terminais;
- 4) **Oracle Forms Generator 4.5** – generatorius, turintis naujas atvirų duomenų bazių kūrimo galimybes, didesnę generuojamų programų pasirinkimą, užtikrinantis generuojamų programų ODBC suderinamumą bei jų integraciją su OLE 2.0, VBX;

- 5) **Oracle Reports Generator 2.5** – generatorius, galintis generuoti ataskaitas pagal šablono Template Data Model objektus ir tokiu būdu kurti ataskaitas su parametrais, PL/SQL, Oracle Forms moduliais;
- 6) **Oracle Server Generator 2.5** – generatorius, skirtas Oracle 7.1 laidai, turintis duomenų bazės objektų reversą (on–line duomenų bazės struktūrų atvaizdavimą projekto duomenų saugykloje) ir generuojantis serverio PL/SQL vaidmenį bei objektus replikavimo užtikrinimui;
- 7) **Oracle Repository Administrator 1.0** – projekto duomenų saugyklos valdymo priemonė, tvarkanti duomenų saugykloje esančias taikomas sistemas, jų versijas, darbuotojų priėjimą prie projekto; ši priemonė turi naujas iškrovimo, sujungimo bei pakrovimo galimybes, skirtas taikomųjų sistemų projektų instaliavimui, ir leidžia kitoms CDE2 CASE paketo sudėtinėms dalims (Oracle Systems Designer, Oracle Process Modeller, Oracle Systems Modeller) valdyti projekto duomenų saugykloje kaupiamus duomenis, naudotis jais.

Tuo pačiu metu buvo pradėtas keisti ir Oracle Tools rinkinys. Jo vietoje buvo pasiūlyta nauja versija – **CDE2 Tools**, susidedanti iš:

- 1) Oracle Forms 4.5,
- 2) Oracle Reports 2.5,
- 3) Oracle Graphics 2.5,
- 4) Oracle Procedure Builder 1.5.

Buvo sukurta ir nauja Oracle Data Access versija – **CDE2 Data Access**, susidedanti iš:

- 1) Oracle Browser 2.0,
- 2) Oracle Data Query 4.0.

Visi trys paketai – **CDE2 CASE**, **CDE2 Tools** ir **CDE2 Data Access** – sudarė naują kliento – serverio sistemų kūrimo terpės **Cooperative Development Environment laidą (CDE2)**.

Tačiau ši laida išsilaikė dar trumpiau, negu pirmoji.

Jau 1995 sausio 17 d. ORACLE korporacija pristatė naują programinių priemonių šeimą – **Oracle Workgroup/2000**, susidedančią iš:

- 1) **Oracle 7 Workgroup Server** – Oracle 7 RDBVS versijos darbo grupėms, dirbančioms operacinėse sistemose Windows NT, Novell NetWare, OS/2, SCO Unix, Solaris x 86, PowerMac ir UnixWare bei simetrinėse daugiaprocesorinėse sistemose;
- 2) **Personai Oracle 7** – vienvartotojiškos Oracle 7 RDBVS asmeniniams kompiuteriams, dirbantiems Windows 3.1, Windows 95, PowerMac ir OS/2 operacinėse sistemose;
- 3) **Oracle Power Objects** – naujos kartos objektiškos programinės priemonės, skirtos taikomųjų sistemų kūrėjams, veikiančios Windows 3.1, Macintosh System 7 ir OS/2 terpėse, turinčios grafinį vartotojo interfeisą ir atliekančios objektų kūrimą bei įjungimą į taikomas sistemas projektuojant, o ne programuojant;
- 4) **Oracle Objects for OLE** – priemonių rinkinio, veikiančio Windows 3.1 terpėje ir sugebančio prijungti prie Oracle RDBVS programines priemones, atitinkančias OLE 2.0 standartą.

Buvo taip pat pranešta, jog ORACLE korporacija artimiausiu metu numato tiekti programinę priemonę **Oracle Radio Agents (Oracle Mobile Asents)** mobiliems duomenų bazių vartotojams, galinčią dirbti lokaliuose kompiuterių tinkluose, naudojantis bevielėmis telefono linijomis.

Dabartiniu metu Oracle 7 duomenų bazių serverio programinė įranga sukonfigūruota jau **trims skirtingų dydžių grupėms: vienam vartotojui (Personal), darbo grupei („Workgroup“) ir įmonei ar kokiai nors kitai organizacijai (Enterprise)**. Vienam

vartotojui skiriama ne tik Personai Oracle 7, bet ir **Personal Oracle Lite** programinė įranga.

Jau sukurta ir **trečioji Oracle 7 serverio versija 7.3**. Šioje Oracle 7 duomenų bazės valdymo sistemos versijoje pateikiama daug naujovių, skirtų duomenų saugykloms, paskirstytoms taikomosioms sistemoms, tranzakcijų vykdymui realiaame laike, darbui su daugialypėmis sistemomis, plečiamų sistemų valdymui, įmonių lygio sistemų valdymo tobulinimui. Pilnas versijos Oracle 7.3 programinių priemonių rinkinys leidžia dirbti su visų rūšių duomenimis: reliaciniais, tekstiniais, daugiamaciais, daugialypiais („multimedia“). Be to, šios versijos programinė įranga sudaro sąlygas visapusiškai Oracle 7 bazėje esančios informacijos integracijai su World Wide Web (WWW): klientai, naudodamiesi Internetu, gali pasiekti Oracle 7 duomenis iš išorės ir iš vidaus. Versija 7.3 yra suderinta su versijomis 7.0, 7.1 bei 7.2 ir nereikalauja jokių taikomųjų sistemų pakeitimų ar duomenų reorganizacijos.

Dar daugiau naujų galimybių atsiras, pasirodžius naujos kartos Oracle produktams, pavyzdžiui, **Oracle 8 – objektinės orientacijos DBSV ir programinei įrangai. pritaikytai darbui su tinklo kompiuteriais (NC) Internet ir Intranet tinkluose.**

1995 m. pasirodė ir **patobulintas IS kūrimo priemonių rinkinys** – paketų Oracle CDE CASE Bundle ir Oracle CDE2 Tools modernizavimo rezultatas – programinės priemonės **Designer/2000** ir **Developer/2000**.

Designer/2000 yra CASE technologiją realizuojančios programinės priemonės, leidžiančios pasirinkti tinkamiausią IS kūrimo eigą, automatizuoti ir realizuoti ją. Šis programinių priemonių rinkinys skirtas sistemų bei procesų modeliavimui ir projektavimui, o taip pat automatiniam Developer/2000 programų generavimui.

Paketą Designer/2000 sudaro:

- 1) Process Modeller (procesų modeliotojas),
- 2) Systems Modeller (sistemų modeliotojas),
- 3) Systems Designer (sistemų projektuotojas),
- 4) Oracle Forms Generator (formų generatorius),
- 5) Oracle Reports Generator (ataskaitų generatorius),
- 6) Oracle Server Generator (serverio generatorius),
- 7) Repository Administrator (saugyklos administratorius).

Process Modeller – tai programinė priemonė, skirta veiklos procesų aprašymui bei modeliavimui ir to aprašymo bei modeliavimo automatizavimui. Ja naudojantis, galima vaizduoti ne tik pačius procesus, bet ir jų tarpusavio ryšius, jų vykdytojus ir t.t. Šioje programinėje priemonėje operuojama tokiais objektais kaip veiklos funkcijos, duomenų srautai, duomenų saugyklos, veiklos funkcijų vykdytojai, organizacijos padaliniai. Gali būti panaudoti įvairūs „multimedia“ objektai: garsas, video ir kt. Ši priemonė tampa ypač reikšminga, kai IS kuriama veiklos procesų reinžinerijos būdu, pradedant nuo fundamentalių procesų išaiškinimo ir jų tobulinimo, nes tuomet organizacijos veiklos analizė ir perprojektavimas dažniausiai pradedamas nuo Process Modeller panaudojimo. Galima sakyti, jog Process Modeller – tai **pagrindinė veiklos procesų reinžinerijos programinė** priemonė, jos „įėjimo taškas“.

Systems Modeller sudarytas iš esybių, funkcijų hierarchijos ir duomenų srautų grafinio vaizdavimo programinių priemonių ir skirtas informacinių sistemų modeliavimui, tų reikalavimų, kuriuos turėtų tenkinti kuriamoji IS, išsiaiškinimui. Systems Modeller operuoja informacinę sistemą sudarančiomis esybėmis, gali naudotis tais pačiais objektais, kaip ir Process Modeller, ir yra integruotas su Systems Designer. Systems Modeller dažniausiai tampa **pirmąja programine darbo priemone**, kai IS kuriama **informacijos inžinerijos būdu**. Šis būdas taikomas tada, kai organizacijos veiklos poreikiai yra sunkiai apibrėžiami ir pati veikla silpnai struktūrizuota, o sprendimus priima ir už veiklą yra atsakingas ne vienas asmuo. Tuomet reikalinga pradėti nuo bendrųjų informacinių poreikių išsiaiškinimo, jų identifikavimo, informacinių procesų projektavimo. Tam kaip tik geriausiai ir tinka Systems Modeller, integruojant jį su Systems Designer. Žinoma, Systems Modeller yra naudingas ir tada, kada IS kuriama sistemų reinžinerijos būdu, t.y. kada IS kūrimas pradedamas ne nuo

„nulinio“, o nuo esamos informacinės sistemos modernizavimo ir praplėtimo.

Systems Designer yra skirtas IS projektavimui, apibrėžia duomenis, veiklos taisykles, procedūras, tranzakcijas, duomenų naudojimą, užklausas. Į jo sudėtį įeina duomenų bazės ir taikomosios sistemos modulių grafinio vaizdavimo ir projektavimo automatizavimo programinės priemonės, modulių logikos navigatorius, Designe/2000 generatorius, valdančių parametrų navigatorius, projektavimo pagal nutylėjimą priemonės. Systems Designer naudojamas, kuriant IS bet kuriais būdais, tačiau nuo jo dažniausiai pradedama tada, kai IS kuriama jau minėtu **greitojo taikomosios sistemos kūrimo būdu**.

Generatoriai (Oracle Forms Generator, Oracle Reports Generator, Oracle Server Generator) yra programinės priemonės, skirtos Designer/2000 saugykloje saugomų duomenų automatiškam transformavimui į kliento ir serverio programų kodus, kuriuos generatoriai sukuria naudodamiesi deklaratyviais aprašymais ir taisyklėmis. Procedūriškai aprašomos tik sudėtingos veiklos taisyklės. Generatoriai kartu su Systems Designer gali automatiškai atstatyti ir projekto duomenų saugykloje išsaugoti esamų sistemų projektus. Tai reikalinga, kuriant IS reinžinerijos būdu. Naudojantis generatoriais, programos, jų išvaizda, navigacija ir t.t. sukuriami, vartojant šablonus, generatorių parametrus ir reikšmes pagal nutylėjimą. Naudojantis Designer/2000 generatoriais, generuojamos Developer/2000 taikomosios programos.

Repository Administrator – tai programinė priemonė, skirta projekto duomenų saugyklos ir jos daugiavartotojiško eksploatavimo organizavimui bei valdymui. Designer/2000 duomenų saugykla ir jos valdymo sistema – tai mūsų jau anksčiau aprašyto CASE*Dictionary patobulinta versija. Naudodamiesi Repository Administrator ir Designer/2000 duomenų saugykla, IS kūrėjai gali saugoti šimtus ir tūkstančius analizės ir projektavimo metu patalpintų į saugyklą įvairių pranešimų, bendrai naudotis jais kiek nori kartų, modifikuoti bei papildyti juos. Visa tai leidžia visiems projektuotojams naudotis bendrais apibrėžimais, atsižvelgti į visos sistemos ir kolegų keliamus reikalavimus ir tokiu būdu sudaro patogias sąlygas bendram didelio kolektyvo darbui, kuriant sudėtingą IS. Tokio darbo palengvinimui naudojamos įvairios priemonės, įeinančios į Designer/2000 sudėtį. Tai ir saugyklos objektų navigatorius, ir įvairių grafikų bei matricų braižymo priemonės ir daugiavartotojiškos saugyklos funkcionavimo organizavimo bei jo valdymo priemonės ir t.t.

Kitas labai svarbus ir reikalingas IS projektuotojams programinių priemonių paketas, pasirodęs 1995 m. – tai **Developer/2000**. Jis pakeičia mūsų jau minėtąjį CD2 Tools ir yra skirtas formų, ataskaitų ir grafinio vaizdavimo taikomųjų programų kūrimui. Developer/2000 sudaro:

- 1) Oracle Forms,
- 2) Oracle Reports,
- 3) Oracle Graphics,
- 4) SQL*Plus,
- 5) Oracle Plus,
- 6) Oracle Book,
- 7) Procedure Builder.

Kuriant taikomas sistemas, galima naudotis ir mūsų jau anksčiau minėtu **Oracle Power Objects** – daug galimybių turinčia nebrangia priemone Visual Basic programuotojams.

Detalus visų aukščiau išvardintų programinių priemonių ir jų vartojimo instrukcijų aprašymas pareikalautų labai daug vietos ir šioje metodinėje priemonėje, skirtoje IS kūrimo koordinacinės tarybos nariams bei Statistikos departamento, jo tarnybų bei skyrių vadovams, būtų nereikalingas. Tiems gi IS projektuotojams, kuriems detalesnė informacija yra gana aktuali, parekomenduosiu Oracle korporacijos leidžiamą literatūrą. (ją galima gauti įmonėje „Informacinės technologijos“):

- 1) Michael J. Corey, Michael Abbey. Oracle: A beginners guide,
- 2) George Koch, Kevin Loney. Oracle: The complete reference, 3rd edition,

- 3) Kevin Loney. Oracle DBA Handbook,
- 4) David McClanahan. Oracle Developer's guide,
- 5) Scott Urman. Oracle PL/SQL programming,
- 6) Michael J. Corey, Michael Abbey, Daniel J. Dechichio Jr. Tuning Oracle,
- 7) Rama Velpuri. Oracle backup and recovery handbook,
- 8) Thomas B. Cox. Oracle Workgroup Server Handbook,
- 9) Bruce Kolste, David Peterson. Oracle Power Objects Handbook,
- 10) Richard Finkelstein, Riek Greenwald, Kasu Sista. Oracle Power Objects Developer's Guide,
- 11) Robert J. Muller. Oracle Developer/2000 Handbook,
- 12) Hugo Toledo. Oracle Networking.

Pirmoji iš šių knygų – „Oracle: pradedančiojo vadovas“ – pateikia lengvai suprantamą visiems ORACLE produktų vartotojams ir pakankamai išsamią informaciją apie tuos produktus. Šią knygą galima rekomenduoti visiems tiems, kurie dar beveik nieko nežino apie Oracle korporacijos produkciją, tačiau nori greitai su ja susipažinti.

Tiems, kuriems informacijos, pateikiamos pirmoje knygoje, nepakanka, galima rekomenduoti antrąją knygą – „Oracle: pilnas žinynas.“ Joje pateikiamos enciklopedinės žinios apie Oracle.

Trečioji knyga – „Oracle duomenų bazės administratoriaus vadovas.“ Joje pateikiama išsami informacija apie tai, kaip efektyviai tvarkyti Oracle duomenų bazę.

Ketvirtoji knyga – „Oracle aplikacijų kūrėjo vadovas“ – moko, kaip kurti efektyvias ir patikimas aplikacijas.

Penktoji knyga – „Oracle PL/SQL programavimas“ – pataria, kaip kurti ir testuoti Oracle PL/SQL kalba parašytas programas, kaip atrasti klaidas jose, kaip tas programas derinti. Šią knygą gali skaityti ir pradedantieji, nes joje pateikiama PL/SQL apžvalga.

Šeštoji knyga – „Oracle optimizavimas“ – aprašo, kaip derinti Oracle duomenų bazę, siekiant maksimalaus visos sistemos darbo efektyvumo.

Septintoji knyga – „Oracle išsaugojimo ir atstatymo vadovas“ – aiškina, kaip sklandžiai administruoti Oracle sistemą, kaip išvengti įvairių sutrikimų bei ką daryti, kai jie jau įvyko.

Aštuntoji knyga – „Oracle Workgroup serverio vadovas – moko, kaip panaudoti įvairias Workgroup serverio galimybes.

Devintoji knyga – „Oracle Power Objects vadovas“ – dėsto Power Objects pradmenis, programavimą vienvartotojiškoje duomenų bazėje, o taip pat sudėtingesnių aplikacijų kūrimą bei Oracle BASIC.

Dešimtoji knyga – „Oracle Power Objects taikomųjų sistemų kūrimo vadovas“ – aiškina, kaip kurti objektiškai orientuotas kliento – serverio taikomąsias sistemas Oracle Power Objects priemonėmis. Kartu su knyga pateikiamas Oracle Workgroup/2000 programinės įrangos CD-ROM diskelis su bandomosiomis Power Objects ir Oracle 7 versijomis.

Vienuliškoji knyga – „Oracle Developer/2000 vadovas“ – moko, kaip, naudojantis paketo Developer/2000 programinėmis priemonėmis, greitai sukurti tvarkingas ir efektyvias taikomąsias sistemas. Pateikiami taikomųjų sistemų kūrimo pagrindai, siūlymai ir sprendimai, rekomenduojami taikomųjų sistemų programuotojams. Kartu su knyga pateikiamas Developer/2000 programinės įrangos CD-ROM diskelis su bandomosiomis Developer/2000 ir Personalinio Oracle 7 versijomis.

Dvyliktojoje knygoje – „Oracle kompiuterių tinklai“ – aprašoma, kaip Oracle sistemos jungiasi LAN, WAN, belaidžiuose ir Internet kompiuterių tinkluose. Joje apžvelgiami tokie kompiuterinius tinklus realizuojantys produktai, kaip SQL*Net, Web System, Mobile Agents, ODBC ir OLE2.

SIS projektuotojams rekomenduočiau būtinai susipažinti ir su knygomis, išvardintomis šios metodinės priemonės 119 puslapyje.

Pirmojoje šių knygų – „CASE*Method. Užduotys ir rezultatai“ – Richard Barker, Europos Oracle viceprezidentas, sistemų kūrimo žinovas, pateikia išsamų CASE metodo vadovą integruotų sistemų kūrimui. Analitikai, projektų vadovai bei informacinių sistemų kūrėjai šioje knygoje ras detalizuotą kuriamos sistemos įgyvendinimo ciklo užduočių planą bei kritinių sėkmės faktorių formuluotes, kokybės užtikrinimo būdus ir pan. Leidinyje taip pat išdėstytos duomenų normalizavimo taisyklės ir technikos charakteristikos, praktiniai patarimai prototipų kūrėjams. Ši knyga skirta ne tik teorinėms studijoms, bet ir duoda atsakymus į daugelį kasdienio darbo klausimų.

Antroji knyga – „CASE*Method. Esybių ryšių modeliavimas“ – skirta veiklos ir sistemų analizės, strateginio planavimo ir duomenų bazių projektavimo bei administravimo specialistams. Autorius – tas pats Richard Barker – žinomas duomenų bazių ir CASE technologijų teoretikas, praktinio jų panaudojimo ekspertas – detalai nagrinėja esybių ryšių modeliavimo techniką, pateikia daug konkrečių pavyzdžių, kaip duomenų modeliavimo technika panaudojama kokybiškų integruotų sistemų kūrimo procese. Knygoje pateikiamas ir svarbiausių terminų žodynas. Galima sakyti, jog knyga „Esybių ryšių modeliavimas“ gali būti panaudota ir kaip studijų vadovėlis, ir kaip kasdienio praktinio darbo vadovas.

Trečiosios knygos – „CASE*Method. Veiklos funkcijų ir procesų modeliavimas“ – autoriai, žinomi Oracle CASE specialistai, lakoniškai pateikia pagrindinius veiklos sistemų funkcionalumo apibrėžimo būdus, o aprašydami funkcijų ir procesų modeliavimą, savo aprašymą iliustruoja praktiniais pavyzdžiais. Knyga skirta sistemų analitikams, projektų vadovams ir kitiems organizacines sistemas kuriantiems specialistams, norintiems išmokyti veiklos funkcijų, procesų bei įvykių modeliavimo, naudojantis CASE programinėmis priemonėmis. Knygoje yra svarbiausių terminų žodynas.

Ketvirtoji knyga – „CASE*Method. Greitaeigis metodas“ – aprašo jau anksčiau minėtą greitąjį IS kūrimo metodą, pasitelkiant organizacijos vadovus bei darbuotojus ir automatizuoto projektavimo priemones.

Penktojoje knygoje – „CASE*Method. Veiklos interviu“ – ypatingas dėmesys kreipiamas sistemų kūrimo sėkmę lemiančiam organizacijos reikmių ir jos darbo aplinkos pakankamai pilno ir gilaus supratimo būtinumui ir tokio supratimo pasiekimo būdams. Knygos autoriai – Linda Hickman ir Cliff Longman – ilgametę strateginio konsultavimo patirtį ir pasaulinę praktiką įgiję specialistai. Jie aprašo pagrindines interviu bei prezentacijų technikas, esminius šių darbo formų kriterijus, interviu bei prezentacijų principus. Knyga suteikia gerą progą analitikams, projektų vadovams ir kitiems sistemų kūrimo specialistams bei konsultantams pabandyti atlikti aktyvaus klausymo, užsirašinėjimo bei kūrybiško klausimų pateikimo pratimus. Joje įdomiai aprašomas nežodinis bendravimas, detalai pateikiamos interviu etapų – planavimo, vykdymo, konsolidavimo – užduočių formuluotės. Knyga tinka savarankiškam mokymuisi ir kasdieniam naudojimui darbo procese.

Kadangi storų knygų, parašytų anglų kalba, skaitymas nėra lengvas ir greitas darbas, tai, tur būt, ne kiekvienam SIS projektuotojų grupės sisteminiam analitikui, programuotojui ar duomenų bazės projektuotojui bei administratoriui pakaks laiko, ryžto ir kantrybės perskaityti net pačias reikalingiausias knygas. Todėl jiems galima rekomenduoti ir kitą – greitesnę – reikalingų žinių bei įgūdžių įsigijimo formą – kvalifikacijos kėlimą įmonės „Informacinės technologijos“ specialiai organizuojamuose mokymosi kursuose. Tuo labiau, kad ir skaitančiam aukščiau išvardintas knygas kils nemažai klausimų, į kuriuos jis norėtų gauti atsakymus, pasikonsultuoti pas kvalifikuotus, patyrusius specialistus. Tiesa, mokymasis kursuose nemažai kainuoja, tačiau nebūtina kiekvieną SIS projektuotojų grupės narį siųsti į tokius kursus: pakanka apmokyti vieną–kitą darbuotoją, o šie jau galės apmokyti visus kitus, konsultuoti juos, iškilus klausimams.

O SIS kūrimo Koordinacinės tarybos nariams bei probleminių grupių ir Statistikos departamento tarnybų bei skyrių vadovams pilnai pakaks šios metodinės priemonės perskaitymo ir susipažinimo su jos priedais. Jiems nei ką tik išvardintų knygų skaitymo, nei IT organizuojamų brangiai kainuojančių mokymosi kursų neprireiks. Iškilus kokiam nors specialiam klausimui, konsultaciją bus galima gauti pas specialiai paruoštus Statistikos departamento Informatikos tarnybos Skaičiavimo centro specialistus

Tiesa, būtų naudinga, kad Koordinacinės tarybos nariai bei probleminių grupių vadovai turėtų šią tokį supratimą apie VIKT (Valstybinių institucijų kompiuterių tinklą), jungiantį jau 12 institucijų (LR Seimą, Vyriausybę, Ekonomiką, Finansų, Ryšių ir informatikos, Užsienio reikalų, Valdymo reformos ir savivaldybių reikalų, Vidaus reikalų ministerijas, Statistikos ir Muitinės departamentus, Vilniaus savivaldybę ir valstybinę įmonę „Infostuktūra“, kuriai pavestos šio tinklo valdymo centro funkcijos), apie VADIS (Vyriausybės administracinę informacijos sistemą), pradėtą kurti 1995 m., ir apie jau anksčiau pradėtą kurti savivaldybių informacinių sistemų projektą. Tai padėtų, formuluojant išorinius reikalavimus, kuriuos turėtų tenkinti SIS – svarbi valstybinės informacinės sistemos dalis.

Būtų taip pat tikslinga žinoti Eurostato reikalavimus, turėti informaciją apie kitų valstybių statistines informacines sistemas (Lenkijos, Vokietijos, Danijos, Norvegijos, Austrijos, Latvijos, Estijos ir kt.), apie Interneto bei SprintNet'o galimybes, apie duomenų perdavimo tinklą „Vilniaus DAPTAK“, apie duomenų perdavimo protokolus TCP/IP ir X25, apie KKR (kadastrai, klasifikatoriai, registrai) sistemą. Manoma, jog būtų naudinga organizuoti specialų seminarą SIS kūrėjams ir Statistikos departamento tarnybų ir skyrių vadovams, supažindinantį juos su išoriniais reikalavimais, kuriuos turėtų tenkinti valstybinė statistinė informacijos sistema. Matyt, reikėtų organizuoti ir SIS kūrėjų susitikimą su VĮ „Infostuktūra“ bei įmonės „Informacinės technologijos“ vadovais ir VADIS projektuotojais, su Ryšių ir informatikos bei Valdymo reformų ir savivaldybių reikalų ministerijų atstovais.

Baigiant skyrelį apie automatizuotą informacinių kompiuterizuotų sistemų projektavimo sistemą CASE, norėtųsi pailiustruoti konkrečiu pavyzdžiu, kiek pagreitina CASE metodo ir priemonių panaudojimas informacinių sistemų kūrimo procesą.

Vilniaus Griežtojo režimo pataisos darbų kolonija 1995 m. sudarė sutartį su įmone „Informacinės technologijos“ dėl kolonijos informacinės sistemos (apimančios atvykusiųjų į koloniją nuteistųjų registravimą, nuteistųjų judėjimo registravimą, papildomos informacijos apie nuteistuosius registravimą, operatyvinės informacijos tvarkymą, visos turimos informacijos, susijusios su nuteistaisiais, teikimą, informacijos apie kolonijos darbuotojus tvarkymą) sukūrimo. Pagrindinė užduotis, kuriant šią sistemą, buvo svarbiausių kolonijos skyrių – įskaitos, režimo, operatyvinio bei kadrų – darbo automatizavimas, užtikrinant informacijos korektiškumą, slaptumą, operatyvumą bei patikimumą, patogų autorizuotų vartotojų prieigą prie jiems reikalingos informacijos.

Tokios IS projektavimui įmonėje „Informacinės technologijos“ buvo sudaryta darbo grupė iš trijų darbuotojų – buvusių šių eilučių autoriaus studentų – Manto Būtos, Evaldo Pamakščio ir Olego Volkovo. IS kūrimo darbuose aktyviai dalyvavo ir kolonijos darbuotojai: Jonas Kriukas – kolonijos viršininkas, Arvidas Ižička – kolonijos viršininko pavaduotojas ROD, Rita Balzarienė – įskaitos skyriaus viršininkė, pulk. Pavalkis – režimo ir priežiūros skyriaus viršininkas, operatyvinio skyriaus darbuotojai Arūnas, Kęstas ir Romas.

Darbai buvo pradėti 1995 m. rugsėjo mėn. Sudarant pradinį darbų planą, sistemos analizei buvo išskirta apie 1 mėn. laiko (nuo 1995.09.20 iki 1995.10.29) ir bendram projektavimui (Design) – taip pat apie 1 mėn. (nuo 1995.10.30 iki 1995.11.29). Realizavimo (Implementation) darbai buvo pradėti 1995.12.01 ir baigti 1996.01.08. Kaip pavyzdį galima pateikti realizavimo darbų planą, nurodantį preliminaras darbų trukmes darbo dienomis.

Vykdytojas – Mantas Būta

Darbo pavadinimas	Darbo trukmė
Nuteistojo požymiai	2
Adresai	2
Giminės ir artimieji	3
Pranešimai	10
Nuteistųjų vardinis sąrašas	2
Etapavimų registravimas	2
Etapavimų pranešimai	2
Apibendrinimas	2

Vykdytojas – Evaldas Pamakštys

Darbo pavadinimas	Darbo trukmė
Visos sistemos meniu	3
Klasifikatoriai	12
Įskaitos skyriui reikalingas auditas	6
Darbuotojai	3
Dokumentacijos apibendrinimas	2

Vykdytojas – Olegas Volkovas

Darbo pavadinimas	Darbo trukmė
Nuteistojo atvykimo registravimo kortelė	4
Nuteistojo atvykimo registravimo kortelės tvirtinimas	1
Nuotraukos	2
Bausmės pakeitimai	2
Nuteistojo atvykimo registravimas	3
Nuteistojo paleidimas + išvykimas	3
Nuteistojo registravimo kortelės ataskaita	3
Nuteistųjų judėjimo informacija	6
Dokumentacijos redakcija	2

Sistemos diegimas (perkėlimas – Transition) ir vartotojų apmokymas buvo pradėtas nuo 1996.01.09.

Kaip pavyzdį pateiksiu naujos informacinės sistemos diegimo planą kolonijos įskaitos skyriuje.

Pavadinimas	Terminas
Tinklo įranga	iki 1995.12.25
Kompiuteriai ir serveris turi būti skyriuje	iki 1996.01.03
Kompiuteriai ir serveris turi būti paruošti sistemos perkėlimui	iki 1996.01.07
Sistemos perkėlimas	1996.01.09
Kompiuterių ir serverio instaliavimas kolonijoje	1996.01.10
Apmokymas ir supažindinimas su dokumentacija	1996.01.11–1996.01.17

Kada	Ką daro	Kas	Kam
1996.01.11.	Klasifikatorių pristatymas ir darbas su jais	Evaldas	Įskaitos skyriui
1996.01.12.	Darbuotojų registravimas	Evaldas	Sekretorei
1996.01.12	Nuteistojo registravimas bei skanerio panaudojimas	Olegas	Įskaitos skyriui
1996.01.15	Etapavimų registravimas ir giminių registravimas iki 13 val.	Mantas	Įskaitos skyriui
1996.01.15	Nuteistųjų judėjimo registravimas nuo 13 val.	Olegas	Įskaitos skyriui
1996.01.16.	Judėjimo ataskaitos ir registravimo kortelės ataskaita iki 13 val.	Olegas	Įskaitos skyriui
1996.01.16	Etapavimų ataskaita ir pranešimų spausdinimas nuo 13 val.	Mantas	Įskaitos skyriui
1996.01.17	Audito pristatymas iki 12 val.	Evaldas	Įskaitos skyriui
1996.01.17	Sistemos pridavimas nuo 12 val.	Visi	Viršininkui

Gana didelės apimties KIS (Kolonijos informacinė sistema) buvo sukurta ir įdiegta gana greitai – per septynis mėnesius. Buvo automatizuota apie 120 veiklos funkcijų, duomenų saugojimui buvo sudaryta daugiau kaip šimtas lentelių. Buvo sudaryta visa reikalinga dokumentacija: Vartotojo vadovas (1) (bendras visai sistemai) – 38 psl., Vartotojo vadovas (2) (įskaitos skyriui) – 118 psl., Vartotojo vadovas (3) (režimo skyriui) – 53 psl., Vartotojo vadovas (4) (operatyviniam skyriui) – 76 psl., Analizės etapo ataskaita (36 psl.) su priedais – ERD ir FHD – 19 psl., Projektavimo etapo ataskaita – 99 psl. Su visa šia dokumentacija galima susipažinti įmonėje „Informacinės technologijos“, o taip pat pas šių eilučių autorių.

Greitą ir kokybišką šios informacinės sistemos sukūrimą garantavo pažangios darbo organizavimo metodikos, besiremiančios CASE metodu, ir darbo automatizavimo priemonių – Designer/2000 ir Developer/2000 – efektyvus panaudojimas.

Manychiau, jog SIS kūrėjams būtų naudinga detaliau susipažinti su VGRPD kolonijoje įvykdytais darbais, sukurta KIS ir jos dokumentacija – kaip konkrečiu CASE metodo ir CASE programinės įrangos panaudojimo pavyzdžiu. Galima būtų pasimokyti ir darbų organizavimo, nes dabartiniu metu Statistikos departamente SIS kūrimo procesas dar nėra pakankamai gerai organizuotas ir patys kūrimo tempai yra, mano manymu, per lėti.

KIS kūrimo pavyzdys rodo, jog, naudojantis CASE metodu ir jo programine įranga, informacinės sistemos kūrimą galima žymiai paspartinti. Tiesa, neretai galima išgirsti nuomonę, jog CASE programinė įranga gana brangiai kainuoja ir todėl ne kiekviena organizacija pajėgi ją įsigyti. Su tokia nuomone tenka sutikti, tačiau tos įrangos licenzijos ir techninio aptarnavimo kainos sparčiai mažėja ir dabartiniu metu yra palyginti jau nebe tokios didelės.

Kainų kitimą pailiustruosiu keliais skaičiais, paimtais iš D.Lazausko straipsnio „Oracle priemonių platinimo tendencijos“ leidinyje „Informacinės technologijos“, 4 (1996 m. pavasaris), 4–5 psl.

Licenzijų kainos:

- 1993 m. Oracle C ASE* Tools 22 499 USD
- 1994 m. Oracle CDE CASE Bundle 12 500 USD
- 1995 m. Designer/2000 4 995 USD

Techninio aptarnavimo kainos:

- 1993 m. Oracle CASE* Tools 3 375 USD
- 1994 m. Oracle CDE CASE Bundle 2 345 USD
- 1995 m. Designer/2000 1 875 USD

Priemonės taikomosios programinės įrangos kūrimui – Developer/2000 – licenzijos kaina šiuo metu yra 4 995 USD (kaip ir Designer/2000), o jos techninio aptarnavimo kaina

yra 3 125 USD metams.

Sumažėjus kainoms, Oracle programinės priemonės tapo prieinamos daugeliui Lietuvos organizacijų, tame skaičiuje ir Lietuvos aukštosioms mokykloms. Tuo labiau, kad pagal Oracle korporacijos aukštojo mokslo įstaigų rėmimo programą Oracle programinė įranga teikiama aukštosioms mokykloms su didele nuolaida.

Vilniaus universitetui pagal TEMPUS programą pateikta programinė įranga Oracle 7 RDBSV, Oracle Text Server, Oracle Designer/2000, Oracle Developer/2000. Matematikos ir informatikos fakultete jau skaitomas teorinis Oracle CASE tematikos kursas, organizuojama speciali mokymo klasė su Pentium darbo stotimis ir PowerPC serveriu (operacinė sistema AIX).

Vilniaus Gedimino technikos universitetas yra įsigijęs Oracle 7 RDBSV ir Oracle Developer/2000. Yra įrengta mokymo klasė. Oracle programinė įranga dirba lygiagrečių procesorių MPP architektūroje, nes VGTU įsigijo IBM RS-6000 kompiuterį.

Kauno technologijos universitetas instaliavo Designer/2000 bei Developer/2000 ir įsirengė aštuonių darbo vietų greito tinklo segmentą. Oracle technologija jau pradėta naudoti mokyme, Informacinių sistemų katedros diplomantai jau ruošia diplominius projektus Oracle Designer/2000 priemonėmis.

Iš to, kas čia pasakyta, galima padaryti išvadą, kad greitai Lietuvoje bus pakankamai daug specialistų, sugebančių naudotis Oracle programinėmis priemonėmis, įvaldžiusių CASE metodą ir galinčių produktyviai dalyvauti IS kūrime.

Tai, kad kompiuterizuotų IS kūrimo tempai ir kuriamų IS skaičius sparčiai auga, o taip pat auga ir IS projektavimo specialistų paklausa, liudija kad ir toks faktas: 1993 m. įmonės „Informacinės technologijos“ bendrosiose pajamose pajamos, gautos už IS kūrimą, sudarė 27,8%, o 1996 m. devynių mėnesių pajamose – jau 51,6%. Pajamos už specialistų mokymą IT organizuojamuose kursuose 1993 m. sudarė tik 0,6% visų pajamų, o 1996 m. devynių mėnesių pajamose – jau 14%.

Malonu pažymėti, kad Statistikos departamentas taip pat turi visą reikalingą SIS kūrimui Oracle programinę įrangą ir kai kurie Departamento darbuotojai jau moka naudotis CASE priemonėmis. Juos reikėtų pasitelkti visų kitų SIS kūrimo grupės programuotojų, projektuotojų, sisteminių analitikų apmokymui, organizuojant praktinius užsiėmimus. Tokį apmokymą galima vykdyti lygiagrečiai su SIS kūrimo darbais.

Tuo ir baigsiu vienuoliktąją šios metodinės priemonės skyrelį apie CASE metodą ir jo priemones. Pereisiu prie dvyliktojo skyrelio, kuriame aprašysiu vieną siūlomų SIS kūrimo darbų organizavimo variantų.

12. SIS KŪRIMO DARBŲ ORGANIZAVIMAS

SIS kūrimo darbų sėkmė priklauso ne tik nuo pasirinktojo esamos sistemos tobulinimo bei naujos sistemos kūrimo metodo, kurio aprašymui buvo skirti ankstesnieji šios metodinės priemonės skyreliai, bet ir nuo to, kaip darbai paskirstomi vykdytojams ir kaip parenkami tie vykdytojai, į kokią struktūrą jie suburiami, kaip vykdomas darbų eigos monitoringas, kokia tų darbų ir jų rezultatų kokybės kontrolės sistema, koks vykdytojų atsakomybės lygis ir kaip jis palaikomas, kaip organizuojama vykdytojų atsiskaitomybė už savo darbą ir jo rezultatus.

Skaitydamas vienuoliktąją šios metodinės priemonės skyrelį, skaitytojas galėjo pastebėti, kokią didelį dėmesį IS kūrimo darbų organizavimui skiria CASE metodo ir CASE priemonių kūrėjai: **CASE priemonės ne tik automatizuoja IS kūrimo darbus, bet ir padeda juos organizuoti bei valdyti.**

Suprasdami IS kūrimo darbų organizavimo svarbą bei norėdami dar labiau ją pabrėžti ir pasidalinti su skaitytoju tokio organizavimo patirtimi, šioje metodinėje priemonėje tam skirsime atskirą – baigiamąjį – jos skyrelį kurį pradėsime nuo **kuriamos bei tobulinamos sistemos S ir kuriančios bei tobulinančios sistemos R** sąvokų įvedimo, sistemos R organizavimo proceso ir jos sąveikavimo su sistema S nagrinėjimo.

Kuriant bei tobulinant sudėtingą sistemą, kurią sutrumpintai žymėsime S (mūsų metodinėje priemonėje sistema S – tai SIS ir bendrai visa Lietuvos statistinės informacijos sistema su Statistikos departamentu kaip savo svarbiausiąją sudėtinę dalimi), **tenka organizuoti sistemos S kūrimo bei tobulinimo sistemą**, kurią sutrumpintai žymėsime R.

Sistemos R paskirtis – analizuoti sistemoje S ir jos aplinkoje susidariusią situaciją bei toje situacijoje iškylančias problemas, priimti iš tos analizės išplaukiančius projektinius problemų sprendimus ir realizuoti juos, pervedant sistemą S į naują būseną.

Priklausomai nuo sistemos S tipo, jos sudėtingumo ir kitų aplinkybių sistema R gali būti kuriama kaip sistemos S dalis (jos padalinys), gali būti sudaroma šalia sistemos S ir neįeiti į jos sudėtį, o taip pat gali įeiti į sistemos S sudėtį tik kuria nors viena savo dalimi, palikdama kitą savo dalį už sistemos S ribų. Galimas net toks kraštutinis atvejis, kai sukuriama tokia sistema R, į kurios sudėtį įeina visa sistema S kaip sistemos R dalis.

Kai sistema S – techninė sistema, kurią reikia suprojektuoti ir pagaminti, sistema R kuriama ir egzistuoja šalia sistemos S, neįeidama į jos sudėtį. Neretai tokiais atvejais kuriama net ne viena, o kelios sistemos R (pavyzdžiui, viena – sistemos S projektavimui, kita – tos sistemos gamybai).

Kai sistema S – organizacinė sistema, tai galimi visi aukščiau išvardinti sistemų S ir R santykiavimo atvejai, tačiau dažniausiai sistema R kuriama kaip kuriamos ar tobulinamos sistemos S sudėtinė dalis, kuri, trūkstant organizacijoje S kai kurių specialistų, gali būti papildyta specialistais iš kitų organizacijų.

Vienas pagrindinių šio – dvyliktojo – metodinės priemonės skyrelio tikslų – **pasiūlyti ir aprašyti vieną galimų sistemos R struktūros ir jos organizavimo variantų**, kuris šių eilučių autoriui atrodo labiausiai tinkamas, kuriant SIS, ir pateisino save AVS kūrimo praktikoje.

Organizuojant sistemą R ir jos veiklą, naudinga laikytis šių principų:

1. Sistema R turi būti kuriama iš organizacijos S (kurios tobulinimas, diegiant naują techniką ir naujas darbo technologijas – R sistemos tikslas) **darbuotojų. Specialistai iš šalies gali būti samdomi, tik jų nesuradus organizacijoje S.**

2. Sistema R turi būti pradedama kurti nuo viršaus – nuo sistemos R veiklos koordinavimo organo – Koordinacinės tarybos (KT) sudarymo. KT sudaroma iš organizacijos S ir jos padalinių vadovų, o taip pat iš organizacinių sistemų veiklos racionalizavimo ir automatizavimo specialistų bei dženeralistų.

3. KT vadovas ir organizacijos S vadovas turėtų būti vienas ir tas pats asmuo. Tik ypatingais atvejais (kai sistema S labai didelė ir jos vadovas labai perkrautas darbu) leidžiama, kad KT pirmininku būtų paskirtas vienas iš organizacijos S vadovo pavaduotojų. Šia išimtaine galimybe nederėtų piktnaudžiauti: net esant gana dideliame organizacijos S vadovo apkrovimui, vis dėlto geriau, kai KT pirmininku tampa sistemos S vadovas, pasiskirdamas operatyviniam einamajam darbui Koordinacinėje taryboje ir organizacijos S vadovavimui veiklius, kompetetingus ir atsakingus pavaduotojus ir pasilikdamas sau principinių sprendimų priėmimo kontrolės funkciją ir atsakomybę už tų sprendimų pasekmes.

4. KT privalo išsiaiškinti ir suformuluoti tuos pagrindinius tikslus, kurių turi būti siekiama, kuriant bei tobulinant sistemą S (mūsų atveju – kuriant SIS), **o taip pat – tas pagrindines problemas, kurios turėtų būti išspręstos.** Po to KT privalo parinkti ir paskirti asmenis, atsakingus už tų problemų sprendimą ir suformuluotų tikslų pasiekimą – būsimųjų **probleminių komisijų vadovus.** Tie vadovai turi būti parenkami iš KT narių, o jei atitinkamos problemos sprendimo organizavimui reikalingas toks specialistas, kokio kol kas nėra KT narių tarpe, tai toks specialistas surandamas ir įvedamas į KT sudėtį.

5. Paskirtieji probleminių komisijų (PK) vadovai nagrinėja galimus problemų sprendimo variantus, pasirenka tinkamiausius ir sudaro problemų sprendimo tikslų medžius, problemų sprendimo technines užduotis (TU) ir probleminių komisijų darbo planų projektus (PKDPP). Savo sprendimus bei sudarytus dokumentus (TU ir PKDPP) PK

vadovai aptaria KT posėdžiuose ir, atsižvelgdami į išsakytas pastabas, koreguoja juos, o po to tvirtina Koordinacinėje taryboje.

6. Gavę KT sutikimą, **PK vadovai formuoja problemines komisijas**, parinkdami į jas problemų sprendimui reikalingus specialistus iš organizacijos S darbuotojų arba (jei sistemoje S tokių specialistų nėra) iš kitų organizacijų.

7. Sudarytų PK nariai kartu su savo vadovais aptaria, patikslina ir detalizuoja problemų sprendimo TU ir PK darbo planus, kuriuose numato, kokios **probleminės darbo grupės (PDG)** turėtų būti organizuotos, sudaro TU kiekvienai tokiai PDG ir pateikia visus tuos dokumentus KT svarstymui ir tvirtinimui.

8. PK, gavusios KT sutikimą, **organizuoja pastoviai veikiančias PDG**, paskirdamos jų vadovus iš PK narių tarpo ir pavesdamos jiems sudaryti PDG darbo (organizacijos S bei jos veiklos analizės, SIS posistemių projektavimo, realizavimo ir diegimo) planus (DP).

9. PDG, patikslinusios ir detalizavusios savo TU ir sudariusios savo DP projektus, pateikia tuos dokumentus savo PK – svarstymui ir tvirtinimui. Pasitvirtinusios šiuos dokumentus, PDG pradeda **organizacijos S analizės ir jos tobulinimo projektavimo darbus**.

10. Probleminei darbo grupei pradėjus darbą, **KT sudaro sau tiesiogiai pavaldžią sisteminę darbo grupę (SDG)**, parinkdama į ją sisteminės analizės, sisteminio modeliavimo bei sisteminio projektavimo specialistus bei dženeralistus ir pavesdama tai grupei (ir jos vadovui – KT nariui) PDG ruošiamų projektų derinimą, įvairių PDG, priklausančių skirtingoms PK, einamojo darbo koordinavimą, derinant su atitinkamomis PK iškilančius klausimus bei pateikiant juos KT svarstymui, kai jų nepavyksta suderinti PDG ir PK lygiuose.

11. **PDG savo užduočių vykdymo rezultatus** (analizės ataskaitas, projektus, realizavimo bei diegimo planus ir t.t.), aptarusios juos savo probleminėse komisijose ir gavusios jų pritarimą, **pateikia sistemei darbo grupei – derinimui ir jungimui į bendrą visos sistemos projektą**.

12. SDG, gavusi projekcinę medžiagą iš įvairių PDG, **jungia ją į bendrą organizacijos S ir jos veiklos tobulinimo projektą**, aptaria to projekto atitinkamus fragmentus su atitinkamomis PDG ir PK, o po to pateikia KT – svarstymui ir tvirtinimui. Tai daroma kiekviename chronologiniame organizacijos S tobulinimo (mūsų atveju – SIS kūrimo) etape.

13. Kiekvieno chronologinio etapo gale PDG atsiskaito už savo darbą ir jos rezultatus savo PK, o kiekviena PK ir SDG – Koordinacinei tarybai. Tuo pačiu metu kiekviena PDG pateikia savo PK ir tvirtina joje savo darbo plano projektą sekančiam chronologiam etapui, o visos PK ir SDG pateikia ir tvirtina sekančio etapo darbo planus Koordinacinėje taryboje.

14. PDG, atsiradus reikalui ir atitinkamai PK leidus, **gali organizuoti savo sudėtyje programuotojų, operatorių, technikų bei kitas grupes (PG, OG, TG ir kt.)**. SDG pageidavimu gali būti organizuojamos ir **bendros PG, OG ir TG, aptarnaujančios kelias PDG bei pačią SDG**.

15. Nauji sistemos R padaliniai gali ir turi atsirasti tik tada, kai jiems susiformuoja darbų frontas ir sudaromos atitinkamos TU. **Neleistina organizuoti kokius nors naujus R padalinius, neaprūpinant jų darbu, neleistina priimti į kokią nors darbo grupę naują darbuotoją, neužtikrinant, kad jis tuoj pat gaus konkrečią ir aiškią užduotį ir nuo pirmosios savo darbo dienos bus pilnai apkrautas**. Kitaip tariant, sistema R kuriasi ir plečiasi palaipsniui, didėdama didėdant darbo apimtims.

16. Sistemos R struktūriniai padaliniai – KT, PK, PDG, SDG, PG, OG ir TG – priklauso **įvairiems sistemos R struktūros lygiams** (žiūr. 11 pav.) ir, atsižvelgiant į vykdomų funkcijų charakterį, gali būti priskirti vienam iš dviejų tipų: **vadovavimo ir vykdymo**.

KT ir PK yra **problemas keliantieji ir vadovaujantys organai, dirbantys epizodiškai**, t.y. susirenkantys tik į iš anksto planuotus, o retkarčiais – ir neplanuotus KT ir

PK posėdžius. Visą kitą, laiką KT ir PK nariai, išskyrus tuos, kurie tapo PDG ir SDG vadovais, dirba reguliarių savo darbą sistemoje S (ar kokioje nors kitoje organizacijoje, jei specialistas buvo pakviestas į KT bei PK iš šalies). KT ir PK narių darbas (išskyrus PDG bei SDG vadovus ir specialistus, pakviestus iš šalies) **nėra apmokamas iš sistemai R skirtų lėšų**. Tačiau KT ir PK nariai iš tų lėšų gali būti premijuojami už pasiektus ypatingai gerus rezultatus, t.y. už reikšmingą indėlį, tobulinant organizaciją S bei kuriant ir diegiant jos naujos būsenos projektą.

PDG, SDG, PG, OG ir TG yra **sistemos R darbo organai**. Jų darbuotojai sistemoje R dirba pastoviai (būdami laisvi nuo kitų darbų) visą darbo dieną arba po pusę darbo dienos kasdien (arba 2–3 dienas per savaitę), gaudami **už tą darbą atitinkamą atlyginimą iš sistemai R skirtų lėšų**. Jie taip pat gali būti iš tų lėšų premijuojami, PK ar KT nariams pasiūlius ir KT priėmus atitinkamą nutarimą.

17. Sistemos R darbo pagrindiniu rezultatu turi tapti toks organizacijos S ir jos veiklos patobulinimas, kad:

- a) **žymiai padidėtų sistemos S funkcionavimo efektyvumas, sumažėtų klaidingų sprendimų ir kitokių klaidų bei trūkumų skaičius;**
- b) **organizacijos S sumažėjusios išlaidos (ar padidėjęs jos pelnas) pilnai padengtų sistemos R sukūrimo ir išlaikymo išlaidas, t.y., kad sistema R (imant pakankamai ilgą laiko intervalą, pavyzdžiui, 5–6 metus) nebūtų nuostolinga.**

18. Įvykdžiusi savo užduotį, t.y. pervedusi organizaciją S į suprojektuotą naują būseną, sistema R nebūtinai turi būti likviduojama. Ji gali būti **transformuojama į pastoviai veikiančią organizacijos S ir jos įrangos aptarnavimo ir tobulinimo padalinį**. Žinoma, sumažėjus sistemos R darbo apimtims, atitinkamai privalo sumažėti ir jos apmokamų darbuotojų skaičius darbo grupėse. Todėl būtina iš anksto pasirūpinti grįžtančių iš sistemos R į sistemą S buvusių sistemos S darbuotojų įdarbinimu.

19. Sistemų S ir R vadovai visą laiką turi turėti omenyje, kad **sistemai R pastoviai gresia pavojus išsigimti į parazituojantį sistemos S anstatą, neduodanti pakankamai naudos sistemai S**.

Todėl visą laiką turi būti apskaičiuojamas (ar kaip nors kitaip įvertinamas) galutinis efektas, gaunamas išsprendus kokią nors problemą, bei visos išlaidos, kurias tenka padaryti, sprendžiant tą problemą, ir neturi būti leidžiama toms išlaidoms per daug priartėti prie gaunamo efekto (o tuo labiau – viršyti jį).

20. Šis reikalavimas turi galioti ne tik sistemai R, sukurtai sistemos S tobulinimui, bet ir visai organizacijos S valdymo sistemai: **valdymo sistema gali būti plečiama tik tada, kai to plėtimo išlaidos yra mažesnės negu tas papildomas efektas, kuri gauna organizaciją S dėl jos valdymo sistemos išplėtimo bei patobulinimo**. Kadangi sistema R, tobulindama organizacijos S funkcionavimą, gali ir privalo tobulinti organizacijos S valdymo sistemą, tai ji turi būtinai atsižvelgti į šį reikalavimą, projektuodama naują organizacijos S būseną.

Visa tai rodo, kad tarp sistemos R ir organizacijos S valdymo sistemos turi būti palaikomas labai tamprus abipusis ryšys. Geriausiai tai pasiseka padaryti, kai **R sistema tampa organizacijos S valdymo sistemos padaliniu, tiesiogiai pavaldžiu organizacijos S vadovui, o pagrindiniu organizacijos S valdymo sistemos aukščiausiojo ešelono tikslu tampa organizacijos S tobulinimas**.

Iš ką tik išsakytų principų išplaukia sistemos R struktūra, kurią vaizduoja 11 pav. Ten parodyti sistemos R lygiai (ešelonai), jiems priklausantys jos padaliniai ir ryšiai tarp jų. Pavaldumo ryšiai vaizduojami rodyklėmis $\longrightarrow$, einančiomis iš valdymo subjekto link valdymo objekto, informaciniai ryšiai – rodyklėmis $\cdots\cdots\longrightarrow$ ir $\longleftarrow\cdots\cdots$, metodiniai nurodymai – rodyklėmis $-----\longrightarrow$.

11 pav. pavaizduotą bloką PK sudaro kelios PK: $PK_1, PK_2, \dots, PK_i, \dots, PK_n$, o kiekviena PK_i gali organizuoti po kelias jos TU vykdančias PDG: $PDG_{i,1}, PDG_{i,2}, \dots, PDG_{i,m_i}$. Todėl PDG bloke gali būti gana daug PDG: $PDG_{1,1}, PDG_{1,2}, \dots, PDG_{i,m_i}, PDG_{2,1},$

..., PDG_{2,m₂}, PDG_{3,1}, PDG_{3,2}, ..., PDG_{3,m₃}, ..., PDG_{i,1}, ..., PDG_{i,m_i}, ..., PDG_{n,1}, ..., PDG_{n,m_n}.

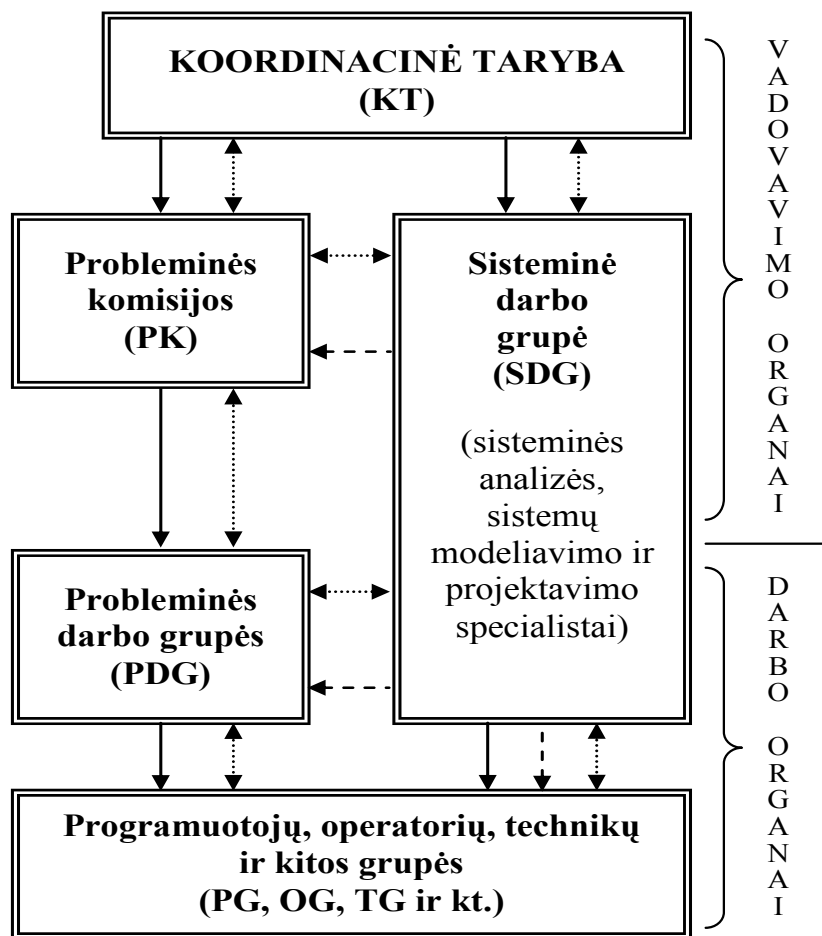
SISTEMOS R STRUKTŪRA

Tikslų bei problemų formulavimo ir sistemos R veiklos koordinavimo lygis

Projektavimo TU sudarymo ir problemų sprendimo procesų vadovavimo lygis

Problemų sprendimo (projektavimo ir projektų diegimo) lygis

Programavimo ir procedūrų vykdymo lygis



11 pav.

Kai kurie PDG darbuotojai patys gali būti kartu ir programuotojais bei operatoriais, tačiau neretai būna tikslinga išskirti ir bendras autonomines programuotojų, operatorių ir technikų grupes, aptarnaujančias visas PDG, o taip pat ir SDG. Tas grupes 11 pav. vaizduoja atskiras ketvirtojo lygio blokas.

Norint, kad sistema R funkcionuotų pakankamai efektyviai, reikia, kad visos tos sistemos funkcijos būtų racionaliai ir vienareikšmiškai paskirstytos 11 pav. pavaizduotiems jos padaliniams, o padaliniuose – atskiriems darbuotojams. **Kiekvienas padalinys ir kiekvienas darbuotojas turi aiškiai žinoti tuos tikslus, kurių jis turi siekti, tas užduotis, kurias jis turi įvykdyti, tuos rezultatus, kuriuos jis turi gauti ir už kuriuos turi atsiskaityti. Turi būti aiškiai apibrėžti visi pavaldumo ir bendradarbiavimo santykiai. Visa tai turi atsispindėti padalinių ir jų darbuotojų darbo planuose bei jų darbą ir santykius reglamentuojančioje normatyvinėje organizacinėje dokumentacijoje.**

R sistemos struktūrinėje schemoje (11 pav.) pavaizduotų struktūrinių padalinių normatyvinių nuostatų pavyzdžiai pridėti prie šios metodinės priemonės kaip jos priedai (žiūr. priedus Nr. 6, Nr. 7, Nr. 8, Nr. 9, Nr. 10). Tokie nuostatai jau patikrinti praktinėje veikloje ir gali būti naudingi, sudarant SIS kūrimo grupių struktūrinę NOD ir organizuojant tas grupeis bei jų darbą.

Labai svarbu pačioje S sistemos tobulinimo ir R sistemos kūrimo pradžioje išskirti visas spręstinas problemas, pakankamai tiksliai jas suformuluoti ir po to tinkamai parinkti

specialistus bei dženeralistus, sugebančius jas spręsti ir išspręsti: sisteminius analitikus ir sisteminius projektuotojus, vadybininkus, statistinės sistemos darbo ir jos specifikos žinovus, duomenų bankų ir programinės bei techninės įrangos projektavimo bei eksploatavimo specialistus. Ypatingas dėmesys turi būti atkreiptas į PK, užsiimančios SD organizacinės struktūros tobulinimo problemomis, ir tai PK pavaldžių PDG komplektavimą bei į SDG sudarymą kurioje darbuotojai, būdami kokios nors srities specialistais, turėtų būti dar ir dženeralistais, neblogai žinančiais visą statistinę sistemą.

Lygindami mūsų siūlomą sistemos R sudėtį ir struktūrą su tuo, ką rekomenduoja CASE metodo autoriai, aprašydami automatizuotos IS kūrimo darbų organizavimą, randame daug panašumų, o kartu – ir esminių skirtumų.

Pavyzdžiui, CASE autoriai vartoja terminą „Vartotojų vadovybė“, apibrėždami vartotojų vadovybę kaip iš organizacijos ir jos padalinių vadovų sudarytą vadovaujantį komitetą, kontroliuojantį IS kūrimą ir sprendžiantį, ar siūlomas IS projektas yra tinkamas organizacijai, ar kuriama sistema bus efektyvi ir patogi vartotojui, ar priimtina AIS kūrimo darbų savikaina ir t.t. Akivaizdu, kad „Vartotojų vadovybės“ samprata labai artima mūsų siūlomos „Koordinacinės tarybos“ sampratai ir skiriasi tik tuo, kad **Koordinacinės tarybos funkcijos yra žymiai platesnės ir jos vaidmuo SIS kūrimo procese yra žymiai aktyvesnis.**

Tai ir suprantama: CASE metodo kūrėjai aprašė IS kūrimą organizacijoje su ilgos evoliucijos gerai nušlifauta ir nusistovėjusia struktūra, kurioje, kuriant AIS, ne taip jau daug ką reikia keisti ir todėl galima apsiriboti tik informacinių procesų automatizavimo darbais, siekiant padidinti organizacijos efektyvumą ir palengvinti jos personalo darbą, o mūsų atveju tenka kurti SI pereinamojo laikotarpio sąlygomis, kai didelė dalis ekonominių ir valstybės valdymo struktūrų kardinaliai keičiasi, kai reikia pereiti nuo sovietinės statistinės sistemos prie sistemos, atitinkančios naujos santvarkos ir Eurostato reikalavimus, ir todėl jau nebegalima apsiriboti vien tik informacinių procesų automatizavimu. Todėl mūsų sąlygomis žymiai padidėja ir suaktyvėja vartotojų ir jų vadovybės vaidmuo, nes tenka spręsti problemas, išsėinančias už grynai informacinių sistemų automatizavimo specialistų kompetencijos ribų. Labai gaila, kad **Statistikos departamente dar ne visi pakankamai giliai supranta ši esminį skirtumą tarp problemų, kurias tenka spręsti, kuriant AIS organizacijoje su nusistovėjusia aplinka ir nusistovėjusia struktūra, ir kuriant SIS mūsų sąlygomis, kai reikia kardinaliai reorganizuoti valstybės valdymą ir jo institucijas, tame tarpe ir valstybės statistikos sistemą, prisitaikant prie „ėjimo į Europą“ reikalavimų.** Juk būtų labai neracionalu įdėti daug darbo ir lėšų į automatizavimą tokių informacinių procesų, kuriuos tuoj pat tektų keisti, išmetant „už borto“ ką tik sukurtą sudėtingą ir brangią SIS programinę įrangą. Žymiai protingiau būtų, kuriant SIS, iškart rekonstruoti visą valstybinės statistikos sistemą, pritaikant ją prie naujų reikalavimų ir „įmontuojant“ SIS į VADIS ir į LVII kaip sudėtinę jų dalį.

Todėl ir atsiranda papildomos KT funkcijos, kurių nėra CASE autorių aprašytos vartotojų vadovybės funkcijų sąrašė. Todėl ir išsiplečia laiduojančio (atsakingo) vartotojo, atitinkančio KT pirmininko pareigas, funkcijos ir atsakomybė, ir todėl mes rekomenduojame KT pirmininku ne „vieną iš organizacijos vadovybės darbuotojų“, o būtent patį visos organizacijos vadovą.

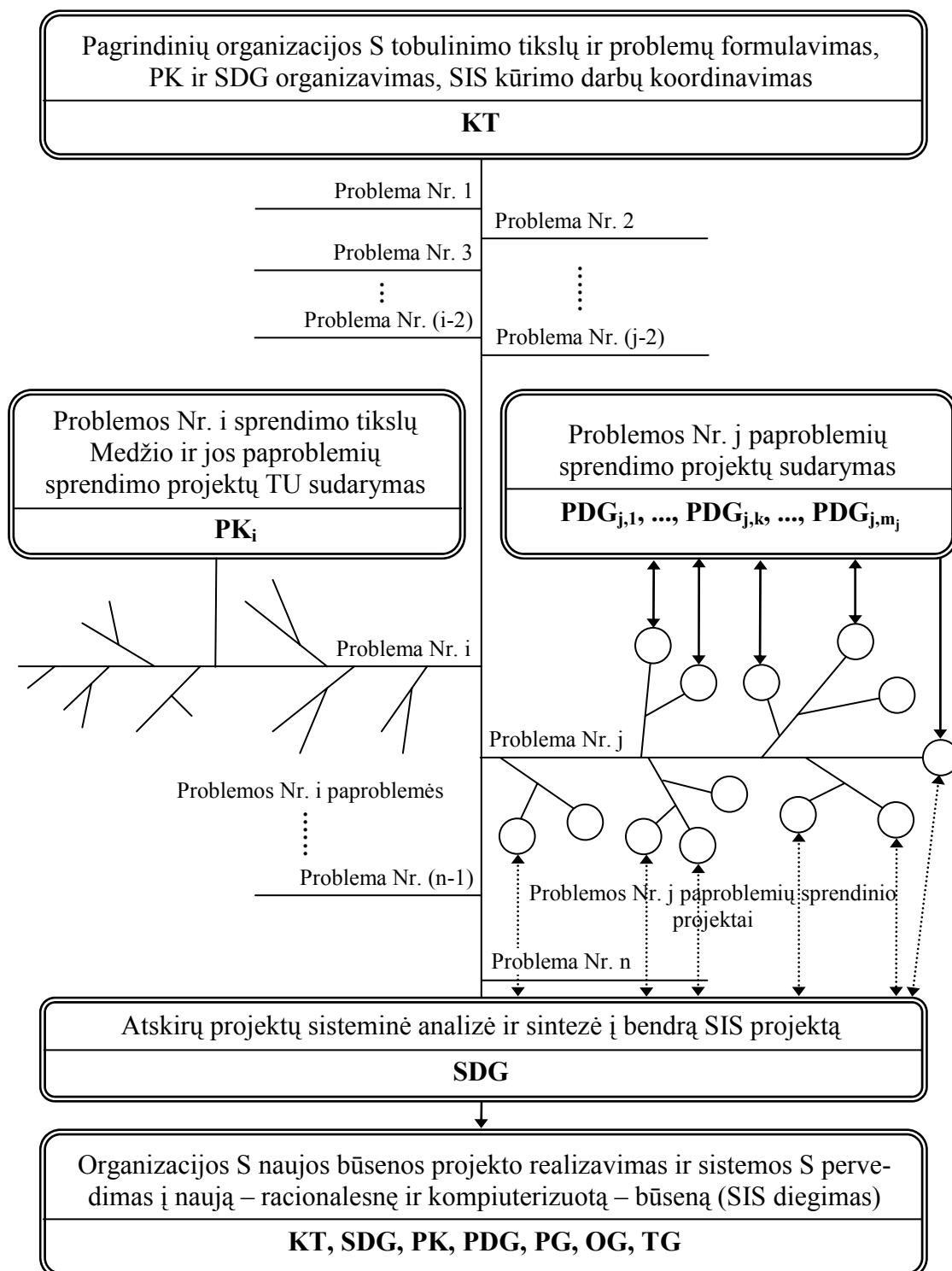
Kadangi vienai nedidelei KT išspręsti per trumpą laiką tiek daug ir tokių sudėtingų problemų tiesiog neįmanoma, o „išpūsti“ KT sudėtį netikslinga, nes tuomet jos darbas taps neefektyvus, todėl mes ir siūlome organizuoti problemines komisijas (PK), neturinčias analogo CASE metodo autorių siūlomoje organizacinėje IS kūrimo struktūroje.

Išsiplėtus sprendžiamų problemų sąrašui, išsiplečia ir probleminių darbo grupių (PDG) sąrašas. Jau nebepakanka vien tik informacinių procesų automatizavimo specialistų (sisteminių analitikų, dizainerių, duomenų administratorių, programuotojų, operatorių, informacinę techniką instaliuojančio bei palaikančio personalo) ir jiems vadovaujančių dženeralistų – projektų menedžerių. Reikia ir specialistų, sugebančių spręsti valstybinės statistinės sistemos

modernizavimo, jos naujos struktūros ir naujos darbo technologijos bei valdymo sistemos projektavimo ir diegimo problemas, o viso to sudėtingo darbo operatyviniam vadovavimui projekto vadovui į pagalbą tenka organizuoti sisteminę darbo grupę (SDG), per jos vadovą – KT narį – tiesiogiai pavaldžią Koordinacinei tarybai.

Sistemos R padalinių vietą ir vaidmenį valstybinės statistinės sistemos tobulinimo ir SIS kūrimo problemų sprendimo procese vaizduoja 12 pav.

SISTEMOS R PADALINIŲ VIETA IR VAIDMUO ORGANIZACIJOS S TOBULINIMO PROBLEMŲ SPRENDIMO PROCESSE, VAIZDUOJAMAME TIKSLŲ MEDŽIO MODELIU



12 pav.

Kadangi sistemą R sudaro gana daug įvairių padalinių, lygiagrečiai vykdančių susijusius sudėtingus darbus, tai iškyla gana nelengva visų tų darbų derinimo problema, kurią turi spręsti KT ir SDG. Sprendžiant tą problemą, kaip jau buvo minėta šios priemonės 5 psl., labai gali padėti sustambinta bendrojo darbų tinklinio grafiko (TG) ir atskirų grupių darbų detalesnių TG sudarymas. Pirmajame IS kūrimo etape nesistengiama sudaryti labai detalių bendrąjį TG visiems etapams, nes toks TG nebūtų gyvybingas: keičiantis aplinkybėms, jis gana greit „subyrėtų“. Sudarius sustambintą bendrąjį TG (vieno tokio TG pavyzdys duotas 11–ajame priede) visiems etapams, pirmojo etapo darbus jau galima detalizuoti, nurodant terminus savačių ir dienų tikslumu. Pirmojo etapo pabaigoje, koreguojant darbų planus, galima koreguoti ir TG, detalizuojant antrojo etapo darbus. Panašus koregavimas ir trečiojo etapo darbų TG detalizavimas turi būti daromas antrojo etapo pabaigoje ir t.t. iki paskutiniojo SIS kūrimo etapo.

Norėčiau atkreipti skaitytojo dėmesį į tai, kad TG – ne tik efektyvus ir patogus darbų planavimo ir jų vykdymo terminų derinimo, bet ir tų darbų eigos operatyvaus monitoringo, jų įvykdymo kontrolės instrumentas. Todėl SIS projekto vadovams (ypatingai KT ir SDG) derėtų nepagailėti jėgų bendrųjų TG, o PK ir PDG vadovams – PK ir PDG darbų TG sudarymui.

Konstruojant SIS kūrimo TG, tikslinga pasinaudoti šios metodinės priemonės vienuoliktajame skyrelyje aprašytų CASE metodo taikymo etapų užduočių sąrašais, o taip pat ankstesniuose priemonės skyreliuose išdėstyta sudėtingų organizacinių sistemų kūrimo sistetine metodologija, atkreipiant ypatingą dėmesį į Lietuvos statistinės sistemos (Statistikos departamento ir jo padalinių apskrityse ir rajonuose) vadovų ir darbuotojų įtraukimą į SIS kūrimo darbus.

Sudarant TG, nepakanka žinoti vykdytinų darbų sąrašus. Reikia išsiaiškinti loginius ryšius tarp darbų ir galimas darbų trukmes. Darbams, kurie susideda iš standartinių veiksmų, gali būti apskaičiuotos normatyvinės trukmės, o tiems darbams, kurių analogai būdavo gana dažnai vykdomi anksčiau, gali būti nustatytos statistinės vidutinės trukmės. Nemažai naudingos informacijos apie IS kūrimo darbų trukmes galima rasti įmonėje „Informacinės technologijos“, jos darbuotojų ataskaitose apie įvykdytus IS kūrimo darbus įvairiose organizacijose. Nustatant darbų trukmes, galima naudotis ir ekspertų įverčiais: optimistinėmis (galimomis trumpiausiomis), pesimistinėmis ir tikimiausiomis darbų trukmėmis. Tokių ekspertinių įverčių apdorojimui galima pasinaudoti plačiai paplitusiu PERT metodu ir jo programine įranga. Ji leidžia ne tik apskaičiuoti vidutines darbų trukmes, bet ir šiek tiek pagerinti TG charakteristikas: padidinti darbų įvykdymo plane numatytų laiku tikimybes, sutrumpinti TG kritinio kelio ilgį. Norint sumažinti darbų įvykdymo bendrąsias išlaidas (arba, turint duotą išlaidų sumą, sutrumpinti kritinio kelio ilgį), galima naudotis ir taip vadinamu kritinio kelio metodu (CPM) bei jo programine įranga.

TG sudarymą ir jų panaudojimą operatyviam IS kūrimo darbų eigos valdymui galima ir tikslinga automatizuoti. Naudojantis tokiais automatizuotais TG bei jų sudarymo ir apdorojimo automatizuotomis sistemomis, galima gauti: reglamentuotame režime (kiekvieno chronologinio etapo pabaigoje) – spausdintas ataskaitas apie atliktus darbus ir naujas (labiau detalizuotas) TG redakcijas sekančiam etapui; signaliniame režime – signalinius pranešimus apie darbus, atsiliekančius nuo plane numatytų vykdymo terminų; užklašos režime – informaciją apie užklašėją dominančių darbų eigą, jų rezultatus. Planinę informaciją apie darbų tinklinius grafikus ir ataskaitinę informaciją apie darbų eigą ir rezultatus tikslinga saugoti CASE duomenų saugykloje (Dictionary, Repository), o tinklinių grafikų panaudojimo darbų planavime ir jų eigos operatyvinėje kontrolėje programinę įrangą būtų tikslinga įtraukti į SIS kūrimo automatizavimo priemonių rinkinį kaip Repository Administrator programinių priemonių papildymą.

Dešimtajame šios priemonės skyrelyje jau buvo užsiminta apie SIS ir VADIS kūrimo darbų derinimo būtinumą. Tai būtų galima padaryti, suderinant SIS ir VADIS kūrimo tinklinius grafikus. Jei VADIS projektuotojai dar neišsiaiškino visų savo darbų ir dar

nepasiruošę VADIS kūrimo darbų TG sudarymui bei įsikomponavimui į LVII, galima pasiūlyti jiems kartu su SIS projekto kūrėjais aptarti vieną galimų bendro TG variantų, kuri būtų galima sukonstruoti, pasinaudojant idėjomis, išdėstytomis vienuoliktajame šios priemonės priede „Valstybės valdymo sistemos Vyriausybės lygyje tobulinimo kryptys ir darbų TG projektas“. Toks iniciatyvos parodymas iš SIS projektuotojų pusės ne tik padėtų VADIS projektuotojams, bet būtų naudingas ir SIS kūrėjams, nes leistų sudaryti tokį VADIS kūrimo planą, kuriame būtų atsižvelgta ir į SIS poreikius bei galimybes.

13. KAIP NAUDOTIS ŠIA METODINE PRIEMONE

Norint pakankamai gerai įsisavinti sisteminę sudėtingų organizacijų projektavimo ir tobulinime metodologiją, šią metodinę priemonę patartina perskaityti mažiausiai du kartus.

Pirmasis skaitymas turėtų būti apžvalginio pobūdžio ir pakankamai greitas, kad skaitytojas, skaitydamas vėlesnius priemonės skyrelius, dar nebūtų suspėjęs pamiršti ankstesniųjų skyrelių pagrindinių idėjų. Skaitant pirmąjį kartą, nebūtina stengtis įsiminti visus terminus, nes, atsiradus reikiamybei, visų svarbesniųjų naujų terminų paaiškinimus skaitytojas galės rasti specialiųjų terminų žodynyje (Glossary) šios priemonės keturioliktajame skyrelyje. Pakanka, kad skaitytojas, perskaitęs visus dvylika pirmųjų šios priemonės skyrelių ir peržvelgęs jos priedus, susidarytų bendrą vaizdą apie sisteminę metodologiją bei jos taikymą ir susipažintų su orgprojektavimo automatizavimo idėjomis ir jo priemonių galimybėmis, su sisteminio orgprojektavimo loginiais ir chronologiniais etapais, su svarbiausiais sistemų teorijos ir sisteminio projektavimo terminais, suvoktų jų prasmę.

Pirmojo skaitymo pabaigoje būtų labai naudinga šiek tiek pastudijuoti keturioliktąjį skyrelį – kai kurių specialiųjų terminų žodynėlį – patikrinant, ar skaitytojui jau pakankamai aiškios pagrindinės specialios sąvokos, vartojamos šioje priemonėje.

Po pirmojo šios priemonės skaitymo būtų naudinga perskaityti septynioliktajame jos puslapyje paminėtąjį šių eilučių autoriaus knygutę apie antrosios kartos AVS principus, parašytą lietuvių kalba.

Po viso to galima pereiti prie antrojo metodinės priemonės ir jos priedų skaitymo – pačios priemonės ir jos priedų studijavimo. Šis skaitymas turėtų būti jau lėtesnis su neretais grįžimais prie jau anksčiau skaitytų skyrelių. Kai kurios svarbesnės skaitytojui dar neįprastos mintys galėtų būti net konspektuojamos tuo tikslu, kad, ką nors užsirašant, geriau įsigilinama į tekstą ir jo prasmė stipriau užsifiksuoja žmogaus sąmonėje ir jo atmintyje. Patartina atidžiau pastudijuoti ir šios priemonės priedus, iliustruojančius pagrindiniame tekste išdėstytų idėjų ir metodų praktinį pritaikymą.

Antrojo skaitymo pabaigoje patartina dar kartą peržvelgti žodynėlį ir įsitikinti, kad visos pagrindinės sąvokos yra pakankamai įsisavintos.

KT ir PK nariams šių dviejų perskaitymų pilnai pakaktų. Vėliau, SIS kūrimo procese, galėtų būti tik epizodiškai kreipiamasi į kai kurias šios priemonės teksto vietas bei į kai kuriuos jos priedus.

PDG ir SDG vadovams būtų naudinga po antrojo šios priemonės skaitymo susipažinti su CASE priemonės aprašančia literatūra, paminėta vienuoliktajame priemonės skyrelyje, o po to kartu su savo grupių bei PG ir OG nariais kai kurias iš ten išvardintų knygų pastudijuoti arba padalyvauti praktiniuose seminaruose, organizuojamuose tų SIS projektuotojų, kurie jau sugeba praktiškai naudotis CASE programine įranga (pavyzdžiui, yra baigę atitinkamus mokymo kursus, organizuojamus įmonėje „Informacinės technologijos“).

Būtų galima organizuoti specialius kursus sistemų R ir S bei jų padalinių vadovams, supažindinant juos su sisteminę SIS kūrimo metodologija. Tai palengvintų jiems ir pagreitintų šios metodinės priemonės ir jos priedų skaitymą bei studijavimą. Vadovai savo ruožtu galėtų organizuoti panašius kursus – seminarus savo pavaldiniams.

Sisteminės SIS metodologijos ir jos taikymo priemonių įsisavinimas žymiai pagreitintų SIS kūrimo procesą ir pagerintų SIS projekto kokybę.

Visais sisteminės metodologijos klausimais galima konsultuotis pas šios metodinės priemonės autorių, iš anksto susitarus telefonu 213 63 90 darbo metu ar namų telefonu 276 98 36 vakarais.

Baigdamas šią metodinę priemonę, norėčiau pasinaudoti proga ir nuoširdžiai padėkoti UAB „Statistikos departamento skaičiavimo centras“ vyr. specialistei p. Ritai Viselgaitei už kruopštų darbą, maketuojant ir spausdinant visą priemonės tekstą ir brėžinius bei priedus. Išreiškiu padėką ir Kompiuterinių sistemų plėtojimo skyriaus viršininkui p. E.Žvirbliui, inicijavusiam ir organizavusiam metodinės priemonės paruošimą, o taip pat bendros Lietuvos–Austrijos įmonės „Informacinės technologijos“ bendradarbiams, buvusiems mano studentams, leidusiems pasinaudoti įmonėje sukaupta medžiaga apie AIS kūrimą naudojantis CASE metodu ir priemonėmis (ataskaitomis, vartotojų vadovais ir literatūra apie CASE metodą ir priemones).

Autorius

1996 m. Vilnius,
Statistikos Departamentas