

Vilniaus universitetas
Tarptautinis žinių ekonomikos ir žinių vadybos centras

Vera Afanasjeva,
Informacijos sistemų vadybos studijų programos studentė

PROGRAMINĖS ĮRANGOS KŪRIMO PROJEKTŲ
KOKYBĖS VADYBA

MAGISTRO DARBAS

Vadovas doc. dr. R. Gudauskas

Vilnius, 2007

_____ magistro darbas

(magistranto (-ės) vardas, pavardė)

tema

parengtas gynimui.

(data) (vadovo parašas)

Darbas įregistruotas _____ centre

(data) (administratorės parašas)

Magistro darbą ginti leidžiu

(data) (centro direktoriaus parašas)

Recenzentu skiriu

(data) (direktoriaus parašas)

Darbą recenzavimui gavau

(data) (recenzento parašas)

Afanasjeva, Vera

Af-07

Programinės įrangos kūrimo projektų kokybės vadyba: magistro darbas / Vera Afanasjeva; mokslinis vadovas doc. dr. R. Gudauskas; Vilniaus universitetas. Tarptautinis žinių ekonomikos ir žinių vadybos centras. – Vilnius, 2007. – 82 lap.: 3 lent. – Santr. angl. – Bibliogr.: p. 69-72 (60 pavad.).

UDK 004.4

Programinės įrangos kokybė, programinės įrangos kokybės charakteristikos, projektų vadyba, projektų vadybos žinių sritys, kokybės vadyba, kokybės valdymo metodai.

Magistro darbo objektas – aukštos kokybės programinės įrangos kūrimo projektų vykdymo ir užtikrinimo galimybių turinys. Darbo tikslas – kokybės vadybos metodų taikymo galimybių nustatymas programinės įrangos kūrimo projektams. Pagrindiniai darbo uždaviniai: išanalizuoti projektų vadybos, kokybės ir programinės įrangos kūrimo projektų sąvokas ir sampratas; aptarti projektų vadybos procesus ir metodus bei jų taikymą programinės įrangos kūrimo projektuose; apžvelgti programinės įrangos rinką ir jos tendencijas; nustatyti kokybiško programinės įrangos produkto charakteristikas, metrikas ir kokybės vertinimo parametrus; apibendrinti kokybės vadybos sistemų galimybes, kuriant programinės įrangos produktus.

Darbe remiamasi mokslinės literatūros, kokybės standartų, straipsnių ir kitų šaltinių, skirtų projektų kokybės vadybos ir programinės įrangos kokybės temai, analize ir kitų autorių atliktų programinės įrangos kokybės metodologijų tyrimų rezultatais.

Šis magistro darbas apibendrina teorinius kokybės vadybos metodų pagrindus ir nurodo programinės įrangos kuriančioms organizacijoms jų praktinio taikymo galimybes ir vertes.

TURINYS

ĮVADAS.....	5
1. PROJEKTŲ VADYBA.....	7
1.1. Projektų vadybos raida.....	7
1.2. Projektų vadybos samprata.....	9
1.3. Projektų vadybos procesai ir metodologijos.....	10
1.4. Projektų kokybės samprata.....	15
2. PROGRAMINĖS ĮRANGOS KŪRIMO PROJEKTAI.....	23
2.1. PĮ rinkos apžvalga.....	23
2.2. Programinės įrangos kūrimo projektų valdymo ypatumai.....	29
2.3. Programinės įrangos kūrimo procesai ir metodai.....	30
2.4. Programinės įrangos kokybė.....	35
2.5. Programinės įrangos kokybės įvertinimas.....	41
3. KOKYBĖS VADYBOS METODAI IR JŲ TAIKYMAS.....	45
3.1. Kokybės vadybos metodų apžvalga.....	45
3.2. Kokybės vadybos metodų taikymas programinės įrangos kūrimo projektams.....	53
IŠVADOS.....	67
BIBLIOGRAFINIŲ NUORODŲ SĄRAŠAS.....	69
PRIEDAI.....	73
1 priedas. Struktūrinių schemų pavyzdžiai.....	73
2 priedas. Kontrolinių diagramų pavyzdys.....	74
3 priedas. Pareto diagramos pavyzdys.....	75
4 priedas. Trendo analizės grafinio vaizdavimo pavyzdys.....	76
5 priedas. Vartotojų poreikių tenkinimo tyrimo modeliai.....	77
6 priedas. Krioklio modelis.....	78
7 priedas. Spiralinis modelis.....	79
8 priedas. Prototipinis modelis.....	80
9 priedas. Pagrindinės PBM brandos lygių proceso sritys.....	81
SUMMARY.....	82

IVADAS

Šiandien projektų vadyba visame pasaulyje tapo pripažinta projektinės veiklos metodologija. Naudodamos projektų vadybos metodus ir priemones, organizacijos gali pasiekti aukštą kokybės lygį, sutaupyti lėšų, laiko, sumažinti riziką ir padidinti kuriamų produktų ir teikiamų paslaugų patikimumą. Projektų vadybos reikšmė ypatingai svarbi programinės įrangos kūrimo projektuose. Netaikant projektų vadybos ar atskirų jos žinių sričių, pavyzdžiui, kokybės vadybos, programinės įrangos kūrimo projektų rezultatas dažnai vietoj naudos, nepaisant organizacijų patirties, atneša reikšmingas tiesiogines ir netiesiogines sąnaudas.

Daugelis organizacijų, tikėdamosi kokybiško produkto savo veiklos procesams pagerinti, į programinės įrangos kūrimą arba įsigijimą investuoja nemažai pinigų. Per pastarąjį dešimtmetį programinės įrangos industrija sparčiai išaugo (1996 m. informacinių technologijų rinkos vertė sudarė 120 milijardų dolerių, o 2006 m. ji siekė net 919 milijardų eurų). Kad organizacijos sukurtų kokybiškesnius programinės įrangos produktus, galinčius maksimaliai patenkinti vartotojų keliamus reikalavimus, didesnę dėmesį reikia skirti visam programinės įrangos kūrimo procesui, nes kokybė pradedama kurti projekto pradžioje ir turi būti planuojama viso projekto eigoje. Aukštą programinės įrangos kokybę pasiekti padeda projektų kokybės vadyba, kuri jungia tarpusavyje sąveikaujančius metodus, priemones ir technologijas.

Šio magistro darbo „Programinės įrangos kūrimo projektų kokybės vadyba“ **tikslas** – kokybės vadybos metodų taikymo galimybių nustatymas programinės įrangos kūrimo projektams. **Tyrimo objektas** – aukštos kokybės programinės įrangos kūrimo projektų vykdymo ir užtikrinimo galimybių turinys. Darbe remiamasi mokslinės literatūros, kokybės standartų, straipsnių ir kitų šaltinių, skirtų projektų kokybės vadybos ir programinės įrangos kokybės temai, analize ir kitų autorių atliktų programinės įrangos kokybės metodologijų tyrimų rezultatais.

Iškeltam darbo tikslui pasiekti keliami tokie **uždaviniai**:

- išanalizuoti projektų vadybos, kokybės ir programinės įrangos kūrimo projektų sąvokas ir sampratas;
- aptarti projektų vadybos procesus ir metodus bei jų taikymą programinės įrangos kūrimo projektuose;
- apžvelgti programinės įrangos rinką ir jos tendencijas;
- nustatyti kokybiško programinės įrangos produkto charakteristikas, metrikas ir kokybės vertinimo parametrus;

- apibendrinti kokybės vadybos sistemų galimybes, kuriant programinės įrangos produktus.

Šis magistro darbas apibendrina teorinius kokybės vadybos metodų pagrindus ir nurodo programinės įrangos kuriančioms organizacijoms jų praktinio taikymo galimybes ir vertes.

1. PROJEKTŲ VADYBA

Projektų vadyba visame pasaulyje tapo pripažinta projektinės orientacijos veiklos metodologija, daugelis žymesnių projektų ar programų realizuojami remiantis projektų vadybos metodologijos baze. Naudodamos projektų vadybos metodus ir priemones, organizacijos gali pasiekti gana aukštą kokybės lygį, sutaupyti lėšų, laiko, sumažinti riziką ir padidinti patikimumą. Projektų vadyba suteikia galimybę organizacijoms sėkmingai veikti besikeičiančiomis rinkos ekonomikos sąlygomis ir įgyvendinti savo tikslus.

Darbo organizavimą projektų pagrindu skatina tokie veiksniai [43, 55]:

- dėl išorinių sąlygų organizacija vienu metu nukreipia savo veiksmus į keletą skirtingų darbų, projektų apimtys ir jų darbai tampa neapibrėžtais ir sudėtingais, todėl dažnai atliekami ne tiesioginiai darbai, o apdorojama perteklinė informacija;
- materialių, finansinių ir darbo išteklių panaudojimo, projektų darbų ir rezultatų kokybės reikalavimų griežtėjimas;
- organizacijos noras pasiekti aukštą našumą ir ekonomiją, lanksčiai naudojant bendruosius ir ribinius išteklius;
- galimybių ir laisvos vietos rinkoje laikinumas, dėl kurio atsiranda laikinos gamybos organizavimo poreikis.

Formalus projektų vadybos metodų panaudojimas vykdant stambius ir sudėtingus projektus yra būtinas, kai organizacija nori laiku užbaigti projektą, laikantis specialių reikalavimų, kokybės lygio ir numatyto biudžeto.

1.1. Projektų vadybos raida

Projektų vadybos mokslo ištakos siekia 1940-1950 metus karo ir aviacijos pramonėje. Per pastaruosius penkiasdešimt metų išsivysčiusiose šalyse projektų vadybos augimas ir pripažinimas reikšmingai pasikeitė ir projektų vadybą suformavo kaip savarankišką mokslo sritį. Paprastai išskiriami keturi projektų vadybos raidos etapai.

1958 metai ir anksčiau. Projektų vadybos pradininku Jungtinėse Amerikos Valstijose laikomas *Henry Gantt*, žinomas dėl savo planavimo ir kontrolės metodo, panaudojant histogramas kaip projektų valdymo priemones. 1940 metais austrų biologas Ludwig von Bertalanff pasiūlė bendrąją sistemos teoriją (angl. *General Systems Theory*), tapusią bendrosios sistemų vadybos (angl. *General Systems Management*), bandančios apjungti ir integruoti skirtingų žinių sričių mokslinę informaciją, pagrindu [14]. Sistemų teorija sprendžia problemas, apžvelgiant visą bendrą problemos vaizdą, o ne analizuojant jos atskirus komponentus [31]. Bendrosios sistemų teorijos tikslas – sukurti tokį vadybos metodą, kuris

leistų vienu metu apjungti daugelį organizacinių disciplinų (pavyzdžiui, finansus, gamybą, projektavimą, marketingą ir kt.) ir išlaikyti vadybos funkcijas.

1958–1980 metai. Projektų vadybos plėtrą lėmė naujų metodų, leidžiančių sėkmingai vykdyti projektus, poreikis. Septintajame dešimtmetyje pradėta ieškoti naujų vadybos metodų ir organizacinių struktūrų, galinčių greitai prisitaikyti prie vykstančių pokyčių. Daugumos kompanijų veikla buvo pagrįsta neformaliu projektų valdymo metodu (projektai buvo valdomi neformaliai, kai projekto vadovo valdžia buvo minimali). 1958 metais buvo sukurtas keletas esminių projektų vadybos priemonių. Tai Programinės analizės ir patikrinimo (angl. *PERT – Program Evaluation and Review Technique*) ir Kritinio kelio (angl. *CPM – Critical Path Method*) metodai [5]. Vėliau PERT metodas buvo papildytas darbų išskaidymo struktūra (angl. *WBS – Work Breakdown Structure*). 70-aisiais metais buvo sukurta kita projektų vadybos priemonė – Medžiagų poreikių planavimas (angl. *Material Requirements Planning*). Jau aštuntajame dešimtmetyje kompanijos pradeda reorganizuotis, remiantis formalizuotu projektų vadybos procesu (pirmiausia dėl jų veiklos dydžio augimo ir sudėtingumo). Literatūroje atsiranda vis daugiau mokslinių darbų, skirtų projektų vadybos klausimams.

1980–1995 metai. Devinto dešimtmečio pradžioje kompanijos pradeda suprasti projektų vadybos naudojimo būtinumą, o pagrindiniu klausimu tampa ne kaip pritaikyti projektų vadybą savo veikloje, o kaip galima greičiau tai padaryti [31]. Devintasis ir dešimtasias XX a. dešimtmečiai pasižymi revoliuciniu informacijos vadybos sektoriaus vystymusi, kuriam didelę įtaką turėjo personalinio kompiuterio ir atitinkamų komunikacijos priemonių sukūrimas. Asmeninis kompiuteris leido žymiai sumažinti projektų vadybos kaštus ir padidinti sudėtingų projektų valdymo ir kontrolės efektyvumą. Projektų vadybos metodai tapo prieinami daugeliui organizacijų. Projektų vadyboje pradedamos naudoti pagalbinės priemonės [31]:

- visuotinės kokybės vadyba;
- lygiagrečioji gamyba;
- sprendimų priėmimo decentralizavimas;
- projektų kaštų kontrolės sistema;
- projekto apimties pokyčių kontrolė;
- rizikos valdymas.

1995 – iki šiol. 1995 metais projektų vadyba tapo įrankiu, padedančiu kompanijoms išgyventi greitai besikeičiančioje rinkoje. Jei ankstesniuose etapuose projektų vadyba turėjo tam tikrą pasipriešinimą, tai dabar ji tapo nepakeičiamu vidiniu kompanijų valdymo įrankiu. Šis periodas taip pat pasižymi Interneto sukūrimu, kuris turėjo didžiulę įtaką verslui ir atskiroms jo veikloms. Įdiegus Internetą, kompanijos tapo labiau orientuotos į vartotoją. Be to, dauguma šiuolaikinių projektų vadybos programinės įrangos turi prisijungimo prie Interneto savybę, leidžiančią automatiškai perduoti

informaciją apie vykdomų projektų ir jų atskirų darbų būseną, greitai informuoti visus projekto dalyvius apie projekto pokyčius, vėlavimus ir kt.

Nors projektų vadyba egzistuoja jau daugiau nei penkiasdešimt metų, iki šiol yra įvairių požiūrių ir neteisingų supratimų apie tai, kas iš tiesų yra projektų vadyba.

1.2. Projektų vadybos samprata

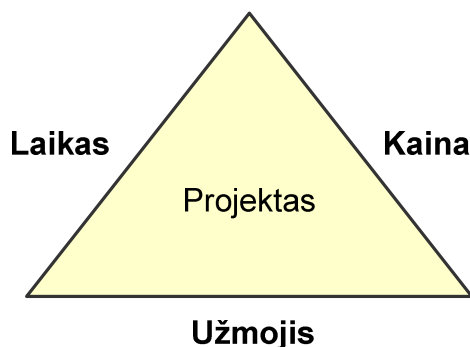
Šiuolaikinėje literatūroje galima rasti daug projektų vadybos apibrėžimų. JAV Projektų vadybos institutas projektų vadybą apibrėžia kaip žinių, įgūdžių, priemonių ir technologijų visumą, kurios pagalba įgyvendinamas projektas atitinka visus projekto reikalavimus ir patenkina visų projekto dalyvių poreikius [1]. Projektų vadybą galima laikyti kaip formalią vadybos discipliną, pagal kurią projektas yra planuojamas ir vykdomas kaip sistemingas ir besikeičiantis procesas. Projektų vadybą taip pat apibrėžiama kaip darbo, reikalingo pasiekti projekto viziją, vadovavimas ir kontrolė.

Projekto įgyvendinimas nuo nuolatinio darbo skiriasi tuo, kad projektas yra laikina ir unikali veikla, o nuolatinis darbas pasižymi nuolatine ir pasikartojančia veikla. Todėl dauguma autorių projektą apibūdina kaip laikinas pastangas, kurių imtasi unikalaus produkto ar paslaugos sukūrimui [1]. Laikinos pastangos reiškia, kad projektui nustatomi tikslūs pradžios ir pabaigos momentai, o produkto unikalumas pabrėžia projekto rezultato naujumą, kuris patenkins tam tikrų klientų ar jų grupės poreikius [43]. Anglijos projektų vadybos asociacija (angl. *Association of Project Management*) į projektą žiūri kaip į veiklą, turinčią konkrečius tikslus, su tiksliais terminų, biudžetų ir pasiektų rezultatų kokybei reikalavimais [4].

Apskritai projektas – tai iš anksto gerai parengtas ir suplanuotas objekto ar sistemos sukūrimas ar jų pakeitimas [43]. Jis dažniausiai siejamas su investicine akcija, apimančia tikslą, išteklius, aplinką ir rezultatą. Kiekvienas projektas turi apribojimus ir prielaidas (angl. *constraints and assumptions*). Galima išskirti tris pagrindinius projekto apribojimus, kurie tarpusavyje turi būti subalansuoti: užmojis, laikas ir kaina.

Projekto užmojį sudaro parametrai, kuriuos turi apimti projektas. Projekto vykdymo metu projekto užmojis gali būti keičiamas, kai prie projektui iškeliami papildomi reikalavimai. Tada projektas reikalauja papildomo laiko ar papildomų sąnaudų. Sudarant projekto tvarkaraštį yra nustatomas projekto įgyvendinimo laikas. Tikslų ir praktišką tvarkaraštį sudaryti yra sunku, todėl jam turi būti skirta daug pastangų. Projekto įgyvendinimo kaina yra labai svarbus projekto aspektas, nes dažniausiai nuo jo priklauso kiti projekto veiksniai. Projekto vadovas turi rasti tokį metodą, kuris padėtų kruopščiai numatyti projekto įgyvendinimo kainą, jis taip pat privalo kontroliuoti projektą taip, kad jis neišeitų už numatyto biudžeto ribų.

1 pav. pateikta schema vaizduoja „geležinį“ užmojo, laiko ir kainos trikampį.



1 pav. Pagrindiniai projekto apribojimai

Be pagrindinių projekto apribojimų projekto vadovas turi atsižvelgti į projekto darbo pobūdį, projekto bei resursų kalendorius, įstatymus, nuostatas ir tiekėjus.

1.3. Projektų vadybos procesai ir metodologijos

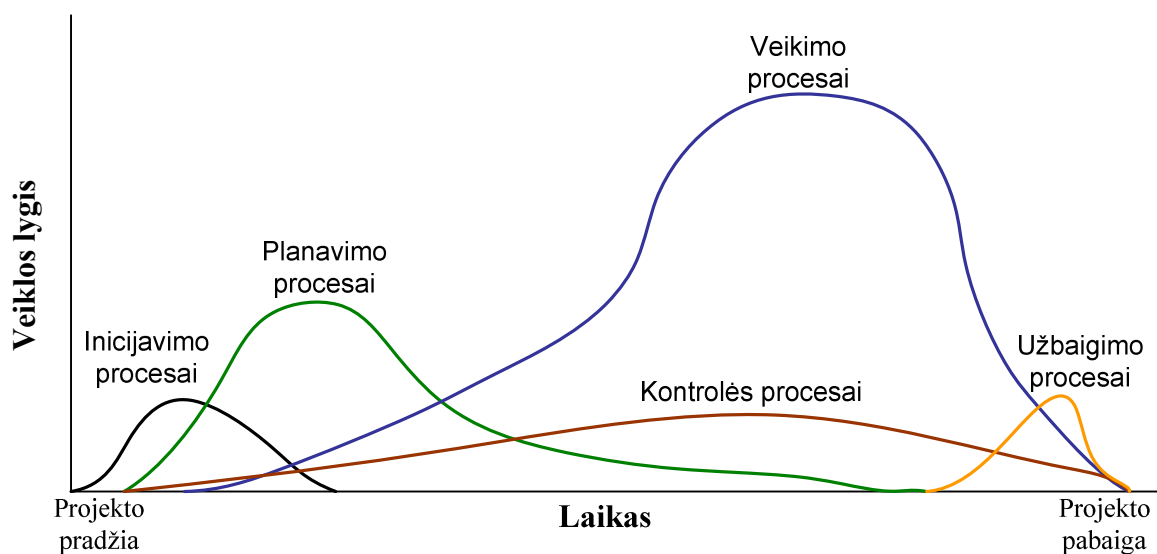
Projekto vadyba yra vykdoma per procesus. Procesas – tai veiksmų seka, sukurianti tam tikrą rezultatą. JAV Projektų vadybos institutas išskyrė penkis kompleksinių procesų grupes: inicijavimas, planavimas, vykdymas, kontrolė ir užbaigimas [1].

Projekto vadybos procesai glaudžiai susiję tarpusavyje, kiekvienam jų atlikti naudojama informacija, sukurta kito proceso valdymo metu. Be to, procesai yra vykdomi tam tikru nuoseklumu. Iš pradžių atliekamas projektų inicijavimas, kurio metu suformuojama ir atrenkama projekto idėja bei inicijuojamas projektas arba jo fazė. Paprastai projekto inicijavimo metu kyla papildomos informacijos poreikis, todėl yra pereinama prie projekto planavimo proceso. Projekto planavimas – tai nepertraukiamas procesas, kuriuo siekiama rasti geriausią veiksmų eigą, atitinkančią projekto aplinkoje susidariusią situaciją, numatyti reikiamus išteklius, sudaryti biudžetą [43].

Projekto vykdymo metu žmonių veikla ir kiti ištekliai yra koordinuojami projekto vadovo, kad būtų įvykdytas numatytas planas. Kadangi šie procesai yra sudėtingi, kartu su projekto planavimo ir vykdymo procesais prasideda projekto kontrolės ir reguliavimo procesas. Kontrolės procesas užtikrina projekto tikslų įvykdymą, nuolat stebint ir vertinant projektą ir nustatant nuokrypius nuo plano, kurie, jei reikia, turi būti pataisyti.

Proceso baigimo fazė dažniausiai trunka apie 12 viso projekto gyvavimo ciklo procentų [43]. Užbaigimo procese yra įforminamas projekto ar fazės priėmimas ir sutartų įsipareigojimų tarp užsakovo ir vykdytojo tvarkingas baigimas. 2 pav. pavaizduota, kaip projekto procesai yra susieti tarpusavyje laike [1].

Projekto vadybos procesai yra skirstyti į devynias žinių sritis: integravimo, užmojo, sąnaudų, kokybės, žmogiškųjų išteklių, komunikacijų, rizikos ir aprūpinimo vadybą [43].



2 pav. Projekto vadybos procesai

Projekto integravimo vadyba apima procesus, kurie užtikrina visų projekto elementų valdymą, projekto tikslų ir alternatyvų nustatymą. Ji reikalauja, kad projektas, nepriklausomai nuo jo dydžio ir reikšmės, būtų suderintas su kitais organizacijos darbais. Projekto integravimo vadyba apima tris procesus:

- **projekto planavimas** – visų projekto planų integravimas ir suderinimas, sukuriant nuoseklų ir aiškų dokumentą;
- **projekto vykdymas** – projekto plane numatytos veiklos įvykdymas;
- **integuota pokyčių kontrolė** – integruota visų projekto plano pokyčių kontrolė.

Projekto užmojo vadyba apima procesus, užtikrinančius, kad projekte būtų numatyti visi projekto apimčiai įgyvendinti reikalingi darbai. Projekto užmojo vadyba naudoja tokius procesus:

- **projekto inicijavimą** – naujo projekto pradžios sankcionavimą;
- **užmojo planavimą** – suformuluojamas projekto užmojis, kuris ateityje bus sprendimo priėmimo pagrindą;
- **užmojo apibrėžimą** – pagrindinių projekto dalių suskaidymą į smulkesnes, lengviau valdomas dalis;
- **užmojo patikrinimą** – formalų projekto užmojo patvirtinimą;
- **užmojo pokyčių kontrolę** – projekto užmojo pokyčių kontrolę ir valdymą.

Projekto laiko vadyba apima procesus, užtikrinančius projekto įvykdymą laiku. Sudarant projekto grafiką yra naudojami tokie pagrindiniai procesai:

- ***darbų nustatymas*** – numatomi darbai, reikalingi projekte numatytiems tikslams pasiekti;
- ***darbų eiliškumo nustatymas*** – nustatomi ir aprašomi susijusių darbų ryšiai;
- ***darbų trukmės įvertinimas*** – nustatoma atskirų darbų trukmė;
- ***grafiko sudarymas*** – darbų nuoseklumo, trukmės ir reikalingų resursų analizė;
- ***grafiko kontrolė*** – projekto grafiko pokyčių kontrolė.

Projekto sąnaudų vadyba apima procesus, reikalingus užtikrinti projekto įvykdymą pagal numatytąjį biudžetą. Vertinant projekto sąnaudas būtina nustatyti visus reikiamus resursus, jų kainą ir grafiką. Projekto sąnaudų vadyboje naudojami tokie procesai:

- ***resursų planavimas*** – nustatoma kiek ir kokių resursų reikia projekto darbams atlikti;
- ***sąnaudų įvertinimas*** – nustatoma apytikslė reikiamų resursų kaina;
- ***projekto biudžeto sudarymas*** – bendras sąnaudų įvertinimas, priskiriamas atskiriems darbams;
- ***sąnaudų kontrolė*** – projekto biudžeto pasikeitimų kontrolė ir valdymas.

Projektų kokybės vadyba apima procesus, kurie užtikrina, kad projektas tenkins numatytus kokybės poreikius ir atitiks organizacijos kokybės politiką. Kokybės vadyba yra tokia pat svarbi projekto valdymo žinių kaip kaštų ar laiko vadyba. Ji aprėpia visus projekto elementus ir sudaro tokius procesus:

- ***kokybės planavimą*** – nustatoma, kokie kokybės reikalavimai ir standartai bus taikomi projektui ir numatomi kokybės užtikrinimo būdai bei priemonės;
- ***kokybės laidavimą*** – reguliariai vertinama projekto vykdymo kokybė ir įsitikinama, kad atlikti projekto darbai tenkina keliamus kokybės reikalavimus;
- ***kokybės kontrolę*** – stebimi projekto rezultatai, nustatoma, ar buvo laikomasi kokybės standartų ir nustatomi pastebėtų trūkumų šalinimo būdai.

Projekto žmogiškųjų išteklių vadyba yra daugiaaspektinė ir apima visus procesus, reikalingus užtikrinti efektyvų į projektą įtrauktų žmonių (projekto komandos nariai, rėmėjai, klientai, partneriai, pagalbininkai, tarpininkai) veiklos panaudojimą. Žmogiškųjų išteklių vadyboje naudojami tokie procesai:

- **organizacinis planavimas** – nustatomi, dokumentuojami ir paskirstomi projekto išpareigojimai, vykdytojai ir jų tarpusavio ryšiai, teikiamos ataskaitos apie atliktus darbus;
- **personalo paieška** – numatytiems projekto darbams atlikti reikalingo personalo paieška ir atranka;
- **personalo vystymasis** – individualios ir grupinės kvalifikacijos kėlimo organizavimas, kad darbai būtų atlikti tinkamai ir laiku.

Projekto komunikacijų vadyba apima procesus, kurie užtikrina atitinkama projekto informacija būtų laiku pateikta, kaupiama, platinama, saugoma ir perduodama galutiniam vartotojui. Ji sukuria kritinius ryšius tarp žmonių, idėjų ir informacijos, kuri būtina sėkmingam projekto įgyvendinimui. Kiekvienas projekto dalyvis turi būti pasiruošęs bendrauti, siūsti ir gauti informaciją bei suprasti, kokią įtaką šis bendravimas turi visam projektui. Projekto komunikacijų vadyba apima tokius procesus:

- **komunikacijų planavimas** – nustatomas visų projekto dalyvių informacijos poreikis: kam, kada ir kokia informacija yra reikalinga ir kokia forma ją reikia perduoti;
- **informacijos paskirstymas** – reikalingos informacijos pateikimas projekto dalyviams reikiamu laiku;
- **įvykdyto darbo ataskaitų tiekimas** – darbo vykdymo informacijos (esamos būsenos apibūdinimas, progreso vertinimas ir prognozavimas) kaupimas ir platinimas;
- **administravimo baigimas** – projekto ar fazės baigimui reikalingos informacijos sukūrimas, surinkimas ir paskelbimas.

Projekto rizikos vadyba – tai sistemingas procesas, kurio metu yra nustatoma, analizuojama ir valdoma projekto rizika. Ji apima teigiamų įvykių tikimybės ir pasekmių maksimizavimą bei projekto tikslams nepalankių įvykių tikimybės ir pasekmių minimizavimą. Projekto rizikos vadyba apima tokius procesus:

- **rizikos vadybos planavimą** – projekto rizikos vadybos funkcijų nustatymas ir planavimas;
- **rizikos identifikavimą** – apibūdinami įvykiai, kurie gali paveikti projekto eigą ar jo elementus ir dokumentuojami jų parametrai;
- **kokybinę rizikos analizę** – atliekama projekto rizikos kokybinė analizė ir nustatomas rizikos pasekmių prioritetasis;

- **kiekybinę rizikos analizę** – kiekybiškai įvertinama apibrėžtos rizikos tikimybė ir nustatomas rizikos įtakos laipsnis projekto rezultatams;
- **rizikos reagavimo planavimą** – formuojamos procedūros ir metodai, kurie padės didinti projekto tikslų galimybes ir sumažinti jų grėsmes;
- **rizikos stebėjimą ir kontrolę** – esamos rizikos stebėjimas, naujos rizikos identifikavimas, rizikos mažinimo planų vykdymas ir jų efektyvumo įvertinamas viso projekto vykdymo metu.

Projekto aprūpinimo vadyba – tai procesas, kurio metu įsigijamos prekės ir paslaugos (medžiaga, įrangą, konsultantus ir kt.), reikalingos pasiekti projekto užmojų. Projekto aprūpinimo vadyba apima tokius procesus:

- **įsigijimų planavimą** – nustatoma ką ir kada reikia įsigyti;
- **prekybos per tarpininkus planavimą** – produktų reikalavimų dokumentavimas ir potencialių tiekėjų nustatymas;
- **prekybą per tarpininkus** – kviečiami potencialūs pardavėjai, renkami jų kvotos, kainos ir pasiūlymai;
- **tiekėjo pasirinkimą** – pasiūlymų svarstymas ir vertinimas, geriausio pardavėjo pasirinkimas;
- **sutarčių administravimą** – santykių su tiekėjais palaikymas;
- **sutarčių užbaigimą** – sutarčių užbaigimas ir sutvarkymas, sutarties įsipareigojimų įvykdymo patvirtinimas.

Projektų vadybos meistriškumo pasiekti neįmanoma be pasikartojančio proceso, vykstančio kiekvieno projekto metu. Šis pasikartojantis procesas remiasi projektų vadybos metodologija, kurią nuolat naudojant galima žymiai padidinti organizacijos sėkmę. Dauguma šiuo metu naudojamų projektų vadybos metodologijų pritaikomos nevisiškai [10]. Tik tinkamas projektų metodologijos panaudojimas gali maksimizuoti projekto vertę. Gera projektų vadybos metodologija turi vystytis ir derintis prie organizacijos pokyčių. Be to, svarbu kiekvienam projektui naudoti tą pačią metodologiją [31].

Tobulinant projektų vadybos metodologiją, kompanijoms teks integruoti į ją daugiau verslo procesų – tiekimo grandies vadybą (angl. *Supply chain management*), projekto įgyvendinimo tyrimas (angl. *Feasibility studies*), išlaidų ir naudos analizę (angl. *Cost-benefit analysis*), kapitalo sąmatos sudarymą (angl. *Capital budgeting*) ir kt.

Geros metodologijos savybės apima rekomenduojamą detalizavimo lygį, šablonų panaudojimą, standartizuotus planavimo ir tvarkaraščio sudarymo bei darbų kontrolės metodus, programos

pritaikymą visiems projektams, standartizuotų gyvavimo ciklo fazių panaudojimą. Yra svarbu tai, kad projektus valdo ne metodologijos, o žmonės. Metodologija yra vykdoma per organizacijos kultūrą, principus.

1.4. Projektų kokybės samprata

Kokybės sąvoką autoriai apibūdina įvairiai, tai priklauso nuo aspekto, kuriuo nagrinėjama kokybė. ISO 9000 standarte kokybė apibrėžiama kaip prekės ar paslaugos savybių ir bruožų visuma, kurios pagrindas – sugebėjimas patenkinti nustatytus ir numatomus poreikius [34]. Pasak A. Feigenbaum, kokybė – tai produktų ir paslaugų marketingo, dizaino, gamybos ir priežiūros charakteristikų visuma, kuri naudojimo metu padeda patenkinti vartotojo lūkesčius. W. E. Deming kokybę apibrėžia kaip esamų ir būsimų vartotojų poreikių patenkinimą. Projektų vadybos aspektu kokybė yra projekto komandos sugebėjimas įvykdyti tai, kas buvo nurodyta ir numatyta projekto užmojyje. Projekto kokybės lygis yra nustatomas vartotojo ir gali skirtis dėl projekto tikslų ir turinio.

Per pastaruosius du dešimtmečius stebima nuolatinė kokybės tobulinimo tendencija. Kokybės tobulinimas apima ne tik produkto kokybę, bet ir kokybės bei projektų kokybės valdymą. J. Charvat [10] išskyrė 8 kokybės požiūrio pasikeitimus (1 lentelė).

1 lentelė

Požiūrio į kokybę pasikeitimas

<i>Ankstesnis požiūris</i>	<i>Dabartinis požiūris</i>
Už kokybę atsako tik tiesiogiai susiję darbuotojai ir darbininkai	Kokybė – tai visų darbuotojų, taip pat tarnautojų ir netiesiogiai susijusių darbuotojų bei vadovų atsakomybė
Kokybės trūkumai turi būti paslėpti nuo klientų ir, jei įmanoma, nuo valdybos	Kokybės trūkumai turi būti akcentuoti ir pataisyti
Kokybės problemos veda prie kaltės, klaidos pateisinimo ir atleidimo	Kokybės problemos veda prie kooperatyvių pataisymo sprendimų
Kokybės problemos taisomos minimaliai dokumentuojant	Dokumentacija yra būtina „mokymuisi iš klaidų“, kad klaidos nesikartotų
Aukštesnė kokybė padidins projekto sąnaudas	Aukštesnė kokybė taupo pinigus ir padidina pelną
Kokybė nukreipiama į vidų	Kokybė nukreipiama į klientą
Kokybė atsiranda savaime	Žmonės gamina kokybiškus produktus
Kokybė sukuriama projekto vykdymo metu	Kokybė pradedama kurti projekto pradžioje ir turi būti planuojama viso projekto eigoje

Dauguma organizacijų į kokybę žiūri kaip į nenutrūkstantį tobulinimo procesą, o ne produktą. Šio proceso metu siekiama:

- išlaikyti esamus klientus;
- susigrąžinti prarastus klientus;
- pritraukti naujus klientus.

Didesnis šiuolaikinės organizacijos dėmesys skiriamas būtent procesų kokybei, o ne kokybiškam produktui, todėl kokybė yra tobulinama per nenutrūkstamą ciklą, kurio fazės apima kokybės planavimą (produkcijos nustatymas, klientų nustatymas, klientų reikalavimų nustatymas, reikalavimų konvertavimas į tiekėjų specifikacijas), kokybės organizavimą (darbo proceso žingsniai, matavimų pasirinkimas, procesų sugebėjimo nustatymas), produkcijos pagaminimą (darbo procesas), kokybės kontrolę (įvertinimas, problemų nustatymas) ir problemos sprendimą.

Kokybės vadyba – tai bendrosios valdymo funkcijos dalis, nustatanti kokybės politiką, tikslus ir pareigas bei tam naudojamas priemonės – kokybės planavimą, kokybės valdymą, kokybės užtikrinimą ir kokybės gerinimą, remiantis kokybės sistema.

Projektų kokybės vadyba apibrėžiama kaip veiklos būdų ir metodų, naudojamų siekiant įgyvendinti užsakovo pateiktus kokybės reikalavimus visuma. Efektyvių projektų kokybės vadybą sudaro vientisa projektų kokybės sistema, atitinkanti šiuolaikinius tarptautinius reikalavimus ir kryptingai veikianti tenkinant dabartinius ir būsimus klientų poreikius [43].

Kokybės vadybos ir projektų vadybos panašios savybės [1]:

- ***vartotojų tenkinimas*** – vartotojų poreikių supratimas, valdymas ir tenkinimas, įgyvendinant jų lūkesčius;
- ***prevencija*** – klaidų prevencijos kaina visada mažesnė, nei jų taisymo kaina;
- ***vadybos atsakomybė*** – sėkmingam projekto įvykdymui reikia visų projekto komandos narių dalyvavimo, tačiau vadyba privalo suteikti visus reikalingus išteklius;
- ***pasikartojantys procesai*** – E. Demingo aprašytas ciklas Planuoti–Daryti–Tikrint–Veikti yra panašus į fazių ir procesų derinį, kurį turi praeiti kiekvienas projektas.

Pasak B. Neverausko ir kt., šiandieninė projektų kokybės vadyba remiasi tokiais principais [43]:

- kokybė – tai neatskiriamas projekto charakteristikos visumos elementas;
- kokybė – tai, apie ką kalba projekto užsakovas, o ne vykdytojas;
- už projekto kokybę atsako konkretus žmogus;
- kokybę padidinti galima tik bendromis visos projekto komandos pastangomis;
- lengviau kontroliuoti projekto procesą, o ne galutinį rezultatą;
- projektų kokybės politika turi būti sudedamoji organizacijos filosofijos ir politikos dalis.

Šie projektų kokybės vadybos principai yra pagrįsti visuotinės kokybės metodologija. Remiantis visuotinės kokybės vadybos sistemos elementais projektų kokybės vadyboje yra keturi

svarbūs momentai: vadovybės atsakomybė, išteklių valdymas, proceso valdymas, matavimai ir analizė, gerinimas.

Projektų kokybės vadyba apima tris pagrindinius procesus (kokybės planavimas, laidavimas ir kontrolė), kuriais siekiama patenkinti iškeltus projekto kokybės poreikius. Šie procesai sąveikauja tarpusavyje bei su kitų projektų vadybos žinių sričių procesais. Kiekvienas procesas gali apjungti vieno ar kelių žmonių ar jų grupės pastangas, priklauso nuo to, kokie yra projekto reikalavimai. Kiekvienas procesas paprastai nors vieną kartą įvyksta kiekvienos projekto fazės vykdymo metu.

Pagal ISO 9000 standartą *kokybės planavimas* apibrėžiamas kaip kokybės vadybos dalis, nukreipta kokybės tikslams nustatyti ir reikiamiems veiklos procesams bei su jais susijusiems ištekliams, būtiniams kokybės tikslams pasiekti, apibrėžti. Projekto kokybės planavimas apima kokybės standartų, kurie turi būti pritaikyti projektui, identifikavimą ir jų pasiekimų būdų nustatymą. Tai vienas iš esminių projekto planavimo procesų, kuris turi būti vykdomas reguliariai ir paraleliai su kitais projekto planavimo procesais.

Pradedant kokybės planavimo procesus, projekto vadovas turi nustatyti kokybės lūkesčius, surinkti atitinkama informaciją apie organizacijos arba projektų kokybės politiką, projekto užmojų, produkto aprašymą, standartus ir kokybės reikalavimus projektams, kitų procesų sukurtą informaciją bei kokybės sistemos dokumentaciją. Kokybės planavimo metu projekto vadovas ir projekto komanda turi nustatyti planavimo reikalavimus, kaip juos galima patenkinti ir kiek tam reikia laiko ir sąnaudų.

Kokybės politika – tai bendri organizacijos kokybės tikslai ir kryptys, oficialiai pareikšti aukščiausios vadovybės [1, 34]. Organizacijos kokybės politika gali laikytis tokių formalių metodų, kaip ISO 9000, Šeši sigma, Visuotinės kokybės vadyba, Keizeno technologija arba turėti savo metodą, tenkinantį kokybės reikalavimus. Organizacijos kokybės politika turi būti pritaikoma vykdomiems projektams. Tačiau, jei organizacijoje nėra formaliai apibrėžtos kokybės politikos arba projektas vykdomas keliose organizacijose, projekto valdymo komanda turi sukurti ir pritaikyti projektų kokybės politiką. Nepriklausomai nuo kokybės politikos šaltinio, projekto vadovas turi informuoti apie ją visus projekto dalyvius, dalyviai turi ją suprasti ir jos laikytis.

Projekto užmojis yra pagrindinis projektų kokybės planavimo pradinė informacija, nes būtent jame yra surašyti pagrindiniai projekto tikslai ir rezultatai, kurie nusako klientų lūkesčius. Nors projekto užmojyje yra nustatytas pradinis produkto aprašymas, pats projekto aprašymas gali suteikti projekto vadovui ir komandai pagalbines informacijos techniniais ir kitais klausimais, turinčiais įtaką projekto kokybės planavimui.

Kiekvienos su projektu susijusios pramonės standartai ir taisyklės turi būti išanalizuoti, nustatytas tiek projekto plano, tiek kokybės plano priimtumas.

Be projekto užmojo ir produkto aprašymo projekto vadovas turi atsižvelgti į kitų projekto žinių sričių ir jų procesų rezultatinę informaciją, nes ji gali paveikti projekto kokybės planavimo procesą.

Planuojant projekto kokybę projekto vadovas gali naudotis skirtingais metodais ir priemonėmis [1, 43]:

- **Išlaidų ir pajamų analizė** (angl. *Benefit/Cost analysis*) – tai procesas, kurio metu yra nustatomi bet kokio proceso, produkto ar veiklos „už ir prieš“. Šis metodas lygina kokybės vadybos veiklos naudą ir jos sąnaudas. Pagrindinė kokybės reikalavimų tenkinimo nauda – pataisymų mažinimas, o tai reiškia didesnę produktyvumą, mažesnes sąnaudas ir klientų pasitenkinimą. Prie kokybės išlaidų galima priskirti projektų kokybės vadybos veiklos sąnaudas.
- **Gairės** (angl. *Branchmarking*). Jos padeda lyginti projektus ir jų veiklą. Šis metodas leidžia palyginti projekto kokybės rodiklių reikšmes su analogiškų projektų kokybės rodiklių reikšmėmis, padaryti atitinkamas išvadas, kurios padės išvengti projekto kokybės plano klaidų.
- **Struktūrinės schemos** (angl. *Flowcharting*). Techniniu požiūriu, struktūrinė schema – tai bet kokia diagrama, iliustruojanti sistemos komponentų veiklą. Projektų kokybės vadyboje naudojamos tokios struktūrinės schemos (1 priede pateikti diagramų pavyzdžiai):
 - priežasčių – pasekmių diagramos (angl. *Cause-and-effect diagrams*), dar vadinamos „žuvies kaulo“ ar Ishikawos, kurios iliustruoja atitinkamų priežasčių ir pasikeitusios projekto kokybės ryšį;
 - sistemos ar proceso srautų diagramos (angl. *System or process flow charts*) parodo, kaip atskiri sistemos elementai sąveikauja tarpusavyje.
- **Eksperimentai**. Eksperimentų sudarymo metodas naudoja statistinius „kas, jei“ scenarijus, kuriais nustatomos geriausio projekto rezultato reikšmės. Paprastai šis metodas pritaikomas projekto produktui, o ne pačiam projektui, tačiau jis gali būti sėkmingai pritaikytas projekto vadybos veiklai. Eksperimentai taip pat panaudojami nustatant, kokios projekto ar produkto reikšmės turi neigiamus, nepriimtinius rezultatus.
- **Kokybės kaštai**. Kokybės kaštai nurodo bendrus produkto ar paslaugos kokybės pasiekimo kaštus. Kokybės kaštai yra skirstomi į tris kategorijas: įvertinimo, prevencijos ir nuostolių [58]. Įvertinimo ir prevencijos kaštai dažniausiai vadinami atitikties kaštais. Nuostolių kaštai yra skirstomi į vidinius ir išorinius.

Kokybės planavimo proceso rezultatas – rasti metodą, kuris bus naudojamas vykdant kokybės politiką. Prie projektų kokybės planavimo rezultatinę informaciją galima skirti kokybės valdymo planą, naudojamus apibrėžimus, kontrolinius sąrašus ir kitų procesų pradinę informaciją [1].

Kokybės valdymo plane aprašoma, kaip projekto valdymo komanda vykdys kokybės politiką. Pagal ISO 9000 terminologiją, šis planas turi aprašyti projekto kokybės sistemą: organizacinę struktūrą, atsakomybes, procedūras, procesus ir išteklius, kuriais bus įgyvendinta kokybės vadyba. Kokybės valdymo plane nurodomi trys aspektai: projekto kokybės kontrolė, kokybės laidavimas ir kokybės gerinimas. Jis gali būti formalus arba neformalus, detalizuotas arba bendrai apibrėžtas ir pagrįstas projekto reikalavimais.

Naudojami apibrėžimai yra kiekybiškai įvertinamos sąvokos ir reikšmės, kuriomis kokybės valdyme yra matuojami procesai, veiklos ir darbo rezultatai. Kai kuriose dalykinėse srityse naudojami apibrėžimai yra dar vadinami metrika.

Kontroliniai sąrašai – tai vienas iš paprasčiausių būdų, užtikrinančių, kad darbas yra atliekamas laikantis kokybės politikos. Kontroliniai sąrašai gali būti paprasti arba sudėtingi. Dažniausiai jie formuluojami kaip įsakymas („Padaryk tai!“) arba kaip klausimas („Ar padarėte tai?“).

Kokybės planavimo proceso metu gali būti nustatytas kitos srities papildomos veiklos poreikis.

Antras projektų kokybės vadybos procesas yra kokybės laidavimas, dar vadinamas kokybės užtikrinimu. Kokybės laidavimas – tai suplanuota, sistemingai vykdoma, laikantis apibrėžtos kokybės sistemos, veikla, kurios tikslas yra užtikrinti, kad projektas atitiktų tiesiogiai susijusius kokybės standartus [1]. Kokybės laidavimą taip pat galima apibrėžti kaip viso projekto veiklos įvertinimo procesą, kurio tikslas – įsitikinti, kad projekto metu sukurtas produktas yra būtino kokybės lygio [31, 4]. Būtent šio proceso metu bandoma visiškai integruoti projekto užmojo, kainos ir laiko funkcijas.

Dažniausiai, bet nebūtinai, kokybės laidavimo procesas yra vykdomas Kokybės laidavimo skyriaus arba kito panašaus organizacijos padalinio projekto eigoje, o ne jo pabaigoje. Galima išskirti vidinius ir išorinius kokybės laidavimo tikslus [1, 60]:

- **vidinį kokybės laidavimą** – kokybės laidavimas organizacijos viduje sukuria projekto vadybos komandos ir vadovų pasitikėjimą;
- **išorinį kokybės laidavimą** – tam tikrose situacijose kokybės laidavimas suteikia vartotojams ar kitiems asmenims pasitikėjimo.

Šis procesas remiasi tiek prieš tai patvirtintu kokybės planu, naudojamais apibrėžimais, kokybės patikrinimo ataskaitomis, tiek kita dokumentacija. Kokybės kontrolės įvertinimų rezultatai yra vienas iš svarbesnių šio proceso pradinių duomenų ir apima kokybės kontrolės ir jos kokybiško įvertinimo įrašus. Šie įrašai yra pateikiami tokiu formatu, kuris leistų juos lyginti ir analizuoti.

Kokybės laidavimas yra įgyvendinamas atliekant planinį ir neplaninį patikrinimą įvairiais kokybės tikrinimo metodais. Projekto vadovas ir komanda gali panaudoti tuos pačius kokybės planavimo metodus ir priemones, taip pat atlikti kokybės auditą. Kokybės auditas – tai nepriklausoma, struktūrinė kitos kokybės vadybos veiklos apžvalga, atliekama kvalifikuoto personalo. Jo tikslas – nustatyti einamojo projekto klaidas ir nustatyti būdą, kaip pagerinti šį projektą, bei kitus organizacijoje vykdomus projektus. Kokybės auditas gali būti planinis arba atsitiktinis, vykdomas atitinkamai apmokyto organizacijos personalo arba trečių šalių.

Geras kokybės auditas turi užtikrinti, kad [31]:

- planuota projekto kokybė bus įvykdyta;
- produktai yra saugūs ir tinkami vartojimui;
- yra laikomasi visų susijusių teisės aktų ir įstatymų;
- duomenų rinkimo sistemos yra tikslios ir adekvačios;
- tinkami pataisos veiksmai bus atliekami tada, kai reikia;
- yra nustatytos tobulinimo galimybės.

Kokybės laidavimo proceso rezultatas – kokybės tobulinimas, apimantis projekto efektyvumo ir veiksmingumo gerinimo veiksmus, projektų dalyviams suteikiančius papildomą naudą. Kokybės tobulinimo realizavimas dažnai reikalauja pakeitimų ir pataisymų, prašymo parengimo ir yra atliekamas vadovaujantis pakeitimo kontrolėje numatytais procedūromis.

Paskutinis, bet ne mažiau svarbus projekto kokybės vadybos procesas yra projektų kokybės kontrolė. Projektų kokybės kontrolė – tai nuolatinis konkrečių projekto rengimo ir įgyvendinimo rezultatų sekimas ir įvertinimas, nustatantis, ar jie atitinka kokybės standartus ir užsakovo iš anksto patvirtintus kokybės reikalavimus [43]. Projektų kokybės kontrolė turi būti atliekama viso projekto metu ir turi apimti tiek kuriamo produkto, tiek projekto vadybos (veiklos, kalendoriaus ir sąnaudų pasikeitimų) kokybės patikrinimo rezultatus. Jei kokybės kontrolės metu buvo nustatyti nepatenkinami rezultatai, atliekama pagrindinės priežasties analizė (angl. *Root Cause Analysis*). Šios analizės metu projekto vadovas nustato užfiksuotos neatitikties priežastį, parenka pašalinimo būdą ir pritaiko atitinkamą taisymo veiksma.

Projektų kokybės kontrolei būtina informacija apie projekto eigą, darbo rezultatų, projekto kokybės planą, naudojamus apibrėžimus ir kontrolinius sąrašus. Darbo rezultatus apima tiek projekto procesų, tiek produkto rezultatai, kurie yra vertinami ir lyginami su naudojamais kokybės standartais.

Kokybės kontrolė yra įgyvendinama tokiais metodais ir priemonėmis [1]:

- **Patikrinimai** apima tokius veiksmus, kaip projekto procesų ir produkto vertinimą, nagrinėjimą ir testavimą, kuriais nustatoma, ar rezultatai atitinka reikalavimus. Tikrinti

galima visą projektą arba jo atskirą dalį. Patikrinimai taip pat žinomi kaip apžvalgos, produkto apžvalgos, auditai.

- **Kontrolės diagramose** proceso rezultatai grafiškai pavaizduojami laike (2 priedas). Kontrolės diagramos paprastai naudojamos projektuose su pasikartojančia veikla (pavyzdžiui, gamyboje), tačiau taip pat sėkmingai gali būti naudojamos stebint projekto tvarkaraščio ir sąnaudų, projekto užmojo apimties pokyčius, projekto dokumentų klaidas ir kitus vadybos rezultatus. Šios diagramos naudojamos nustatant, ar procesas yra kontroliuojamas. Proceso kontrolės metu jis neturi būti koreguojamas, jį galima pagerinti, bet nekoreguoti. Kontrolės diagramas galima naudoti bet kokio tipo kintamojo rezultato kontrolės metu.
- **Pareto diagramos** yra duomenų histogramos, kuriose kokybės problemų duomenys išdėstomos nuo dažniausiai iki rečiausiai pasitaikančių duomenų. Toks kategorijų sutvarkymas padeda atlikti tinkamus pataisymo veiksmus, nes, norint pagerinti projekto kokybę, pirmiausia būtina pašalinti problemas, sukeliančias dažniausiai pasitaikančius defektus. Pareto diagramos yra tiesiogiai susijusios su Pareto dėsniumi, pagal kurį sąlyginai mažas priežasčių skaičius tipiškai sukelia didesnę problemų ar defektų dalį. Šis dėsnis dar yra vadinamas 80/20 principu, kur 80 proc. problemų atsiranda dėl 20 proc. priežasčių. Pareto diagramos pavyzdys pateiktas 3 priede.
- **Statistinė atranka** tai procesas, kurio metu yra pasirenkamas atsitiktinis procentinis rezultatų dydis, tam tikra tikrinamos visumos dalis. Tinkama atranka dažnai gali sumažinti kokybės kontrolės kaštus. Naudojant šį metodą gali atsirasti nevienareikšmių rezultatų, jei nebuvo laikomasi adekvataus testavimo plano ir tvarkaraščio.
- **Trendo analizė** naudoja matematinius metodus, skirtus numatyti būsimų veiksmų pasekmes pagal ankstesnius rezultatus. Trendo analizės metu reikia atitinkamų įrašų, kuriais bus remiamasi numatyti rezultatus ir nustatyti esamus lūkesčius. Šis metodas dažniausiai naudojamas tikrinti techninį (kiek buvo nustatyta klaidų ar defektų, kiek dar nepataisyti) ir sąnaudų bei tvarkaraščio funkcionavimą (kokia veikla buvo atlikta su reikšmingais nuokrypiais). 4 priede pateiktas trendo analizės grafinio pavaizdavimo pavyzdys.

Projektų kokybės kontrolė turi baigtis kokybės pagerinimu. Remiantys kokybės kontrolės panaudotų metodų ir priemonių rezultatais projekto vadovas ir komanda turi atlikti pataisymų veiksmus, kuriais turi būti eliminuota nepriimtina kokybė ir pagerinta visų projekto vadybos procesų kokybė.

Gera projektų kokybės kontrolės sistema turi apimti išsamią visų procesų dokumentaciją, metodologines instrukcijas pagal projektų kokybės sistemos elementus, darbo ir kontrolės instrukcijas, aprašančias kontrolės procedūras, techninę dokumentaciją bei techninę literatūrą.

2. PROGRAMINĖS ĮRANGOS KŪRIMO PROJEKTAI

Per pastaruosius trisdešimt metų programinės įrangos industrija sparčiai išaugo. 1996 metų programinės įrangos pardavimo pajamos sudarė 120 milijardų dolerių, o 2006 metais – 919 milijardų eurų. Nors apie 50 procentų visos programinės įrangos rinkos sukoncentruota JAV, spartus programinės įrangos industrijos augimas pastebimas ir kitose besivystančiose šalyse [21].

Programinės įrangos industrija pasiekė globalizacijos amžių. Galima išskirti kelias programinės įrangos globalinio „outsorsingo“ paplitimo priežastis, tai:

- sistemų kūrėjų stygius visame pasaulyje;
- žymūs darbo jėgos kainos skirtumai įvairiose valstybėse;
- augantis neatliktų darbų palaikymas;
- globaliai naudojamų sistemų vietinių pakeitimų poreikis.

Paprastai apie 60-80 proc. organizacijos programinės įrangos biudžeto ir daugiau nei 50 proc. organizacijos informacinių sistemų darbuotojų skirti palaikyti, o ne kurti programinę įrangą. Patobulintos technologijos (pvz., pagerintos telekomunikacijos, didesnis programinės įrangos standartizavimas) programinės įrangos kūrimą ir palaikymą leidžia vykdyti ir stebėti iš nuotolinės vietos.

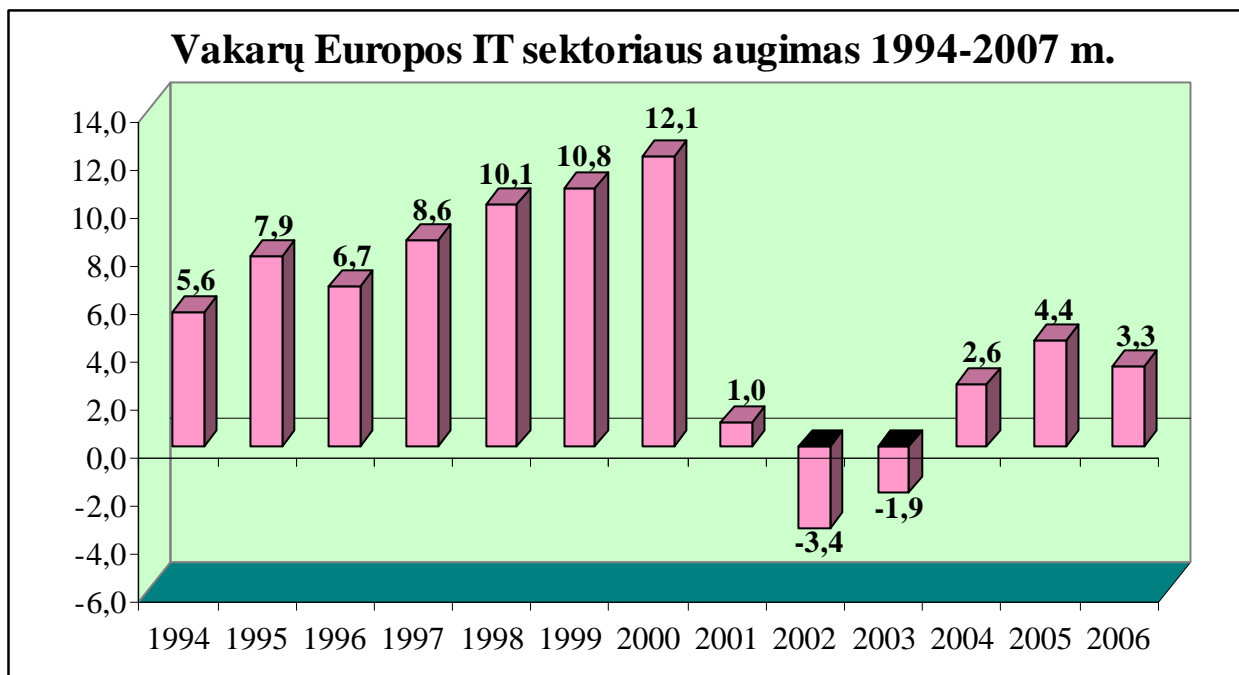
Daug organizacijų suprato, kad jų globalus ir produktyvus plėtimas leidžia kurti kokybiškas prekes ir paslaugas. 1991 metų Programinės įrangos projektavimo instituto (angl. *SEI – Software Engineering Institute*) įvertinime nustatyta, kad 93 proc. įvertintų organizacijų neturėjo gerai apibrėžto programinės įrangos kūrimo proceso. Iš jų 81 proc. neturėjo kokybės valdymo sistemos, nepaisant to, kad yra pakankamai daug tarptautinių standartų ir sistemų, galinčių suteikti programinės įrangos industrijai visas priemones sukurti kokybės sistemas, pritaikytas jų procesams. Tokių standartų ir sistemų pavyzdžiais gali būti ISO 9000 standartų šeima, TickIT, Brandos proceso modelis (angl. *CMM – Capability Maturity Model*), Programinės įrangos proceso gerinimo pajėgumo nustatymo (angl. *SPICE – Software Process Improvement Capability Determination*) standartai.

2.1. PĮ rinkos apžvalga

Programinės įrangos rinkos sektorių vienareikšmiškai nustatyti sunku, kadangi skirtinguose šaltiniuose jis apibūrinamas nevienodai. Dažnai jis įvardijamas kaip sudedamoji informacinių ir komunikacinių technologijų rinkos segmento dalis. Remiantis Europos informacinių technologijų stebėjimo tarnybos duomenimis [11, 23, 24], informacinių technologijų sektorių sudaro tokie komponentai: programinė įranga ir paslaugos, techninė įranga, duomenų apdorojimo ir duomenų

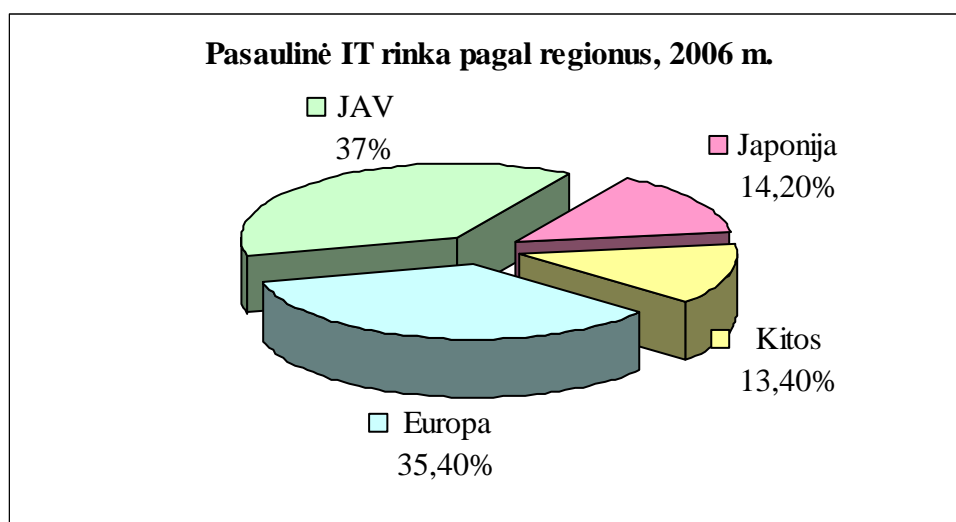
perdavimo įranga. Programinės įrangos ir paslaugų sektoriui priskiriama sisteminė bei taikomoji programinė įranga, konsultavimai, programinės įrangos diegimas, procesų valdymas ir palaikymas.

IT sektoriaus augimas pastebimas antroje 1990-ųjų metų pusėje (3 pav.). Nežiūrint į tai, kad Interneto ir elektroninės prekybos plėtra išskėlė programinės įrangos ir kompiuterinių paslaugų poreikį, pastaraisiais metais IT sektorius patyrė fundamentalias permainas. 2000 metų pradžioje prasidėjo šio sektoriaus nuosmukis, kuris turėjo didelę įtaką viso pasaulio verslo įmonių veiklai. Tačiau, kaip matyti iš grafiko, IT sektorius pradėjo atsigaivinti, jo augimo tempai 2005 metais sudarė 4,4 proc.



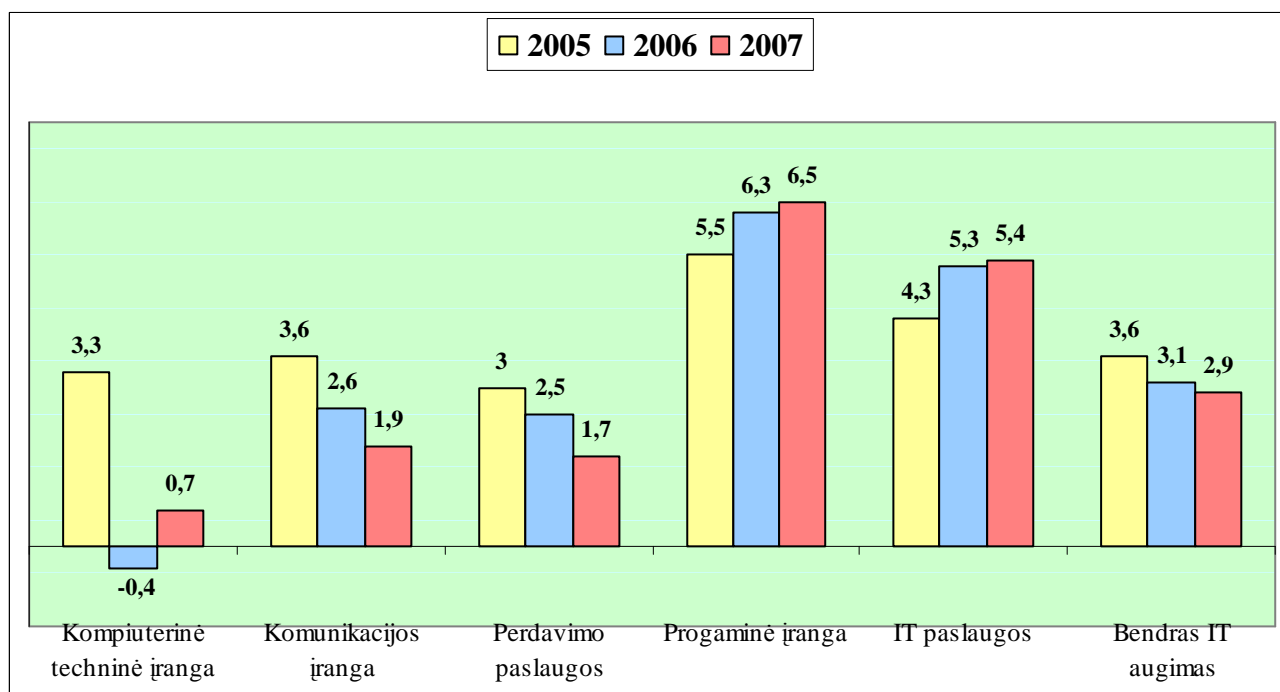
3 pav. Vakarų Europos IT sektoriaus augimas 1994–2007 m., % [eito2007].

2006 m. EITO tyrimų duomenimis, Europos šalių pasaulio IT sektoriuje sudaro 35,4 proc. - apie 300 milijardų eurų (4 pav.). Didžiausią IT rinkos dalį turi JAV – 37 proc., nes dauguma didelių tarptautinių korporacijų, veikiančių IT sektoriuje, ištakos yra JAV.



4 pav. Pasaulinė IT rinka pagal regionus, 2006 m. [23].

IT rinką galima išskaidyti į tokius struktūrinius elementus: kompiuterinė techninė įranga, komunikacijos įranga, duomenų perdavimo paslaugos, programinė įranga ir IT paslaugos. Kaip matyti 5 pav., didžiausias augimas Europos Sąjungos valstybėse vyksta programinės įrangos (2006 m. – 6,3 proc., 2007 m. prognozuojamas augimas – 6,3 proc.) ir IT paslaugų (2006 m. – 5,3 proc., 2007 m. prognozuojamas augimas – 5,4 proc.) srityje.



5 pav. Europos Sąjungos IT rinkos augimas pagal segmentus, 2005–2007 m. [23].

IT sektorius auga greičiau nei telekomunikacijų. 2006 m. programinės įrangos rinkos segmentas išgyveno didžiausią dinamiką (išaugo 6,3 proc.). Pastebima didelė sistemų infrastruktūros ir

saugos priemonių paklausa. IT paslaugų sektoriaus augimas nežymiai mažesnis – 5,3 proc. Tokios profesionalios paslaugos, kaip paslaugų nuoma, konsultavimas ir sistemų integravimas, turi didžiausią įtaką IT paslaugų rinkos augimui, nes vis daugiau kompanijų bando pagerinti savo veiklos efektyvumą ir produktyvumą per IT paslaugų tiekėjus. EITO prognozių duomenimis, programinės įrangos ir paslaugų sektorius turi augti ir 2007 metais.

Pagrindiniai veiksniai, darantys įtaką programinės įrangos ir IT paslaugų sektorių augimui, yra internacionalizacija, specializacija, inovacijos ir kiti technologiniai pokyčiai. Programinės įrangos ir IT paslaugų sektoriaus internacionalizacija įvyko dėl tiesioginių užsienio investicijų. Daugiausiai kompanijų, besinaudojančių internacionalizacijos privalumais, yra iš JAV, kurios pagrindinis investicijų taikinytis buvo Europa. Pavyzdžiui, Airijoje, Nacionalinio programinės įrangos direktorato duomenimis, tik 140 iš 900 kompanijų, 2002 m. dirbančių programinės įrangos srityje, buvo užsienio įmonės, tačiau jų pajamos sudarė apie 89 proc. visų programinės įrangos sektoriaus metinių pajamų.

Pastaraisiais metais investicijos į programinės įrangos sektorių padidėjo visame pasaulyje, ypač Indijoje, Japonijoje ir Izraelyje. Kompanijų susijungimai ir susiliejinimai yra viena iš labai svarbių priemonių, sudarančių naujas programinės įrangos sektoriaus augimo galimybes, kadangi vis daugiau kompanijų, prisitaikydamos prie greitai besikeičiančios rinkos, išnaudoja atsirandančias technologijas. Specializacija yra svarbus veiksnys, lemiantis programinės įrangos ir IT paslaugų sektoriaus sėkmę ir augimą. Neišplėtos ir neišnaudotos rinkos nišos radimas leidžia sėkmingai plėtotis vietinėms įmonėms. Informacinių technologijų rinka pasižymi greitais pokyčiais, todėl svarbu neprarasti kritinio momento – naujo produkto atsiradimas – pranašumo. Siekiant tai įgyvendinti, būtina nuolat palaikyti aukšto lygio investicijas į inovacijas, tyrimus ir plėtrą.

Plėtojantis interneto paslaugoms, jas realizuojančioms taikomosioms programoms ir jas valdančioms informacinėms technologijoms, ateina permainos ir į programinės įrangos sektorių. EITO duomenimis, programinės įrangos ateitis vis labiau priklauso nuo perėjimo iš produkto į paslaugos kategoriją. EITO išskiria pagrindinius veiksnius, skatinančius programinės įrangos ir IT paslaugų sektoriaus augimą:

- ekonomikos atsigavimą ir investicijas į pramonę;
- kvalifikuotų specialistų skaičiaus augimą;
- didėjančią verslo procesų priklausomybę nuo informacinių technologijų ir jų plėtos;
- investicijas į tyrimus ir plėtrą;
- stiprėjančius ryšius tarp informacinių technologijų ir komunikacijų sektorių.

EITO tyrimų duomenimis, Europos Sąjungos valstybių IT rinkos augimas skiriasi. 2006 m. didžiausi augimo tempai buvo naujosiose Europos Sąjungos valstybėse narėse.

Lietuvoje kiekvienais metais atsiranda vis daugiau įmonių, kurių pagrindinė veiklos kryptis – programinės įrangos kūrimas ir IT paslaugų teikimas. Programinės įrangos kūrimo proceso gerinimas ir sertifikavimas yra svarbus Lietuvos IT bendrovėms, kurių tikslas – kokybiško programinės įrangos produkto sukūrimas. Lietuvoje gerėjant ekonominei situacijai, IT bendrovės yra pajėgios skirti lėšų, kad būtų užtikrinta produktų kokybė, investuoti į kokybę jas verčia didėjanti konkurencija ir vartotojų poreikiai. Lietuvos IT bendrovės neseniai žengė link pripažintos kokybės. 2000 m. bendrovei „Sidabrinis tinklas“ buvo suteiktas tarptautinis kokybės pažymėjimas ISO 9001. Ji tapo pirmąja informacijos technologijų kompanija Lietuvoje, pelnusia Nacionalinį kokybės prizą [16]. Siekdamas patekti į Europos Sąjungos ir kitas tarptautines IT rinkas bei tenkinti pasaulinius standartus, Lietuvos IT bendrovės turi skirti vis daugiau dėmesio kokybei, įrodydamos gebėjimą sukurti reikiamos kokybės programinės įrangos produktą.

Dauguma didelių Lietuvos IT bendrovių suprato, kad tik bendro kokybės standarto ISO 9000 įdiegimo nepakanka, norint įrodyti, kad organizacija skiria pakankamai resursų produkto kokybei. Todėl, atsižvelgiant į pasaulinę IT kompanijų praktiką, Lietuvos IT bendrovės pradeda diegti tokius specifinius standartus: Programinės įrangos kūrimo proceso brandos modelį, Programinės įrangos kūrimo proceso tobulinimo sistemą, ISO 15504, ISO 9126 ir kt. Be to, dalis bendrovių pasirinko kitą kelią – darbą perorganizavo remdamosi tokiais proceso modeliais kaip RUP (angl. Rational Unified Process) ir Ribinis programavimas (angl. XP – eXtreme Programming).

Organizacijų veiklai pertvarkyti pagal standartų reikalavimus reikia ilgo pasirengimo, apimančio pačių reikalavimų supratimą, savo veiklos dokumentavimą ir atitikimo standartų reikalavimams nustatymą, didelių investicijų darbuotojų apmokymams ir metodinei medžiagai, todėl tik didžiausios ir finansiškai stipriausios Lietuvos IT įmonės yra pajėgios savarankiškai gerinti savo veiklą [16].

Remiantis Prime Investment atliktų tyrimų duomenimis, 2005 m. pagal sukurtą IT paslaugų apyvartą ir pridėtinės vertės sukūrimo efektyvumą buvo sudarytas didžiausių Baltijos šalių IT paslaugas teikiančių kompanijų sąrašas (2 lentelė).

Darbe analizuojama 2 lentelėje pateiktų kompanijų kokybės valdymo sistemos, nes būtent šių kompanijų indėlis į Baltijos šalių programinės įrangos ir IT paslaugų rinką Prime Investment duomenimis yra didžiausias. Pagrindinė šių kompanijų veiklos kryptis – programinės įrangos kūrimas.

Kompanijos, siekdamos kurti kokybiškus programinės įrangos produktus, kurie galėtų konkuruoti platesnėje rinkoje, darbą organizuoja remiantis atitinkama kokybės valdymo sistema ar metodologija. Pastaroji yra pritaikoma prie programinės įrangos kūrimo proceso.

Aptarsime konkrečias kompanijas ir jų naudojamas kokybės valdymo sistemas.

Baltijos IT kompanijų dvidešimtukas [56]

Kompanija	2005 m.	2004 m.	Augimas, %
Exigen Group (LV)	19994	17585	14
Sonex Group (LT)	15639	10861	44
MicroLink Eesti (EE)	8115	7412	9
Alna Group (LT)	7768	7697	1
Elsis Group (LT)	4300	3840	12
Baltic Data Center (LT)	4021	3615	11
MicroLink Lietuva (LT)	4018	2823	42
Helmes (EE)	3727	2234	67
Webmedia (EE)	3712	2083	78
Blue Bridge (LT)	3539	2007	76
Sintagma Group (LT)	2882	2491	16
Tilde SIA (LV)	2636	2260	17
New Vision Baltija Group (LT)	2143	2027	6
Regio (EE)	2097	496	322
Algoritmų sistemos (LT)	1722	1970	-13
HNIT-Baltic Geoinfoservisas (LT)	1702	1183	44
Baltic Amadeus (LT)	1561	1277	22
Prototechnika Group (LT)	1334	1300	3
Santa Monika Networks (EE)	1248	582	114
Edrana (LT)	1206	1056	14

Exigen Group suteiktas ISO 9001:2000 kokybės sistemos sertifikatas. Kompanija, vykdydama projektus, remiasi PMI projektų vadybos metodologija. Jos programinės įrangos kūrimo proceso brandos lygis yra 4, taip vadinamas valdomas lygis. ISO 9001:2000 standartą atitinka daugelis kitų 2 lentelėje nurodytų kompanijų (Sonex Group, Regio, Algoritmų sistemos, HNIT-Baltic Geoinfoservisas, Baltic Amadeus, Edrana). Tačiau, kaip buvo minėta, dėl to, kad ISO 9001:2000 standarto taikymas yra sudėtingesnis (jį taikant būtina atsižvelgti į kitų ISO šeimos standartus ISO 15504, ISO 12207, ISO 90003) negu CMM ar CMMI, kitos įmonės būtent pastaruosius ir taiko. Alna turi 2, kartojamojo, lygio CMM sertifikatą. Sintagma Group savo veikloje orientuojasi į tarptautinį programinės įrangos gamybos standartą bei CMMI metodiką. Kompanija Webmedia pasirinko kitą kelią. Jos programinės įrangos kūrimo projektai grindžiami RUP.

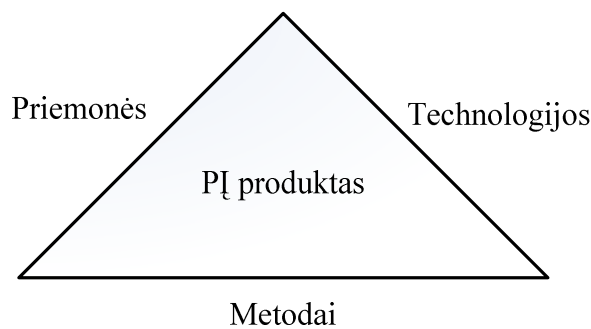
Kaip matyti, kokybiškas programinės įrangos produkto sukūrimas įmanomas tik naudojant kokybės valdymo sistemas ir metodologijas, jas pritaikant prie konkretaus įmonės programinės įrangos kūrimo proceso.

2.2. Programinės įrangos kūrimo projektų valdymo ypatumai

Sėkmingas programinės įrangos projektų valdymas turi apimti metodus, priemones ir technologijas, kurie dėl tam tikro proceso sąveikauja tarpusavyje (6 pav.). Šie trys elementai sudaro programinės įrangos kokybės ir valdymo pagrindą. Programinės įrangos projekto kontekste metodas apibrėžiamas kaip procesas ar būdas, kuriuo sukuriamas kokybiškas programinės įrangos produktas.

Priemonė apibrėžiama kaip darbo arba užduoties sprendimo įrankis ar mechanizmas, padedantis sukurti kokybišką produktą. Programinės įrangos priemonės galima išskirti į grupes, atsižvelgiant į programinės įrangos kūrimo gyvavimo ciklo procesus: reikalavimus, projektavimo priemones, kūrimą, testavimą, palaikymą, konfigūracijų valdymą, gyvavimo ciklo projektavimą, kokybės valdymą ir kitas priemones.

Technologijos apibrėžiamos kaip sistemingų žinių pritaikymas programinės įrangos projektuose.



6 pav. *Technologijos, metodai ir priemonės*

Robert T. Futrell ir kt. nustatė tris pagrindinių kompetencijų, kurias privalo turėti sėkmingas programinės įrangos projektų vadovas, kategorijas: produktas, projektas ir žmonės [12]. Produkto kūrimo metodai apima procesų ir jų alternatyvų įvertinimo, procesų standartų išmanymo, reikalavimų valdymo, projekto metodų ir priemonių pasirinkimo, PĮ produkto kokybės stebėjimo ir PĮ kūrimo veiklų supratimo kompetencijas. Be specifinių produkto kompetencijų sėkmingas projekto vadovas privalo turėti projekto valdymo įgūdžius, kurie padeda kurti projekto struktūrą, dokumentuojamus projekto planus, sudaryti efektyvų projekto tvarkaraštį ir įvertinti reikalingas pastangas ir sąnaudas, pasirinkti tinkamą projektų valdymo priemonę, stebėti projektą ir jo progresą bei valdyti riziką.

Sėkmingo projekto vadovo žmonių valdymo įgūdžiai apima projekto komandos ir jos narių pasirinkimą, darbo komandos veiklos vertinimą, profesinės veiklos planavimą. Be to, projekto vadovas turi būti komunikabilus, sėkmingai vesti derybas ir spręsti konfliktus.

Proceso valdymas yra vienas iš svarbesnių PĮ kūrimo projektų uždavinių ir turi būti apsvarstytas projekto pradžioje. Proceso valdymas užtikrina teisingą organizacinių procedūrų, strategijų ir gyvavimo ciklo modelio vykdymą. Jis kontroliuoja visas PĮ kūrimo veiklas. Kadangi PĮ

kūrimas yra nepaprastai sudėtingas ir kiekviena projekto veikla gali turėti daug alternatyvų, apibrėžtas procesas gali žymiai palengvinti projekto vadovo ir komandos narių darbą.

Kadangi PĮ kūrimo projektai labai skiriasi, jų kūrimo procesai turi būti skirtingi, pritaikyti prie specifinių organizacijos poreikių. Taigi PĮ kūrimo projekto pradžioje turi būti pasirinktas tinkamas projekto gyvavimo ciklas. Tipinis technologinis procesas yra metodinis PĮ gyvavimo ciklo pagrindas. Jį sudaro etapai ir operacijų sekos, užtikrinančios tvarkingą darbų atlikimą visose projekto fazėse.

Koncepcinis ir organizacinis gyvavimo ciklo ir programinės įrangos kokybės valdymo pagrindas yra apibrėžiamas aštuoniais pagrindiniais principais (orientacija į klientą, lyderystė ir vadovavimas, personalo įtraukimas, procesinis požiūris, sisteminio administravimo valdymo požiūris, nuolatinis tobulinimas, sprendimų priėmimas remiantis faktais ir naudingi santykiai su tiekėjais), deklaruotais ISO 9000:2000 ir ISO 15504:1-9:1998 standartuose, ir sudaro šių standartų technologinį pagrindą. Šių principų laikymasis pagerina organizacijos valdymo kultūrą, sudaro palankias sąlygas kokybės valdymo sistemos panaudojimui ne tik PĮ projektuose, bet visose organizacijos veiklose ir užtikrina kuriamos PĮ produkcijos konkurencingumą.

Atsižvelgiant į PĮ projektų įgyvendinimo sudėtingumą, projekto vadovas turi susipažinti su organizacijomis, siūlančiomis PĮ projektų vykdymo gaires, standartus ar kitus išteklius. Robert T. Futrell ir kt. išskyrė septynias svarbiausias tarptautines organizacijas: Projektų valdymo institutas (angl. *PMI – Project Management Institute*), Amerikos kokybės visuomenė (angl. *American Society of Quality*), Elektros ir elektronikos institutas (angl. *IEEE – Institute of Electrical and Electronics Engineer*), Tarptautinė standartizacijos organizacija (angl. *ISO – International Organization for Standardization*), Nacionalinis Amerikos standartizacijos institutas (angl. *ANSI – American National Standards Institute*), Nacionalinis standartizacijos ir technologijų institutas (angl. *NIST – National Institute of Standards and Technology*) ir Programinės įrangos kūrimo institutas (angl. *SEI – Software Engineering Institute*).

2.3. Programinės įrangos kūrimo procesai ir metodai

ISO/IEC 12207 standarte apibrėžiami pagrindiniai ir papildomi PĮ kūrimo gyvavimo ciklo procesai. Pagrindiniams procesais laikomi:

- *įsigijimo procesas* (angl. *Acquisition*), aprašantis sistemą, programinę įrangą ar programinės įrangos paslaugą įgyjančios organizacijos veiksmus ir apima tokias veiklas: inicijavimą, konkurso paskelbimą, kontraktų sudarymą, tiekėjo stebėjimą, priėmimą ir užbaigimą;
- *tiekimo procesas* (angl. *Supply*), aprašantis sistemą, programinę įrangą ar programinės įrangos paslaugą teikiančios organizacijos veiksmus ir apima

inicijavimą, atsakymo paruošimą, kontrakto sudarymą, planavimą, vykdymą ir kontrolę, priežiūrą ir įvertinimą, pateikimą ir užbaigimą;

- *kūrimo procesas* (angl. *Development*), aprašantis programinę įrangą kuriančios organizacijos veiksmus ir apima proceso realizaciją, sistemos reikalavimų analizę, sistemos architektūros projektavimą, sistemos integravimą, testavimą, diegimą ir priėmimą.
- *eksploatavimo procesas* (angl. *Operation*), aprašantis organizacijos, eksploatuojančios sistemą realioje aplinkoje (realiems vartotojams), veiksmus ir apima procesų realizaciją, testavimą realioje aplinkoje, sistemos eksploatavimą ir vartotojų palaikymą;
- *palaikymo procesas* (angl. *Maintainace*), aprašantis organizacijos, prižiūrinčios PI, veiksmus ir apima problemų analizę, pataisymų vykdymą, priežiūrą ir demontavimą.

Pagalbiniai procesai papildo kitus procesus ir sudaro tų procesų sudėtinę dalį su atskira, specializuota, paskirtimi ir prisideda prie tų procesų sėkmės ir rezultato kokybės. Pagalbinis procesas sudaro dokumentavimo, konfigūracijų valdymo, kokybės valdymo, patikrinimo, patvirtinimo bendrojo įvertinimo, audito ir problemų sprendimo procesai.

Yra sukurta daugybė modelių, aprašančių programinės įrangos kūrimą. Galima išskirti tokius klasikinius modelius: krioklio modelis, spiralinis modelis ir prototipinis modelis. Toliau atskirai aptarsime kiekvieną modelį, pabrėždami jo pagrindinius aspektus, privalumus ir trūkumus.

„Krioklio“ kūrimo modelis (angl. *The Waterfall Development Model*), dar vadinamas kaskadiniu modeliu, buvo sukurtas Winston W. Royce 1970 metais kaip formali kodo ir pataisymo (angl. *code-and-fix*) programinės įrangos kūrimo modelio alternatyva [12]. Tai buvo pirmasis modelis, formalizuojantis programinės įrangos kūrimo fazes, pabrėžiantis išankstinius reikalavimus ir kūrimo veiklas bei dokumentų kūrimą pradinėse fazėse. Klasikinį krioklio modelį sudaro tokios fazės (6 priedas) [12]:

- *konceptijos tyrimas* (angl. *Concept exploration*) – analizuojami sisteminio lygio reikalavimai, nustatomas įvykdomumas;
- *sistemos tyrimas* (angl. *System exploration*) – ši fazė privaloma tik tuose projektuose, kuriose kuriama techninė ir programinė įranga, nes jos metu reikalingos funkcijos pritaikomos techninei ir programinei įrangai, atsižvelgiant į visos sistemos architektūrą.
- *reikalavimų analizė* (angl. *Requirements*) – nustatomos programinės įrangos reikalavimai sistemos informacijai, funkcijoms, charakteristikoms ir sąsajai;

- *projektavimas* (angl. *Design*) – projektuojama integruota techninė programinės įrangos specifikacija, apimant duomenų struktūras, programinės įrangos architektūrą, sąsajos atvaizdavimus ir procedūrinius elementus;
- *realizavimas* (angl. *Implementation*) – programinės įrangos specifikacijų vertimas į produktą, sukuriamas pirminis kodas, duomenų bazės ir dokumentacija;
- *įdiegimas* (angl. *Installation*) – sukurtos programinės įrangos diegimas, patikrinimas ir formalus kliento priėmimas veikimo aplinkoje;
- *veikimas ir rėmimas* (angl. *Operations ir support*) – vartotojo veiksmų atlikimas, nuolat vykstantis palaikymas, įskaitant techninės pagalbos suteikimą, vartotojų konsultavimą, vartotojų pakeitimų ir patobulinimų pageidavimų fiksavimą ir klaidų ar pataisymų tvarkymą;
- *palaikymas* (angl. *Maintenance*) – programinės įrangos klaidų, defektų, gedimų, patobulinimų ir pakeitimų įvykdymas;
- *naikinimas* (angl. *Retirement*) – programinės įrangos panaikinimas, sustabdant jos veikimą arba pakeičiant naują, patobulinta programinės įrangos versija.

Kaip ir kiti programinės įrangos kūrimo modeliai, „Krioklio“ modelis turi daug privalumų, kai yra pritaikomas atitinkamam projektui. Vienas iš pagrindinių privalumų yra tas, kad šis modelis leidžia tiksliau sekti projekto eigą ir anksti identifikuoti galimus neatitikimus. Šio modelio pagalba pati organizacija, kurianti programinę įrangą, tampa labiau organizuota ir valdoma. Toks struktūrinis požiūris yra ypač svarbus didelėms organizacijoms, vykdančioms didelius ir sudėtingus kūrimo projektus. Jis reikalauja, kad kiekvienos fazės metu būtų sukurta atitinkama dokumentacija, kurią vėliau galima panaudoti testuojant ir palaikant programinę įrangą.

Reikia pabrėžti tai, kad šis modelis yra lengvai pritaikomas, nes išlaiko loginį kūrimo procesų eiliškumą. Jis išsprendžia projektų sudėtingumą organizavimo pagalba, yra gerai suprantamas ir tuo pačiu metu yra kompleksinis. Be to, jis apibrėžia kokybės kontrolės procedūras, kurias naudoja projekto komanda nustatydamas programinės įrangos kokybę.

Be privalumų galima išskirti ir šio modelio naudojimo trūkumus. Vienas iš svarbesnių trūkumų yra tas, kad iš prigimties šis modelis yra tiesinės sekos pobūdžio. Tokia savybė apsunkina sugrįžimą prie dviejų ar daugiau fazių taisant problemą ar defektą ir gali padidinti viso projekto kaštus bei vykdymo laiką. Šis modelis neatspindi programinės įrangos problemų sprendimo savybės, nes fazės yra griežtai susietos su veikla, o ne su realiais darbais.

Ši metodą tikslinga naudoti, kai vartotojų poreikiai yra tiksliai nustatyti ir visi reikalavimai yra apibrėžti gyvavimo ciklo pradžioje, kitaip projekto įgyvendinimas naudojant „Krioklio“ modelį bus

labai brangus ir gali užsitęsti. Jis gali būti sėkmingai pritaikytas produktams su stabiliu apibrėžimu ir gerai suprantamomis techninėmis metodologijomis.

Spiralinis programinės įrangos kūrimo ir tobulinimo modelis (angl. *Spiral model*) buvo sukurtas Berry W. Boehm ir paskelbtas IEEE Computer 1988 metais [12]. Kitaip nei „Krioklio“ modelis, spiralinį modelį sudaro santykinai tolygios ir organizuotos programinės įrangos kūrimo žingsnių sekos (7 priedas). Be to, šis modelis yra lankstesnis už „krioklio“, nes didesnę dėmesį skiria prototipams ir rizikos valdymui.

Spiralinis modelis apima stipriąsias „Krioklio“ modelio puses ir rizikos analizę, rizikos valdymą, palaikymo ir rėmimo procesus. Jis taip pat leidžia kurti produktus, naudojant prototinius metodus arba ketvirtosios kartos kalbas ir projektavimo priemones.

Spiralinis modelį sudaro keturi kvadratai, kurie apima tikslo ir palaikymo veiklas [12]:

1. **Uždavinių, alternatyvų ir suvaržymų nustatymas.** Tokie uždaviniai kaip vykdymas, funkcionalumas, galimybė pritaikyti pakeitimus, programinės ir techninės įrangos sąsaja ir kritiniai sėkmės faktoriai yra apibrėžiami. Nustatomos alternatyvios produkto įgyvendinimo priemonės ir kiekvienos alternatyvos suvaržymai (kaina, tvarkaraštis, sąsaja, aplinkos apribojimai). Dokumentuojama rizika susijusi su naujų technologijų panaudojimu, patirties neturėjimu, nelanksčiais grafikais ir kt.
2. **Alternatyvų įvertinimas, rizikos nustatymas ir sprendimas.** Įvertinamos uždavinių ir suvaržymų alternatyvos, įvyksta rizikos nustatymas ir sprendimas.
3. **Kito lygio produkto kūrimas.** Šiame kvadrato atliekami tokie tipiniai uždaviniai kaip projektavimas, projektavimo peržiūra, kodo kūrimas, kodo patikrinimas, testavimas ir produkto paketavimas. Pradinė versija yra peržiūrima vartotojo. Po to prasideda planavimo fazė, kurios metu programinės įrangos produktas yra patobulinamas atsižvelgiant į vartotojų reagavimus. Kiekvienoje vėlesnėje versijoje formuojamas geresnis vartotojų reikalavimų supratimas. Pakeitimų laipsnis mažėja kiekvienoje naujoje versijoje.
4. **Kitos fazės planavimas.** Šios kvadrato tipinės veiksmos yra projekto plano sukūrimas, konfigūravimo valdymo, testavimo ir įdiegimo planų sudarymas.

Spiralinis modelis akcentuoja alternatyvų ir rizikos apskaičiavimo įvertinimą. Kiekvienos fazės pabaigoje atliekama peržiūra užtikrina kitos fazės pradžią be klaidų, o jei reikia, nustato reikiamus pataisus darbus. Šio modelio privalumai slypi rizikos analizės, pabrėžimo ir pritaikomumo skirtingiems gyvavimo ciklo metodams procedūrose.

1988 metais Berry W. Boehm išskyrė spiralinio modelio privalumus ir trūkumus [12, 29]. Svarbiausiais šio modelio privalumais galima laikyti jo parinktis, apjungiančias geras programinės įrangos kūrimo proceso modelių savybes, ir daugumą rizikos valdymo pagalba eliminuojamų tų metodų sunkumų. Be to, šis modelis suteikia mechanizmą, inkorporuojantį programinės įrangos kokybės tikslus į programinės įrangos produkto kūrimą. Jis taip pat koncentruojasi ties klaidų ir nepatrauklių alternatyvų pašalinimu. Spiralinis modelis suteikia sistemą, apjungiančią techninės ir programinės įrangos kūrimą, nes rizikos vertinimo principas gali būti pritaikytas tiek programinei, tiek techninei įrangai.

Iš kitos pusės, spiralinio modelio panaudojimas gali sukelti sunkumų. Jei vykdomo projekto rizika yra žema, šio modelio pritaikymas gali būti labai brangus ir užimti daug laiko. Jis yra sudėtingas, todėl gali būti sunku pritaikyti. Spiralinis modelis reikalauja aukštos rizikos vertinimo kompetencijos. Be to, didelė pereinamų etapų gausa gali sukurti papildomus vidinius ir išorinius dokumentavimo procesus.

Atsižvelgiant į pateiktus spiralinio modelio privalumus ir trūkumus, šis modelis gali būti sėkmingai panaudojamas įgyvendinant stambius projektus, kai organizacija turi pakankamai įgūdžių, reikalingų šiam modeliui įgyvendinti. Be to, šį modelį tikslinga naudoti tada, kai reikia parodyti projekto pasiekimus ir kuriamos programinės įrangos kokybę tam tikru projekto įgyvendinimo momentu.

Prototipinis programinės įrangos kūrimo modelis (angl. *Prototype software development approach*) yra vienas iš populiariausių modelių. Prototipas – tai nustatytų reikalavimų vykdymo strategija, kurios metu išskiriami, fiksuojami ir tenkinami vartotojų poreikiai naudojant galutinio programinės įrangos produkto darbinį modelį [29].

Programinės įrangos kūrimą naudojant prototipinį modelį, sudaro tokie žingsniai (8 priedas) [29]:

- reikalavimų rinkimas ir analizė;
- projektavimas;
- prototipo sukūrimas;
- vartotojų prototipo įvertinimas;
- prototipo projekto tobulinimas;
- klientų pasitenkinimo įvertinimas;
- visiškas programinės įrangos produkto sukūrimas.

Reikalavimų rinkimo ir analizės fazėje įvertinami pradiniai vartotojų poreikiai. Naudojant gautą informaciją greitai suprojektuojamas ir sukuriamas produkto prototipas. Sukurtą prototipą

įvertina vartotojai, kurie pateikia programinės įrangos kūrėjams savo atsiliepimus apie produkto silpnąsias ir stipriąsias puses. Penkto žingsnio metu prototipas yra tobulinamas atsižvelgiant į vartotojų pastabas. Po to kūrėjai turi įvertinti vartotojų pasitenkinimą ir, jei jų tikrieji poreikiai nėra patenkinti, grįžti prie penkto žingsnio. Kitu atveju sukuriamas galutinis programinės įrangos produktas.

Kaip šio modelio privalumą reikėtų paminėti tai, kad galutiniai vartotojai turi galimybę įvertinti kuriamą programinės įrangos produktą ir pamatyti, kaip jų reikalavimus supranta programinės įrangos kūrėjų. Kadangi kūrėjai turi galimybę bendradarbiauti su vartotojais visą projekto kūrimo laiką, galimų klaidų ir nesusipratimų atvejai sumažinami, o produkto kokybė padidinama. Be to, šis modelis apima rizikos kontrolę ir dokumentaciją, nukreiptą į galutinį produktą.

2.4. Programinės įrangos kokybė

Prieš programinės įrangos ir informacinių technologijų atsiradimą kokybė buvo siejama su fiziniais objektais ir sistemomis, o kokybės užtikrinimas – su gamybos procesu. Pagrindinis dėmesys buvo telkiamas ties kuriamo produkto atitikimu jo specifikacijoms. Kadangi dauguma specifikacijoje nurodytų elementų išreiškiami tokiais dydžiais kaip leistinos paklaidos, pagrindinis statistinės kokybės kontrolės tikslas buvo paklaidos mažinimas gamyboje [54].

Vystantis paslaugų industrijai pasikeitė požiūris į kokybę, nes organizacijoms reikėjo prisitaikyti prie besikeičiančių vartotojų lūkesčių ir pereiti nuo nulinių produkto trūkumų prie nulinių klientų pasitraukimų. Klientų lojalumas tapo svarbesnis už apibrėžtų standartų ir specifikacijų atitikimą.

Pasak Prahalad ir Krishnan, programinės įrangos industrija sujungia standartų ir specifikacijų atitikimo bei paslaugų kokybės požiūrius. Aukštos kokybės programinę įrangą apibrėžia trys pagrindiniai elementai: atitikimas, pritaikomumas ir naujoviškumas.

Ilgai atskiri autoriai ir ištisos organizacijos kokybės terminą apibrėždavo skirtingai. Phil Crosby 1979 metais kokybę apibrėžė kaip vartotojų reikalavimų atitikimą. Watts Humphrey, Proceso brandos modelio iniciatorius, kokybę aprašo kaip labai gero tinkamumo naudoti lygio pasiekimą. Savo ruožtu IBM kompanija kokybę apibrėžė kaip valdomą rinkos poreikių (angl. *market-driven quality*).

Programinės įrangos kokybės sąvoką vienareikšmiškai apibrėžti sunku. Jos apibrėžimas priklauso nuo daugelio aspektų – programinės įrangos tipo, naudojimo aplinkybių ir kt. Todėl iš pradžių reikia nustatyti aukštos kokybės programinės įrangos savybes ir apžvelgti skirtingus programinės įrangos kokybės požiūrius.

Išskiriami 5 pagrindiniai požiūriai į PĮ kokybę [48]:

- transcendentiniu (angl. *transcendental view*) – kokybė sunkiai apibrėžiama ar nustatoma naudojant abstrakčius terminus, tačiau jei ji yra, ją lengva atpažinti. Ji

dažnai siejama su tam tikromis nematerialiomis savybėmis, kuriomis vartotojai yra patenkinti;

- vartotojų (angl. *user view*) – kai kokybė suprantama kaip paskirties ir vartotojų poreikių atitikimas;
- gamintojų (angl. *manufacturing view*) – kai kokybė atitinka proceso standartus;
- produkto (angl. *product view*) – pabrėžiamos neatskiriamos paties produkto charakteristikos, tikintis, kad kontroliuojant šiuos pagrindinius kokybės rodiklius bus pagerinta išorinė produkto elgsena;
- vertės (angl. *value-based view*) – tai klientų noras užmokėti už programinę įrangą.

Kalbant apie programinės įrangos kokybę, žmonių, atliekančių skirtingas funkcijas ir pareigas, nuomonė skirsis.

Šiuolaikinėje konkurencingoje aplinkoje programinės įrangos gamintojai turi atsižvelgti į vartotojų poreikius, lūkesčius ir nuomonę. Pagrindiniai programinės įrangos vartotojų kokybės lūkesčiai yra, kad programinė įranga patikimai atliktų nustatytas funkcijas ir visiškai tenkintų jų poreikius. Šiuolaikinėse programinės įrangose ir sistemose daugelis vartotojų vertina jų naudojimo paprastumą.

Programinės įrangos gamintojų ir vartotojų kokybės požiūriai skiriasi. Gamintojams kokybė susijusi su sutartinių įsipareigojimų įvykdymu, sukuriant tokį produktą, kuris atitiktų apibrėžtas specifikacijas, arba suteikiant paslaugas, atitinkančias paslaugų sutartį.

Aptarti programinės įrangos kokybės požiūriai paaiškina programinės įrangos kokybės sąvoką. IEEE programinės įrangos kokybę apibrėžia kaip klientų ar vartotojų poreikių ar lūkesčių patenkinimo laipsnį.

Tarptautinis programinės įrangos įvertinimo standartas ISO-9126 (angl. *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*) numato hierarchinę kokybės apibrėžimo bazę, sudarytą iš 6 pagrindinių kokybės charakteristikų grupių ir 21 detalios charakteristikos [21]:

1. **Funkcionalumas** (angl. *Functionality*) – tai funkcijų, kurių pagalba programinė įranga patenkina nustatytus ar numatomus vartotojų poreikius ir atitinka specifikacijas, visuma. Funkcionalumo grupės charakteristikos:
 - a) tinkamumas (angl. *Suitability*) – tai tinkamų funkcijų suteikimas vartotojui nurodytiems uždaviniams spręsti;
 - b) tikslumas (angl. *Accuracy*) – tai teisingų ar sutartų rezultatų teikimas tam tikru tikslumo laipsniu;

- c) operacinis suderinamumas (angl. *Interoperability*) – taip programinės įrangos gebėjimas sąveikauti su viena ar keliomis nurodytomis sistemomis;
 - d) saugumas (angl. *Security*) – tai duomenų tikslumo ir užbaigtumo užtikrinimas bei nesankcionuotos prieigos kontrolė.
2. **Patikimumas** (angl. *Reliability*) – tai atributų rinkinys, nusakantis, kad programinė įranga nustatytą laiką išlaiko savo funkcionavimo kokybės lygį pagal griežtai apibrėžtas sąlygas. Patikimumo charakteristikos:
- a) užbaigtumas (angl. *Maturity*) – klaidų išvengimas programinės įrangos gedimo atvejais;
 - b) klaidų toleravimas (angl. *Fault tolerance*) – nustatyto veikimo lygio palaikymas PĮ gedimo ar sąsajos pažeidimo atvejais;
 - c) atkuriamumas (angl. *Recoverability*) – nurodyto veikimo lygio pakartotinis sukūrimas ir su pažeidimu tiesiogiai susijusių duomenų atstatymas.
3. **Tinkamumas** (angl. *Usability*) – tai programinės įrangos dabartinių ir galimų vartotojų poreikių atitikimo laipsnis. Tinkamumą išreiškia tokios charakteristikos:
- a) suprantamumas (angl. *Understandability*) – galimybė vartotojui suprasti, ar programinė įranga yra tinkama ir kaip ją galima panaudoti atliekant tam tikrus uždavinius nustatytais sąlygomis;
 - b) galimybė mokytis (angl. *Learnability*) – tai galimybės vartotojui išmokti dirbti su programa suteikimas;
 - c) darbingumas (angl. *Operability*) – tai vartotojo galimybė valdyti ir kontroliuoti programinę įrangą;
 - d) patrauklumas (angl. *Attractiveness*) – tai programinės įrangos gebėjimas būti patraukliam vartotojui.
4. **Efektyvumas** (angl. *Efficiency*) – tai efektyvus atitinkamų išteklių panaudojimas vykdant nustatytas funkcijas apibrėžtomis sąlygomis. Skiriamos tokios efektyvumo charakteristikos:
- a) laiko režimas (angl. *Time behaviour*) – atsakymo ir apdorojimo laiko bei pralaidumo rodiklio teikimas;
 - b) resursų panaudojimas (angl. *Resource utilisation*) – atitinkamo resursų kiekio ir tipo panaudojimas;
5. **Palaikymo galimybė** (angl. *Maintainability*) – tai programinės įrangos pakeitimo darbų atlikimo galimybė. Pakeitimus gali apimti pataisymai, patobulinimas arba programinės įrangos pritaikymas prie pakitusios aplinkos, reikalavimų ir funkcinių specifikacijų. Palaikymo galimybės charakteristikos:

- a) analizės patogumas (angl. *Analysability*) – galimybė diagnozuoti programinės įrangos trūkumus ir klaidų priežastis;
 - b) pakeitimo galimybė (angl. *Changeability*) – pakeitimų įvykdomumas;
 - c) stabilumas (angl. *Stability*) – nenumatytų pakeitimų galimų padarinių išvengimas;
 - d) testavimo galimybė (angl. *Testability*) – tai pakeitimų testavimo atlikimas.
6. **Mobilumas** (angl. *Portability*) – tai galimybė perkelti programinę įrangą iš vienos aplinkos į kitą. Aplinka gali būti organizacinė, techninė ar programinė. Į šią charakteristikų grupę įeina:
- a) adaptyvumas (angl. *Adaptability*) – prisitaikymas prie skirtingų nurodytų aplinkų be papildomų veiksmų ar pastangų;
 - b) įdiegimo galimybė (angl. *Installability*) – programinės įrangos įdiegimas kitoje nurodytoje aplinkoje;
 - c) prisitaikymas (angl. *Conformance*) – standartų ar kitų mobilumo susitarimų laikymasis;
 - d) pakeičiamumas (angl. *Replaceability*) – galimybė naudoti programinę įrangą be modifikacijos vietoje kito objekto, vykdant tam tikrus reikalavimus.

ISO 9126 standartas išskiria du programinės įrangos produkto kokybės parametrų lygius: vidinį ir išorinį. Vidinius parametrus galima pritaikyti nevykdomam PĮ produktui (specifikacijoms ar kodui) projektavimo ir programavimo etapuose. PĮ produkto kūrimo metu turi būti įvertinti tarpiniai rezultatai, naudojant vidinius parametrus, kurie matuoja esmines PĮ savybes. Pagrindinis vidinių parametrų tikslas yra užtikrinti būtinos išorinės ir naudojimo kokybės pasiekimą. Šie parametrai suteikia vartotojams, vertintojams, testuotojams ir projektuotojams akivaizdžios naudos: jie gali įvertinti PĮ kokybę ir atkreipti dėmesį į kokybes problemas prieš PĮ sukūrimą.

Išoriniai parametrai naudoja sistemos funkcionalumo PĮ įvertinimus testuojant, vykdant ir stebint vykdomąją PĮ ar sistemą. Išorinių parametrų privalumas – PĮ kokybės įvertinimas testavimo ir veikimo metu.

Programinės įrangos kokybės vertinimo parametrus galima grupuoti kitu aspektu: produkto, proceso ir projekto[29].

Produkto parametrai nusako tokias produkto charakteristikas kaip PĮ dydis, sudėtingumas, veiklos efektyvumas ir kokybės lygis. Proceso parametrus galima panaudoti gerinant PĮ kūrimą ir palaikymą, pavyzdžiui, trūkumų pašalinimo efektyvumas kūrimo metu, pataisymo proceso atsakymo laikas. Projekto parametrai aprašo projekto charakteristikas ir jo vykdymą, pavyzdžiui, PĮ kūrėjų skaičius, kaštus, grafiką ir produktyvumą. Kai kurie parametrai priklauso kelioms grupėms.

Išskiriamos tokios produkto kokybės metrikos [29]:

- vidutinis gedimų intervalas (angl. *mean time to failure*);
- defektų tankis (angl. *defect density*);
- problemos, su kuriomis susiduria klientai;
- klientų pasitenkinimas.

Vidinė produkto kokybė dažnai matuojama programinės įrangos klaidų (funkcinių defektų) skaičiumi (defektų tankis) arba programinės įrangos veikimo be avarijos laiku (vidutinis gedimų intervalas). Vidutinio gedimų intervalo parametras dažnai naudojamas sistemose su padidintu saugumo laipsniu, pavyzdžiui, oro transporto kontrolės sistemose, aviacijos elektronikoje. Defektų tankio parametras naudojamas daugumoje komercinėse programinės įrangos sistemų.

Kitas produkto kokybės parametras, naudojamas pagrindinių programinės įrangos kūrėjų, matuoja problemas, su kuriomis susiduria klientas naudodamas produktą. Šis parametras apima ne tik galiojančius defektus, bet ir naudojimo problemas, neaiškią dokumentaciją ar informaciją ir net vartotojų klaidas. Šios problemos kartu su defektų problemomis kliento požiūriu sudaro visą programinės įrangos problemų sritį.

Kliento problemos parametras galima laikyti kaip tarpinį matą tarp defektų matavimo ir klientų pasitenkinimo. Mažinant klientų problemas, būtina sumažinti funkcinis programinės įrangos defektus ir pagerinti kitus faktorius, pavyzdžiui, tinkamumą, dokumentavimą. Gerinant klientų pasitenkinimą, reikia mažinti defektus ir bendras problemas bei valdyti platesnio užmojo veiksniai, tokius kaip produkto naudingumas, organizacijos įvaizdis, aptarnavimas ir kt. Viso produkto ir jo atskirų aspektų kokybės pasitenkinimo parametras dažnai gaunamas per įvairių klientų tyrimų metodus.

Nors proceso kokybės parametrai formaliai sunkiai apibrėžiami, jie yra ne mažiau svarbūs už galutinio produkto parametrus. Proceso kokybės parametrai gali kisti priklausomai nuo organizacijos požiūrio. Kai kurioms organizacijoms proceso kokybės parametrai paprasčiausiai reiškia defektų atsiradimo stebėjimą įrangos testavimo metu, kitos organizacijos šiuos parametrus sieja su kiekviena kūrimo ciklo faze.

Programinės įrangos kokybės valdymas pritaikomas visiems procesų, produktų ir išteklių aspektams. Programinės įrangos kokybės valdymas apibrėžia jų, procesų valdytojus, procesų reikalavimus, procesų ir jų rezultatų pokyčius bei grįžtamojo ryšio kanalus.

Kokybės valdymo procesus sudaro daug veiksmų. Kai kurie iš jų padeda tiesiogiai surasti defektus, kiti padeda nustatyti, kur reikėtų atlikti detalesnius tyrimus.

Programinės įrangos kokybės planavimas apima galutinio produkto apibrėžimą kokybės charakteristikų terminais ir reikalingo produkto sukūrimo procesų planavimą. Šie procesai skiriasi nuo

programinės įrangos kokybės valdymo, nes jie pirmiausia nukreipti į planuojamų kokybės charakteristikų vertinimą, o ne į šių planų įvykdymą.

Kai kurie specialūs programinės įrangos kokybės valdymo procesai nusakyti ISO 12207 standarte. Tai:

- kokybės užtikrinimo procesas;
- patikrinimo procesas;
- patvirtinimo procesas;
- bendros analizės procesas;
- audito procesas.

Visi šie procesai palaiko kokybės užtikrinimo siekius ir padeda galimų klaidų paieškos metu. Jie taip pat padeda užtikrinti geresnės kokybės programinės įrangos sukūrimą.

Programinės įrangos kokybės užtikrinimo procesai patvirtina, kad programinių produktų ir projekto gyvavimo ciklo procesas atitiktų nustatytus reikalavimus. Kokybės užtikrinimas remiasi veiksmy, nukreiptų į tai, kad kokybė taptų neatskiriama programinės įrangos dalimi, planavimo ir įvykdymo. Toks požiūris aiškiai ir tiksliai nustato problemą ir jos sprendimo reikalavimus. Programinės įrangos kokybės užtikrinimas leidžia užtikrinti kokybę kūrimo ir palaikymo metu, vykdant atitinkamus veiksmus visais gyvavimo ciklo etapais. Tai leidžia identifikuoti problemas ankstyvose stadijose.

Programinės įrangos kokybės užtikrinimo užduotis – atitinkamų procesų planavimas, tolesnis procesų vykdymas, remiantis nustatytu planu, ir reikalingų procesų pakeitimų atlikimas. Programinės įrangos kokybės užtikrinimo planas numato priemones, kurios bus panaudotos užtikrinant kuriamo produkto atitikimą nustatytiems vartotojų reikalavimams ir aukštą kokybės lygį. Norint tai pasiekti, kokybės tikslai pirmiausia turi būti tiksliai apibrėžti ir suprantami. Šis planas turi būti suderintas su programinės įrangos konfigūracijos valdymo planu.

Kokybės užtikrinimo planas identifikuoja dokumentus, standartus, praktikas ir susitarimus, naudojamas projekto kontrolės metu. Be to, planas nustato metrikas, statistines technikas, pranešimų apie iškilusias problemas formavimo procedūras ir atitinkamus taisymo veiksmus.

Programinės įrangos kokybės užtikrinimą galima pasiekti šiam tikslui skirtų tarptautinių modelių ir standartų pagalba. Tokiems modeliams galima skirti ISO 9000 standartų šeimą, CMM/CMMI, TickIT, SPICE, Trillium ir kt.

2.5. Programinės įrangos kokybės įvertinimas

Pagrindinis kokybės užtikrinimo proceso stebėjimo tikslas – užtikrinti tinkamą planuotų veiksmų įvykdymą naudojant įvairius matavimus. Kokybės įvertinimo metu gauti matavimai suteikia pradinius duomenis vėlesnės analizės ir modeliavimo veiksmams.

Kokybės tinkamumo aspektas dažnai gali būti tiesiogiai gautas iš jo apibrėžimo arba gali būti susijęs su įvairių defektų matavimais. Pavyzdžiui, sistemos klaidų tikimybė siejama su patikimumo charakteristika. Tokie produktų defektų matavimai kaip bendras sistemos klaidų skaičius gali būti panaudoti nustatant sistemos kokybę. Išoriniai pramonės duomenys ir susiję kokybės modeliai taip pat gali būti naudojami apytiksliai vertinant produkto kokybę, tiesiogiai net nevertinant pačios sistemos. Tačiau tokius apytikslius matavimus galima naudoti tik kaip pradinį kokybės planavimo tašką, kai dar nėra faktinių projekto matavimo duomenų.

Tiesioginis kokybės matavimas paprastai reikalauja įvertinti atskirus kokybės užtikrinimo veiksmus ir susijusius defektus. Rezultatų ir defektų matavimai kartais gali būti atlikti atskirai nuo kitų matavimų. Pavyzdžiui, nemažai programinės įrangos kūrimo organizacijų stebi kokybės užtikrinimo veiksmus per atskleistus defektus, neatsižvelgiant į šių defektų dalį visuose defektuose. Visų defektų skaičių galima apskaičiuoti naudojant įvairius programinės įrangos įvertinimo metodus. Tokius tiesioginius kokybės matavimus galima naudoti atskirai pasirinktiems kokybės modeliams.

Tam tikrais atvejais tiesioginius kokybės matavimus reikia naudoti kartu su kitais kokybės modeliavimo netiesioginiais matavimais, stebint kokybės užtikrinimo ir programinės įrangos procesus, bei teikti analizės metu gautus duomenis.

Dauguma rezultatų ir defektų matavimų gaunami tik vėlesniuose programinės įrangos kūrimo procesuose. Be to, problemų sprendimas vėlesniuose etapuose yra finansiškai neefektyvus, nes kuo ilgiau defektai liks sistemoje, tuo daugiau žalos bus padaryta. Taigi labai svarbu naudoti kitus pirmalaikius indikatorius ir susijusius kokybės modelius, prognozuojant kokybę. Tokie netiesioginiai kokybės indikatoriai paprastai gali būti išmatuoti naudojant aplinkos, vidinius produkto ir veiksmų matavimus. Daugumą šių tiesioginių ir netiesioginių kokybės matavimų galima gauti iš kokybės užtikrinimo veiksmų arba iš viso programinės įrangos kūrimo ar palaikymo procesų. Tokie matavimai klasifikuojami kaip procesų matavimai.

Norint tinkamai rinkti ir naudoti įvairius matavimo duomenis, reikia atkreipti dėmesį į:

- nuolatinę interpretaciją ir stebėjimą. Defektams nustatyti išskiriamos vykdymo klaidos, vidiniai gedimai ir žmonių veiksmų klaidos. Tokias problemas reikia nuolat sekti ir skaičiuoti. Panašiai nuolatinę interpretaciją reikalauja išlaikyti kiti matavimai;
- savalaikes defektų ir duomenų ataskaitas. Kadangi šie matavimai naudojami stebėti ir kontroliuoti programinės įrangos projektus, tai, norint palaikyti informacijos aktualumą, reikia užtikrinti pranešimų apie defektus ir kitus dinامينius matavimus savalaikiškumą;

- teisingą duomenų skaidymo gylį. Skirtingi kokybės analizės modeliai gali reikalauti skirtingų duomenų skaidymo gylių lygių.

Svarbi kiekvieno kokybės užtikrinimo veiksmo dalis yra rezultatų analizė, problemų sprendimas ir atitinkamų koregavimo veiksmų atlikimas. Unikalus kiekvieno defekto nustatymas palengvina jo stebėjimą ir sprendimą. Kita informacija apie defektą gali būti renkama ir atnaujinama defektų sprendimo proceso metu, būtent:

- išsami informacija apie defekto nustatymo aplinkybes;
- tiksli pataisytų ar rastų trūkumų vieta.

Kiekvienas defektas turi būti įvertintas ir su juo susijusi informacija, kartu su kita projekto informacija, panaudota analizės metu, taikant įvairius kokybės įvertinimo modelius. Tokie modeliai suteikia tokius kokybės įvertinimus, kaip defektų tankumas ir pasiskirstymas, produkto patikimumas, kokybės užtikrinimo veiksmų eigos informacija ir kt. Dažnai papildoma informacija apie defektus renkama (defektų tipas, sunkumas) tolimesnei analizei.

Aplinkos matavimai skiriami prie bendrų proceso, produkto ir žmonių savybių [41] programinės įrangos produkto hierarchijoje. Proceso savybės apima naudojamą procesą (krioklio, spiralinis, prototipinis), veiksmus ir jų ryšius, specifinius kūrimo metodus. Prie žmonių savybių skiriami įgūdžiai ir patirtis, išsipareigojimai ir funkcijos, organizacijos struktūra ir projekto komandos sudėtis. Produkto savybes sudaro pagrindiniai galutinių vartotojų lūkesčiai, aukšto lygio produkto funkcionalumas, produkto marketingo aplinka ir specifinės techninės ir programinės įrangos konfigūracijos.

Vidiniai produkto matavimai apibūdina įvairias pasirinktos programinės įrangos vidines produkto savybes. Šie matavimai apima [41]:

- programinės įrangos artefaktų (angl. *software artifacts*) išmatavimą, įskaitant programinės įrangos reikalavimų technines sąlygas, projektus, programinius kodus, testavimo atvejus, susijusius dokumentus ir kitus programinės įrangos artefaktus;
- produkto vidinių savybių išmatavimus, įskaitant kontrolę, duomenis ir pateiktis;
- struktūrų išmatavimus. Skirtingas programinės įrangos artefaktų savybes galima traktuoti kaip nestruktūrizuotų simbolių aibę arba kaip sintaksines struktūras, netgi kaip susijusias semantines esybes.

Dauguma vidinių produktų matavimų gaunami analizės ir projektavimo etapuose, todėl jie dažnai naudojami įvairiuose modeliuose įvertinant produkto kokybę ir nustatant problematines sritis kokybės tobulinimo metu. Veiksmų matavimai tiesiogiai matuoja specifinius programinės įrangos

kūrimo ir palaikymo veiksmus ir susijusias pastangas, laiką ir kitas sąnaudas. Skirtingai nuo aplinkos ir vidinių produkto matavimų, didelė veiksmų matavimų dalis yra dinaminė. Šio tipo matavimus galima atlikti skirtinguose skaidymo gylio lygiuose.

Programinės įrangos kokybei įtaką daro įvairios vidinės produkto savybės, programinės įrangos ir jų vartotojų sąveika bei bendros produkto, kūrimo proceso ir aplinkos savybės. Be to, tikslūs tiesioginiai kokybės matavimai gali būti atlikti tik projekto pabaigos etapuose, o netiesioginiai kokybės matavimai pateikia kokybės prognozės rezultatus pradiniuose projekto etapuose. Todėl svarbu šiuos matavimo tipus sujungti kokybės įvertinimo modeliuose. Kokybės įvertinimo modeliai skirstomi į dvi kategorijas, atsižvelgiant į produkto matavimus ir rezultatus: apibendrinamieji (angl. *generalized models*) ir specifiniai produkto modeliai (angl. *product-specific models*).

Apibendrinamieji modeliai teikia apytikrius produkto kokybės įvertinimus gamybos vidurkių ar bendrų aplinkos tendencijų forma, nenaudojant duomenų susijusių su konkrečiu produktu. Skiriami tokie apibendrinamųjų modelių tipai [41]:

- bendrasis modelis (angl. *overall model*) – pateikia vieną bendrą produkto kokybės įvertinimą;
- segmentinis modelis (angl. *segment model*) – pateikia skirtingus kokybės įvertinimus atskiriems gamybos segmentams;
- dinaminis modelis (angl. *dynamic model*) – pateikia kokybės tendenciją ar pasiskirstymą laike ar programinės įrangos kūrimo etapuose.

Specifiniai produkto kokybės įvertinimo modeliai teikia tikslesnius kokybės įvertinimus, kadangi panaudoja specifinius produkto duomenis. Specifiniai produkto modeliai apima:

- poreikiams pritaikyti modelius (angl. *semi-customized models*), kurie nustato vykdomo projekto kokybę, naudojant bendras produkto, proceso ar aplinkos savybes ir istorinius duomenis. Šių modelių rezultatinė informacija gali būti tiesiogiai panaudota nustatant produkto defektų pasiskirstymą;
- stebėjimo modelius (angl. *observation-based models*), kurie įvertina kokybę, naudojant vykdomo projekto stebėjimo duomenis;
- matavimo pagrindu prognozuojamus modelius (angl., *measurement-driven predictive models*), kurie nustato prognozuojamus ryšius tarp įvairių pradinių matavimų ir produkto kokybės, remiantis istoriniais duomenimis, teikia pirmalaikės kokybės prognozes ir nustato problemas, kurias galima išspręsti pradiniuose projekto etapuose ir taip pagerinti produkto kokybę.

Skirtingus kokybės vertinimo modelių tipus ir jų ryšius galima paliginti įvertinant jų gebėjimą teikti naudingą informaciją, galimybę juos pritaikyti skirtingoms projekto aplinkoms ir jų tarpusavio ryšius. Kitaip sakant, galima paliginti [41]:

- modeliavimo rezultatų *naudingumą*, kokybės įvertinimo *tikslumą* ir modelio *pritaikymą* skirtingoms aplinkoms;
- modelio *vidinius ryšius*, kurios galima nagrinėti dviem skirtingomis kryptimis:
 - apibendrinamų kokybės modelių pritaikymo galimybės , kai gaunama specifinė informacija apie produktą;
 - specifinių produktų modelių apibendrinimas, kai užtenka skirtingų produktų ar projektų empirinių duomenų.

Kokybės įvertinimo modeliai duomenims kelia skirtingus reikalavimus. Aptarti modeliai naudoja tiesioginius kokybės matavimus. Kai kurie paminėti modeliai įvertinant kokybę gali naudoti ir netiesioginius kokybės matavimus.

3. KOKYBĖS VADYBOS METODAI IR JŲ TAIKYMAS

Įgyvendinant projektus, organizacijoje projekto vadovas turi atsižvelgti į organizacijos kokybės politiką. Organizacijos kokybės politika turi būti nukreipta į organizacijos veiklos, procesų, produktų kokybės ir kokybės vadybos sistemos kokybės nuolatinį tobulinimą. Šią nuolatinio tobulinimo idėją nuo šeštojo dešimtmečio aktyviai nagrinėjo daugelis specialistų. Šiuo metu nuolatinio kokybės gerinimo reikalavimas pripažintas svarbiu organizacijos konkurencingumo pasiekimo ir išlaikymo būdu. Kokybės gerinimas turi būti įtrauktas į pačios organizacijos struktūrą ir pobūdį, jis turi tapti ne tik organizacijos, bet ir kiekvieno jos dalyvio tikslu.

Šiam tikslui pasiekti skirti įvairūs praktiniai metodai: ISO 9000 kokybės standartai, Visuotinės kokybės vadyba, Šeši sigma kokybės metodologija, Edwardo Demingo kokybės vadybos principas, Kauru Ishikawa priežasčių ir pasekmių metodas, Genichi Taguchi kokybės siekimo metodas ir kt.

Darbe išsamiau aprašyti trys dažniausiai naudojamų metodai: ISO 9000:2000 kokybės standartas, Šeši sigma kokybės metodologija ir visuotinės kokybės vadyba.

3.1. Kokybės vadybos metodų apžvalga

ISO 9000 standartų šeima buvo įdiegta, kai devintojo dešimtmečio viduryje pradėjo formuotis Europos Sąjunga, nes reikėjo pasirūpinti standartais, kurie tarp ES narių palengvintų prekybą. Vėliau ISO 9000 tapo tarptautiniu standartu. ISO 9000 – tai verslo valdymo sistema, kurią pradėjo rengti tarptautinė standartizacijos organizacija (angl. *The International Organization for Standardization – ISO*), įkurta Ženevoje, Šveicaroje ir apimanti apie 100 pasaulio industrinių valstybių. Šie standartai pasaulinėje rinkoje pripažinti kaip kokybės sistemų reikalavimų pagrindas.

ISO 9000 kokybės sistemos standartų tikslas – skatinti kokybišką mąstymą ir naudotis geriausia patirtimi [17]. Šie standartai sudaro darnią kokybės vadybos sistemos standartų grupę, padedančią kurti savitarpio supratimą nacionalinėje ir tarptautinėje prekyboje.

Iki 2000 m. ISO 9000 standartai apėmė:

- visus tarptautinius standartus, turinčius ISO 9000 – 9004 numerius, tai pat visus ISO 9000 ir ISO 9004 standartų skyrius, kurie gali būti modifikuoti atskirai (ISO 9000-3, ISO 9000-4, ISO 9004-2, ISO 9004-3, ISO 9004-4);
- visus tarptautinius standartus, turinčius ISO 10001 – 10020 numerius, bei jų skyrius;
- ISO 8402, kai kurias atvejais – kitus standartus, nusakančius specifinę veiklą.

2000 m. šie standartai buvo pertvarkyti ir sujungti į ISO 9000 (apima ISO 8402 ir ISO 9000-1), ISO 9001 (apima ISO 9001, ISO 9002 ir ISO 9003) ir ISO 9004 (apima ISO 9004-1) bei pavadinti standartų šeima. Greta šių pagrindinių kokybės vadybos šeimos standartų naudojami anksčiau išleisti

ar atnaujinti standartai. ISO 9000:2000 standartų šeimos negalima laikyti nuoseklią, ankstesnių standartų evoliuciją, nes jie remiasi visiškai kitais, naujais principais, tai Europos kokybės vadybos fondo tobulinimo modelis, apimantis tokius principus: dėmesys vartotojui; lyderystė ir tikslo pastovumas; darbuotojų kompetencijos gerinimas ir dalyvavimas tobulinimo procese; procesais ir faktais besiremianti vadyba; nuolatinis mokymasis, inovacija ir tobulinimas; partnerystės plėtra; atsakomybė visuomenei; orientavimasis į rezultatus.

ISO 9000:2000 versijos standartai nustato kokybės vadybos sistemos reikalavimus ir gali būti naudojami organizacijose, kurių tikslas – užtikrinti kliento patenkinimą, vykdant nustatytus reikalavimus, taip pat sertifikavimo organizacijose ir vidaus auditoriaus darbo metu.

Yra trys pagrindiniai ISO 9000:2000 šeimos standartai [58]:

- **ISO 9000:2000** kokybės vadybos sistemos. Pagrindai, terminai ir apibrėžimai. Čia aprašomi kokybės vadybos sistemų pagrindai, pateikiami kokybės vadybos sistemų terminai ir apibrėžimai.
- **ISO 9001:2000** kokybės vadybos sistemos. Reikalavimai. Šiame standarte pateikti kokybės vadybos sistemos reikalavimai, kai organizacija :
 - siekia įrodyti savo gebėjimą nuolat teikti produktus, atitinkančius vartotojo ir atitinkamų reglamentuojančių teisės aktų reikalavimus;
 - rezultatyviai taikydama šią sistemą, įskaitant jos nuolatinio gerinimo procesus bei vartotojų ir atitinkamų reglamentuojančių teisės aktų reikalavimų užtikrinimą, siekia kuo didesnio vartotojo pasitenkinimo;
- **ISO 9004:2000** kokybės vadybos sistemos. Veiklos gerinimas. Rekomendacijos. Šis standartas pateikia rekomendacijas kokybės vadybos sistemų rezultatyvumui ir efektyvumui didinti. Jo paskirtis – organizacijos veiklos gerinimas ir klientų bei kitų suinteresuotų šalių patenkinimo didinimas apimant platesnę kokybės vadybos sistemos perspektyvą.

ISO 9000 standarte nusakomi aštuoni kokybės vadybos principai, kuriais turi vadovautis organizacijos vadovai [34]:

- orientavimasis į vartotoją;
- lyderystė;
- darbuotojų įtraukimas;
- procesinis požiūris;
- sisteminis požiūris į vadybą;
- nuolatinis gerinimas;
- faktais pagrįstų sprendimų priėmimas;
- abipusiai naudingi ryšiai su tiekėjais.

ISO 9001:2000 standartas nustato reikalavimus kokybės vadybos sistemai ir gali būti taikomas organizacijos vidaus poreikiams, sertifikuoti arba sutartims sudaryti. Standartas skirtas kokybės vadybos sistemos rezultatyvumui didinti tenkinant vartotojo reikalavimus. Šio standarto esmė – procesinio požiūrio taikymas visiems organizacijos atliekamiems darbams.

ISO 9001:2000 ir ISO 9004:2000 standartų leidimai parengti kaip bendra kokybės sistemų standartų pora su panašia struktūra. Šie standartai ne tik papildo vienas kitą, bet gali būti naudojami ir atskirai [35].

ISO 9004:2000 pateikia rekomendacijas didesnei kokybės vadybos sistemos tikslų apimčiai nei ISO 9001:2000, ypač bendrosios organizacijos veiklos nuolatiniam gerinimui ir efektyvumo bei rezultatyvumo didinimui. ISO 9004 rekomenduojamas organizacijoms, kurių vadovai siekia tenkinti ne tik ISO 9001:2000 reikalavimus, bet ir nuolat gerinti savo veiklą. Tačiau jis netaikytinas sertifikuojant ar sudarant sutartis. ISO 9004:2000 standarte nusakomas organizacijos sėkmės pagrindas – „įgyvendinta ir prižiūrima vadybos sistema, sukurta nuolat gerinti organizacijos veiklos rezultatyvumą ir efektyvumą, atsižvelgiant į visų suinteresuotų šalių poreikių tenkinimą“ [36]. ISO 9004 standarte taip pat pabrėžiama neatitiktį valdymo svarba. Organizacijos darbuotojai bet kuriame proceso etape turi pranešti vadovybei apie neatitiktį, kad būtų užtikrintas savalaikis neatitikties nustatymas ir pašalinimas. Informaciją apie neatitiktis turi analizuoti įgalioti darbuotojai.

Remdamasi baziniu kokybės vadybos sistemos modeliu, pateikto šiuose standartuose, organizacija gali nustatyti projektavimo, gamybos, produkto ar paslaugos tiekimo procesus. Kad būtų užtikrintas organizacijos procesų efektyvumas, nuolat turi būti atliekami vartotojų ir kitų suinteresuotų šalių patenkinimo tyrimai, vidaus auditas, finansinių rodiklių matavimai ir savęs įvertinimas [36, 49].

Šeši sigma metodologija yra aukščiausios kokybės siekimo metodologija, kokybės filosofija. Šią metodologiją 1990 m. pradžioje pirmą kartą palaikė ir pritaikė savo veikloje *Motorola* kompanija. Nuo to laiko kompanijos, įgyvendinusios šeši sigma kokybės koncepciją, yra žinomos kaip vienos iš sėkmingiausių (*General Electric, Allied Signal, Polaroid, Sony, Honda, American Express* ir kt.) ir jų dėka ši verslo strategija pradėjo skleistis visame pasaulyje.

Šešių sigma sistemos tikslas dvejopas [17]:

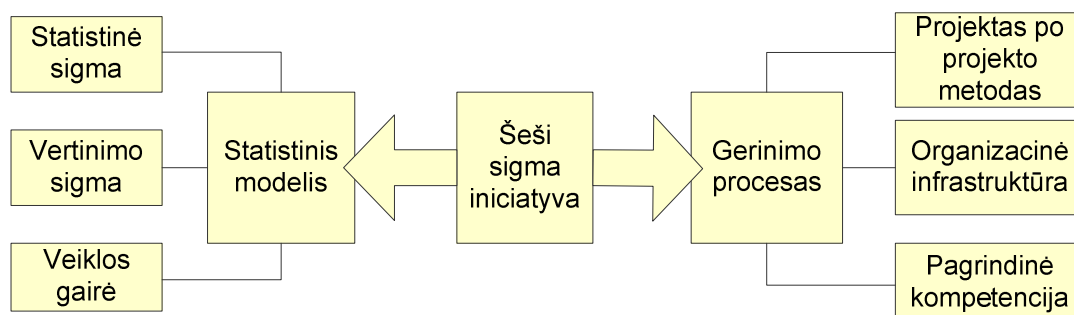
- identifikuoti visus vertinimus ir matavimus, kurie susietų pagrindinių procesų matavimo priemonės su bendrovės pelningumu ir padarytų galimybes tokias matomas, kad būtų neįmanoma jų ignoruoti;
- paspartinti verslo veiklos gerėjimą.

Šeši sigma metodologiją suformavo šios *Motorola* kompanijos iškeltos sąvokos [40]:

- Pabrėžiama, kad defektų skaičius produkcijos vienetui ir defektų skaičius vienam milijonui įvykių yra standartiniai matai, pritaikomi visiems bet kokios organizacijos aspektams.
- Atliekami intensyvūs projekto komandos apmokymai rentabilumo kėlimo, neproduktyvumo veiksmų sumažinimo ir gamybinio ciklo sutrumpinimo sumetimais.
- Darbo komandos veiklai paremti naudojami kooperaciniai rėmėjai, užtikrinantys mechanizmus, kurių pagalba darbo komanda lengviau prisiderina prie pokyčių ir gauna papildomų resursų, ir kontroliuojantys, kad komanda neišeitų už organizacijos strateginių tikslų ribų.
- Ruošiami aukštos kvalifikacijos verslo procesų gerinimo ekspertai (dar vadinamų „juodas diržas“, „žalias diržas“ ir „geltonas diržas“), mokantys taikyti kokybines ir kiekybines priemones strateginiams organizacijos tikslams pasiekti.
- Procesų pakeitimų priemonės nustatomos ir planuojamos iš anksto, o pakeitimai atspindi verslo rezultatus prieš ir po atliktų pakeitimų.
- Apmokytų ir sertifikuotų ekspertų, vadovaujančių darbo komandai, priėmimas į darbą nuo vienerių iki trejų metų.

Terminas „šeši sigma“ yra perkeltas iš matematinės statistikos, kur sigma (σ) reiškia standartinį nuokrypį nuo matuojamų kokybinių parametru vidurkio. Ši koncepcija sujungia pačius efektyviausius statistinius ir nestatistinius metodus, siekiant visapusiškai pagerinti įmonės veiklą.

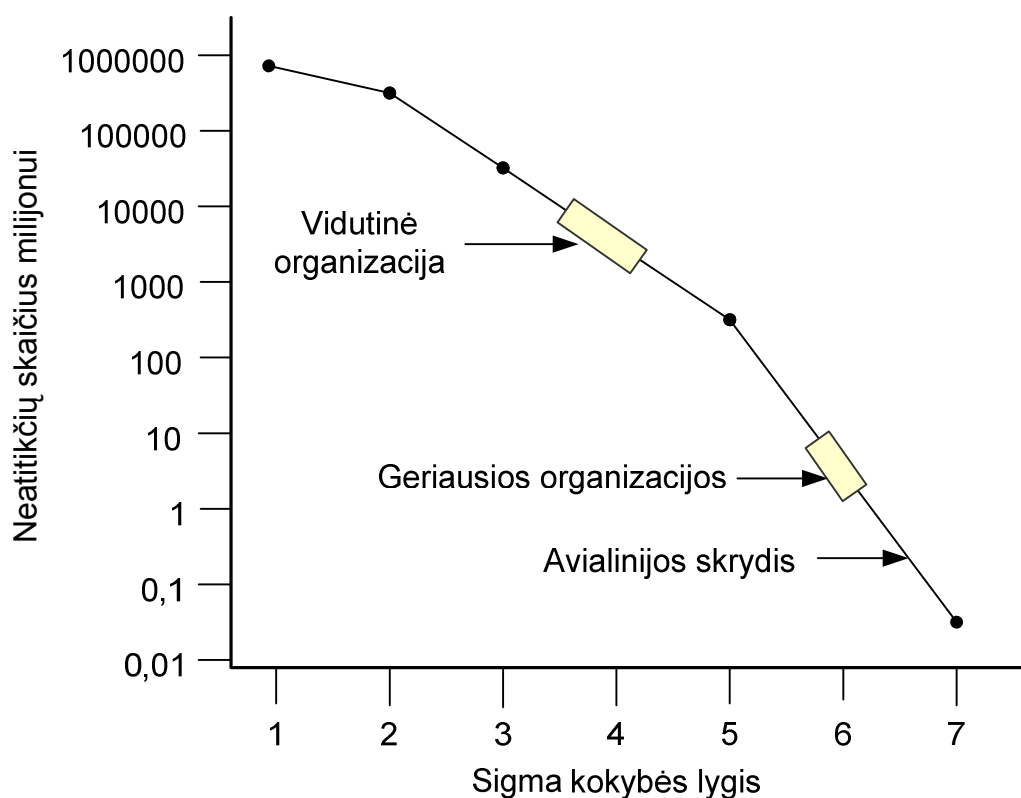
Šeši sigma verslo iniciatyvą sudaro statistinis modelis ir nuolatinio kokybės gerinimo procesas [57]. Statistinis modelis aprėpia statistinę sigma, vertinimo sigma ir veiklos gaires, o nuolatinis gerinimas – „projektas po projekto“ metodą, organizacinę infrastruktūrą ir pagrindinę kompetenciją (7 pav.).



7 pav. Šeši sigma verslo iniciatyvos aspektai

Šeši sigma metodas yra sudarytas laikantis griežtos struktūros ir remiantis pagrįstais kokybės vadybos principais ir metodika, kuriais siekiama sukurti produktą ar paslaugą geriau, greičiau ir pigiau už konkurentus. Kiekviename projekte visada gali atsirasti tam tikras kintamumas, kuris gali pasireikšti bet kuriame procese. Vadyboje svarbu žinoti, dėl kokių priežasčių toks kintamumas atsiranda. Jei įmanoma, reikia sukurti kiekybinę kintamumo matavimo sistemą ir matuoti pokyčių dydžius, kad būtų galima imtis tam tikrų koregavimo veiksmų. Siekiant šeši sigma kokybės, visų pirma reikia panaikinti neatitinkčių priežastis, kol dėl jų neatsirado neatitinkčių. Neatitinkčių nustatymas ir ištaisymas reikalauja papildomų išteklių.

Tradiciskai naudojamos procesų kintamumo matavimo sistemos pagrįstos trijų arba keturių sigma kokybe (8 pav.), o tai reiškia, kad yra leidžiama nuo 6210 (keturi sigma) iki 66810 (tris sigma) neatitinkčių vienam milijonui atveju. Šeši sigma kokybė reiškia ne daugiau nei 3,4 neatitikties atveju milijonui vienetu. Literatūroje galima rasti teorinių straipsnių, kuriuose teigiama, kad greitai bus praktiškai pasiekta septynių ar net aštuonių sigma rezultato.



8 pav. *Sigma kokybės lygių grafikas*

Šeši sigma koncepciją plačiausiai naudojama kuriant produktą, gamyboje ar teikiant paslaugas. Tačiau ji tinka beveik visiems organizacijose vykstantiems procesams tobulinti (pavyzdžiui, finansiniame organizacijos procese). Taikant šeši sigma metodologiją projektų vadyboje, reikia atsižvelgti į tai, kad standartinį šeši sigma projekto gyvavimo ciklą sudaro tokie etapai [57]:

- projekto identifikavimas;
- projekto nustatymas;
- esamo proceso veiklos įvertinimas;
- esamo proceso analizė;
- patobulinimų kūrimas, išbandymas ir patikrinimas;
- pakeitimų įgyvendinimas;
- kontrolės vykdymas;
- įgytų žinių perdavimas, sprendimo perkėlimas į panašias sritis.

Sėkmingas šeši sigma kokybės metodologijos įdiegimas priklauso nuo tinkamo jos supratimo, vadovų pritarimo ir efektyvios organizacijos struktūros. Šeši sigma – tai daugiau nei tam tikra kokybės filosofija, nes į ją žiūrima kaip į visuotinę verslo proceso tobulinimo metodologiją. Nors jos matavimo vienetu gali būti defektų skaičius vienam milijonui įvykių, tačiau pagrindiniai tikslai išaiškėja būtent šalinant neproduktyvius įvykius, mažinant gamybinio ciklo laiką ir didinant pelną. Efektyvus šeši sigma pritaikymas organizacijoje sukuria tokią valdymo sistemą, nustatančią proceso komponentus ir ribas; identifikuojančią ir bendrai naudojančią proceso valdytojus, vidinius ir išorinius klientus ir kitus dalyvius; sukuriančią efektyvaus projekto valdymo aplinką, kurioje verslas pasiekia maksimalaus pelno; sukuriančią projekto vertinimo kriterijus ir jų tinkamą dokumentavimą.

Visuotinės kokybės vadyba – tai tokia vadybos filosofija ir metodai, kuriuos pasirinkusi organizacija nuolat tobulėja, įtraukdama į tobulinimo veiklą darbuotojus ir siekdama kuo geriau patenkinti vartotojų poreikius, gerindama produktų kokybę ir mažindama kaštus [58]. Pasak Walsho, visuotinės kokybės vadyba – tai organizacinės kultūros sukūrimas, kurioje naudojant integruotą įrankių, metodų ir mokymų sistemą, skatinamas ir palaikomas nuolatinis klientų poreikių patenkinimo siekimas [50].

Visuotinės kokybės vadybos metodologija remiasi vartotojų poreikių patenkinimo būtinybe: darbas yra atliktas kokybiškai tik tada, kai jo rezultatu yra patenkintas vartotojas. Ji apima visos kompanijos (žmonių, sistemų, užduočių) kokybę.

Organizacijos, įgyvendinusios visuotinę kokybę, yra dinamiškos, sugebančios adaptuotis prie besikeičiančių išorinių ir vidinių sąlygų bei sėkmingai reaguoti į nenumatytus pasikeitimus.

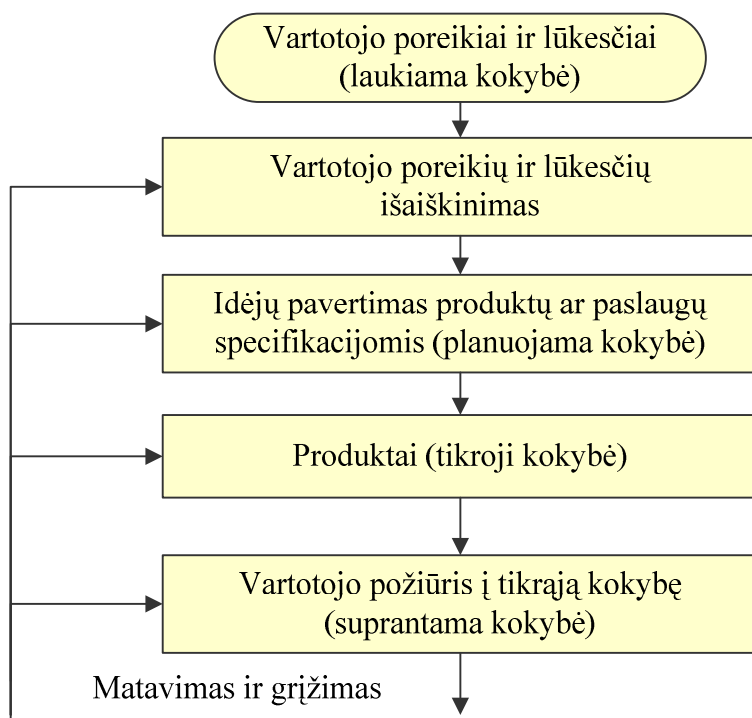
Skiriami tokie pagrindiniai visuotinės kokybės principai:

- orientavimasis į vartotoją ir procesą;
- esamų ir numatomų vartotojų poreikių tenkinimas mažiausiai kaštais;
- nuolatinis kokybės tobulinimas;
- visų darbuotojų ir aukščiausios vadovybės dalyvavimas tobulinimo procese.

Vienas iš esminių visuotinės kokybės ir kitų kokybės vadybos metodologijų yra tas, kad visuotinės kokybės vadyboje išskiriamos dvi vartotojų grupės [58]:

- **išoriniai vartotojai** – klientai, kurių poreikio tenkinimas yra vienas iš pagrindinių organizacijos veiklos tikslų. Atsižvelgiant į išorinių vartotojų poreikius ir norus, yra tobulinamas produktas ir jo specifikacija iki tokio lygio, kad vartotojų esami ir numatomi poreikiai būtų visiškai patenkinti;
- **vidiniai vartotojai** – tai organizacijos padaliniai ar atskiri darbuotojai, atliekantys tam tikrą integruotų veiklos procesų dalį. Jie gali atlikti tiek tiekėjo, tiek vartotojo vaidmenį. Kiekvienas darbuotojas yra savo kolegų darbo rezultatų vartotojas, vertintojas.

Sunku pagaminti produktą, kuris tenkintų išorinių vartotojų poreikius, jeigu organizacijoje nėra įdiegta vidinių vartotojų poreikių tenkinimo mechanizmo. Be jo neįmanoma sukurti visais atžvilgiais kokybiškų produktų mažiausiais kaštais ir be nuostolių dėl blogos veiklos kokybės. Nustatant vartotojų poreikius yra naudojamas vartotojų poreikių ir lūkesčių, paverčiamų produktais, patenkinimo procesas (9 pav.).



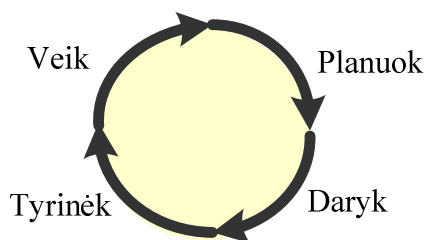
9 pav. Vartotojų poreikių ir lūkesčių patenkinimo produktais schema [58].

Tikrieji vartotojų poreikiai ir lūkesčiai vadinami laukiama kokybe. Organizacija išsiaiškina vartotojų poreikius bei lūkesčius ir paverčia juos produkto ar paslaugos specifikacija. Produkto pagaminimo rezultatas yra tikroji kokybė, kurią gaus vartotojas. Jei tikroji kokybė skiriasi nuo

laukiamos kokybės, atsiranda neatitikimas, kuris dažnai turi neigiamą įtaką vartotojų patenkinimui. Tačiau kartais laukiamos ir tikros kokybės skirtumas gali lemti ir teigiamą poveikį tiek vartotojui, tiek pačiai įmonei. Tai gali atsitikti tais atvejais, kai tikroji kokybė viršija laukiamą kokybę ir vartotojo lūkesčius.

Pasaulinėje kokybės vadybos praktikoje vartotojų poreikių patenkinimui tirti naudojami įvairūs metodai. Dauguma jų yra specifiniai, pritaikyti atskiroms organizacijoms. Tačiau visuotinės kokybės vadyboje naudojami šalies masto modeliai, pavyzdžiui, Švedijos vartotojų patenkinimo barometro modelis, Amerikos vartotojų patenkinimo indekso modelis, Europos vartotojų patenkinimo indekso modelis. Šių grafinis modelių pavaizdavimas pateiktas 5 priede.

Vartotojų poreikių patenkinimas mažiausiai kaštais neįmanomas be kito labai svarbaus visuotinės kokybės principo – nuolatinio tobulinimo. Nuolatinis tobulinimas reikalingas norint pasiekti geriausių įmonės rezultatų kaštų mažinimo požiūriu. Nuolatinis tobulinimas yra nepertraukiamas procesas, kurį galima pavaizduoti kaip Shewharto-Demingo ciklą. Šį ciklą sudaro keturi etapai: planuok, daryk, tyrinėk ir veik (10 pav.).



10 pav. Shewharto-Demingo nuolatinio tobulinimo ciklas

J. Demingas ir kiti autoriai savo darbuose apie visuotinės kokybės vadybą pabrėžia orientavimosi į proceso svarbą [58, 3]. Šie autoriai nustatė, kad dauguma su kokybe susijusių problemų atsiranda dėl netinkamo gamybos ir vadybos procesų valdymo. Todėl labai svarbu procesus tinkamai valdyti (veiklos planavimas ir organizavimas) ir tobulinti.

Be nuolatinio tobulinimo literatūroje taip pat galima surasti reaktyvinio tobulinimo ir proaktyvinio tobulinimo sąvokas. Reaktyvinis tobulinimas yra veikiančių procesų tobulinimas, reaguojant į defektus, vėlavimus ir nuostolius. Proaktyvinis tobulinimas – tai strategijos kūrimo elementas, situacijos sukūrimas ir valdymas, paėmus iniciatyvą į savo rankas, kai organizacijai reikia pasirinkti tolimesnes tinkamas veiklos kryptis. Skirtingai nuo proceso valdymo, kurio metu vartojami kiekybiniai rodikliai, proaktyvinio tobulinimo metu naudojami kokybiniai metodai.

Visuotinis dalyvavimas nuolatinio tobulinimo procese yra svarbi visuotinės kokybės sudedamoji dalis. Organizacijoje didelis dėmesys turi būti skirtas žmogiškųjų išteklių strateginiam

planavimui: darbuotojų ugdymui, darbo organizacijai, premijavimui, pripažinimui, atlyginimo už darbą sistemai ir įdarbinimo procedūroms.

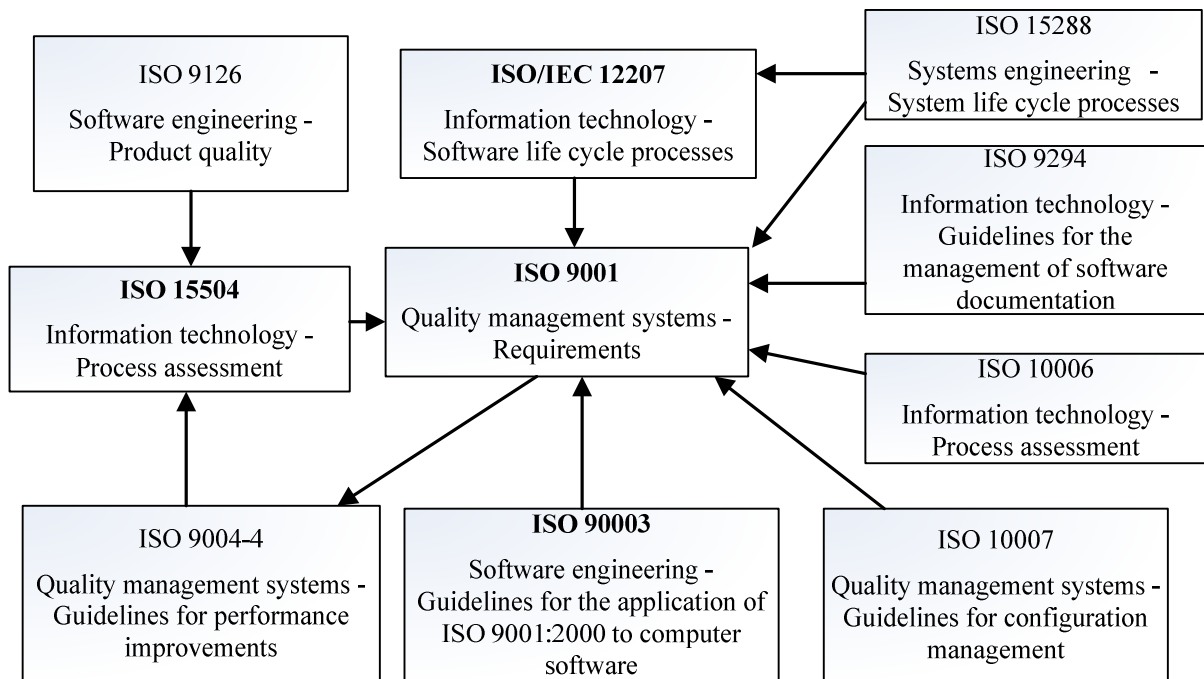
Sėkmingas visuotinės kokybės vadybos įgyvendinimas priklauso nuo visuotinės kokybės vadybą propaguojančių organizacijų, socialinės propagavimo veiklos, teisinio reglamentavimo, mokslinių tyrimų, naujų metodų kūrimo, informacijos sklaidimo, praktinio mokymo ir kt. Vadovai turi gerai suprasti visuotinės kokybės vadybos esmę ir būtinumą bei jais vadovautis savo veikloje. Tik tada į visuotinės kokybės vadybos įgyvendinimo procesą galima įtraukti specialistus, technikus ir darbininkus.

3.2. Kokybės vadybos metodų taikymas programinės įrangos kūrimo projektams

Tarptautinės standartizacijos organizacijos ISO standartai pasaulyje labiausiai žinomi ir paplitę. Tai universalūs standartai, kuriuos galima naudoti kaip modelius nepriklausomai nuo srities, kurioje veikia organizacija. ISO kokybės standartų modeliai turi tiek privalumų, tiek trūkumų. Kokybės sistemos sukūrimas, remiantis ISO modeliu, reikalauja naudoti daugelį kitų šakinių ir ISO standartų.

Iš pradžių ISO 9000 šeimos standartuose buvo mažai dėmesio skirta programinės įrangos kūrimo šakai, jei buvo labiau orientuoti į gamybos sritį. Aštuntojo dešimtmečio pabaigoje Didžiojoje Britanijoje buvo sukurta TickIT grupė, kurios tikslas buvo ISO 9001 standarto pritaikymas programinės įrangos kūrimo projektams. Šios grupės darbo rezultatas buvo ISO 9000-3:1997 standarto sukūrimas. Nepaisant to, kad ISO 9000-3 standartas naudoja programinės įrangos kūrimo terminologiją ir nagrinėja klausimus, susijusius su programinės įrangos industrija, jis iki galo neatitinka programinės įrangos projektų specifikacijų. Šiuo metu ISO 9000-3 standartą pakeitė jo naujesnė modifikacija – ISO 90003:2004 standartas, kuris savo ruožtu yra ISO 9001:2000 standarto pritaikymo programinės įrangos kūrime rekomendacijos. Palyginus su ISO 9000-3, šis standartas geriau pritaikytas šios srities specifikai, nes remiasi programinės įrangos gyvavimo ciklu ir detaliai nagrinėja klausimus, susijusius su programinės įrangos kūrimu. ISO 90003:2004 standartas – tai kokybės užtikrinimo standartas, kuris negali būti naudojamas vertinant brandos lygį ir nusakant programinės įrangos projekto rezultatus, todėl tokiais atvejais vertėtų remtis ISO/IEC 15504 standartu. ISO/IEC 15504 standartas buvo sukurtas 1993 m. bendro tarptautinių ISO ir IEC organizacijų projekto, vadinamo SPICE (angl. *Software Process Improvement for Capability dEtermination*), metu. Šis standartas skirtas informacinių sistemų ir programinės įrangos kūrimo proceso įvertinimui.

11 paveiksle pateikta ISO standartų, susijusių su programinės įrangos kūrimu, schema.



11 pav. ISO standartų šeima

Programinės įrangos kokybės valdymą galima prilyginti jos kokybės užtikrinimo procesui. Sėkmingas kokybės valdymas turi būti labiau orientuotas į viso proceso tobulinimą, o ne į galutinio produkto kontrolę. ISO 9001 standartas visame pasaulyje pripažintas kaip kokybės valdymo sistemų, pritaikytų tam tikro produkto (taip pat ir programinę įrangą) kūrimo situacijoms, nustatymo ir įvertinimo priemonė. ISO 9000 standarto reikalavimus galima naudoti valdymo požiūriu jungiant programinės įrangos kokybės užtikrinimo praktiką ir programinės įrangos proceso tobulinimo programas.

Pagrindinis ISO 9000 principas – kokybės užtikrinimas per proceso tobulinimą ir jį apibrėžiantį kontraktinio pagrindo užtikrinimą. ISO 9000 standartų šeima apibrėžia tris kokybės valdymo sistemų modelius: ISO 9001, ISO 9002 ir ISO 9003. Programinės įrangos ir informacinių sistemų kūrimo procesas turi atitikti ISO 9001 standarto reikalavimus. Šis standartas yra plačiausias, nes apima būtinas kūrimo, gamybos, paslaugų ir palaikymo sritis. Programinės įrangos kūrėjai turėtų panaudoti ISO 90003 standartą, ISO 9001 taikymo tobulinant, tiekiant, įrengiant ir prižiūrint kompiuterio programinę įrangą vadovas (angl. *Guidelines for the Application ISO 9001 to the Development, Supply, Installation and Maintenance of Software*), dėl programinės įrangos produkto specifiškumo.

ISO 90003 – tai ISO 9001 taikymo programinės įrangos industrijoje rekomendacijos. ISO 9000-3 rekomendacijų punktus galima susieti su ISO 9001 elementais.

Y.Helio Yang pasiūlė aštuonių žingsnių įgyvendinimo sistemą, kuri visas dvidešimt ISO 9000 kokybės sistemos sąlygų skirsto į grupes [19].

1 žingsnis: organizacijos ir darbuotojų atsidavimo ISO 9001 programai nustatymas. Pirmas žingsnis skirtas tam, kad vadovybė ir darbuotojai būtų pasirengę sukurti kokybės programą remiantis ISO 9000 standartu. Šio žingsnio metu nustatomi organizacijos tikslai, suprantami ISO 9000 reikalavimai ir standartai bei sudaroma valdymo komanda, atsakinga už programos įvykdymą.

2 žingsnis: aukštos kokybės sistemos struktūros nustatymas ir dokumentavimas, kokybės instrukcijų rašymas. Kokybės sistemos tikslas – sukurti dokumentuotą viso programinės įrangos produkto gyvavimo ciklo ir pagalbinių veiklų struktūrą. Kokybės sistema turi užtikrinti produkto kokybę jo kūrimo proceso metu, o ne proceso pabaigoje. Pabrėžiama problemos prevencija, o ne problemos pataisymas. Šiame žingsnyje sudaroma kokybės politika, aukšto lygio kokybės programos struktūra, įskaitant dokumentacijos struktūrą ir kokybės instrukcijas.

3 žingsnis: pagrindinių kokybės sistemos elementų nustatymas. Šio žingsnio metu nustatomi tokie pagrindiniai procesai: dokumentų kontrolė, pataisos ir prevencijos veiksmai, kokybės įrašų kontrolė ir apmokymai efektyviam ISO 9001 standarto įgyvendinimui. Organizacijai reikia nustatyti, kaip dokumentacija bus tvarkoma, saugoma ir numeruojama. Turi būti sukurta formali jos pateikimo ir stebėjimo procedūra. Be to, apibrėžiamos kokybės programos uždaviniai ir pagrindinė struktūra, su jais supažindinami visi organizacijos nariai. Nustačius apmokymų programos struktūrą, atliekami pirmieji kokybės įrašų kontrolės apmokymai.

4 žingsnis: kūrimo metodų nustatymas. Organizacijoms, kuriose nėra apibrėžti programinės įrangos metodai, šis žingsnis gali užsitęsti, nes apima daug esminių dalykų – kūrimo kontrolę, produkto identifikavimą ir stebėjimą, kontraktų peržiūrą, proceso kontrolę, patikrinimus ir testavimą. Kiekvienas šio žingsnio elementas priklauso nuo organizacijos. Kūrimo kontrolė yra sunkiausiai programinės įrangos kūrimo organizacijoje įvykdomas elementas. Dauguma programinės įrangos kūrėjų programinės įrangos kūrimą traktuoja kaip meną, o ne mokslą ir atsisako sekti griežtų metodų. Nepaisant to, yra daug skirtingų metodų. Jei organizacijoje nėra nustatytos metodologijos ir, jei ji nenaudoja objektinio programavimo kalbų ir priemonių, geriausia pradėti nuo „krioklio“ modelio. Nors šis modelis turi tam tikrų trūkumų, jis lengvai pritaikomas ir nuo jo galima lengvai pereiti prie kitų programinės įrangos kūrimo metodų. Kontraktų peržiūra priklauso nuo kuriamos programinės įrangos tipo. Organizacija gali kurti programinę įrangą tam tikram užsakovui arba parduoti atviroje rinkoje. Turi būti nustatyti programinės įrangos testavimo tipai, į kuriuos įeina integracijos, sistemos ir priėmimo testavimai. Patikrinimai ir testavimo būsenos yra konfigūracijų valdymo žingsnio dalis.

5 žingsnis: pagrindinių palaikymo elementų nustatymas. Norėdami tiksliai nustatyti ir efektyviai įvykdyti palaikymo veiksmus, rekomenduojama atlikti 5 ir 6 žingsnius. Penktas žingsnis apima veiksmus, susijusius su vartotojų poreikių neatitinkančių produktų kontrole, priežiūra, pakavimu, išlaikymu ir pristatymu. Turi būti atliekamos procedūros, užtikrinančios nekokybiškų produktų (pavyzdžiui, produktai, kurie neatitiko testavimo, konfigūravimo ar pristatymo reikalavimų)

atskyrimą nuo gerų produktų. Jei organizacija platina programinės įrangos produktus Internetu, turi būti numatytos atitinkamos platinimo procedūros.

6 žingsnis: likusių palaikymo elementų nustatymas. Likusieji palaikymo veiksmai apima vidaus kokybės auditą, klientui pateiktų produktų kontrolę, patikrinimų, įvertinimų ir įrangos testavimo kontrolę ir statistinius metodus. Vidaus kokybės auditas leidžia organizacijai prieš išorinį auditą nustatyti programų defektus. Vidaus audito komandą turi sudaryti skirtingų funkcinių sričių atstovai. Yra taikomi du statistinių metodų tipai: produkto įvertinimas ir proceso įvertinimas. Pagrindinis tokių vertinimų tikslas – nustatyti sritis, linkusias į klaidas.

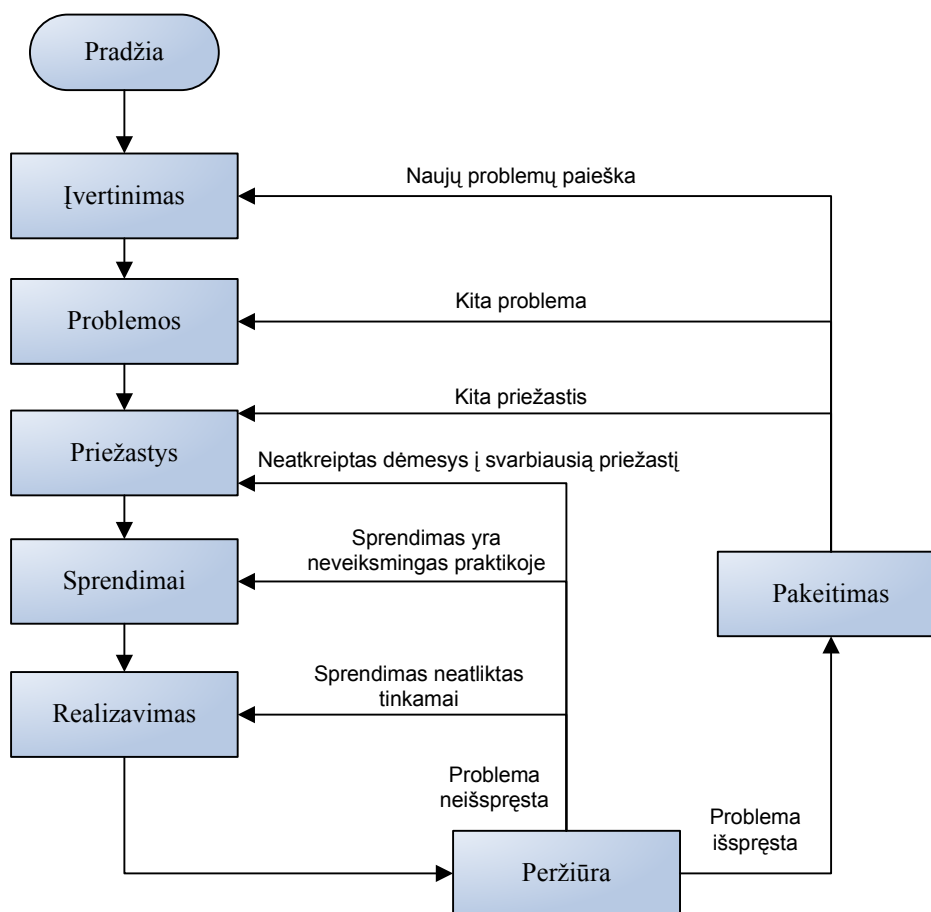
7 žingsnis: preliminarus vertinimo audito atlikimas ir defektų šalinimas. Kai organizacija sudokumentuoja savo procesus ir pasirenka registratorių, turi būti atliktas preliminarus vertinimo auditas. Toks auditas suteikia galimybę rasti nustatytų procesų defektus prieš atliekant galutinį auditą. Preliminarus vertinimo auditas paprastai trunka dvi dienas. Jei audito metu buvo aptikti defektai, jie turi būti pašalinti.

8 žingsnis: registravimo audito atlikimas. Kai organizacija išsprendžia visas problemas, atliekamas galutinis auditas. Šis auditas vyksta apie tris dienas. Jei audito metu nebuvo rasta klaidų ar problemų, auditoriai paruošia galutinę ataskaitą su rekomendacija registruoti ir pateikia ją ISO vyriausiajai valdybai Briuselyje. ISO vyriausioji valdyba peržiūri ataskaitą ir per aštuonias savaites patvirtina ir išduoda sertifikatą.

Programinės įrangos proceso tobulinimo sistema remiasi programinės įrangos įvertinimo procesu ir tobulinimo priemonėmis bei Demingo ciklu, kurį sudaro keturios tobulinimo proceso veiklos: planuok, daryk, tyrinėk, veik.

Programinės įrangos proceso tobulinimo sistema labiau tinka organizacijoms, turinčioms ISO 9001 sertifikatus ir siekiančioms įsteigti visuotinės kokybės vadybos kultūrą, nes dauguma šios sistemos veiklų reikalauja kokybės sistemos infrastruktūros. Programinės įrangos proceso tobulinimo sistema numato, kad kiekvienas programinės įrangos kūrime dalyvaujantis asmuo turi prisidėti prie programinės įrangos proceso tobulinimo.

Programinės įrangos proceso tobulinimo sistemą sudaro 7 fazės [18] (12 pav.). Ji prasideda nuo įvertinimo (angl. *Assess*), kurios tikslas – įvertinti programinės įrangos procesą, naudojant programinės įrangos proceso brandos modelį ir programinės įrangos defektų analizės priemones. Proceso brandos modelį sudaro penki nepertraukiamo proceso tobulinimo etapai. Jis suteikia plėtros kelią, kuris padidina organizacijos programinės įrangos proceso brandą. Programinės įrangos defektų duomenys gali būti surinkti iš tam tikros produkto komandos, projekto ar produkto, todėl tokių duomenų analizė tam tikram procesui yra specifinė. Surinkti duomenys leidžia paaikškinti kritines proceso tobulinimo problemas.



12 pav. Programinės įrangos proceso tobulinimo sistemos fazės

Problemų fazėje nustatomos, patikrinamos ir išdėstomos pagal svarbumą potencialios problemos, susijusios su pagrindinėmis proceso sritimis. Pirmiausia sprendžiamos prioritetinės problemos. Priežasčių fazės tikslas – nustatyti pagrindines nagrinėjamos problemos priežastis. Nustatytos priežastys turi būti patikrintos ir sugrupuotos pagal svarbumą. Sprendimų fazėje pasirenkama svarbiausia problemos priežastis ir nustatomi galimi sprendimai. Geriausias sprendimas pasirenkamas tada, kai visi sprendimai įvertinti pagal jų tinkamumą, galimus šalutinius poveikius, sąnaudas ir naudingumą. Pasirinktam sprendimui galima pritaikyti uždavinio, tikslo ir parametru paradigmą. Tobulinimo sprendimo uždavinys ir tikslai turi būti tiksliai apibrėžti, o programinės įrangos parametrai, naudojami sprendimui patikrinti – išaiškinti. Įgyvendinimo fazės paskirtis – bandomasis sprendimo pritaikymas ir jo veikimo praktikoje patikrinimas. Sprendimo veikimo efektyvumas nustatomas naudojant ankstesnėje fazėje apibrėžtus parametrus. Peržiūros fazė – tai programinės įrangos proceso tobulinimo gairės. Sprendimo sėkmingumas pažymimas nustatant tobulinimo proceso reikšmingumą užsibrėžtiems tikslams. Jei patobulinimas yra nereikšmingas, turi būti nustatytos priežastys. Atsižvelgiant į peržiūros fazės rezultatus gali būti atlikti keli veiksmai. Jei nagrinėjama problema išspręsta, sprendimas turi būti standartizuotas ir dokumentuotas pakeitimo

fazėje, tobulinimo veiksmai pritaikomi kitoms priežastims, problemoms arba grįžtama prie įvertinimo fazės, tęsiant kokybės tobulinimo procesą. Kitu atveju pateikiami neišspręstų problemų paaiškinimai įgyvendinimo, sprendimų ar priežasčių fazėms. Tik pakeitimo fazėje daromi faktiniai programinės įrangos proceso pakeitimai. Įrodytas patobulinimas dokumentuojamas ir integruojamas į pasirinktą procesą.

Pagrindinė kokybės kėlimo taisyklė – visada įmanoma pagerinti kokybę ir kartu mažinti kaštus. Viena iš svarbesnių visuotinės kokybės idėjų yra tai, kad visi proceso dalyviai turi būti įtraukti į kokybės kėlimo darbus, kurie reikalauja, kad statistiniai kokybės kontrolės principai būtų apibendrinti kiek įmanoma paprasčiau, o veiksmingos priemonės turi naudoti visi dalyviai. Šiam tikslui Kaoru Ishikawa sujungė septynias kokybės kontrolės priemones. Nors šios septynios priemonės sėkmingai naudojamos nuo 1960 metų daugelyje sričių, aštuntojo dešimtmečio pradžioje jos buvo papildytos septyniomis naujomis kokybės kontrolės priemonėmis (dar vadinamomis septyniomis vadybos priemonėmis).

Nors kai kurios priemonės, pavyzdžiui, histogramos, Pareto analizė ir Ishikawa diagramos, buvo naudojamos programinės įrangos kūrimo ir kokybės užtikrinimo veiklose, 14 kokybės kontrolės priemonių ilgai programinės įrangos proceso gerinimo kontekste nebuvo sistemingai naudojamos. He ir kt. šias keturiolika priemonių išnagrinėjo programinės įrangos proceso gerinimo sistemos kontekste.

Kontroliniai sąrašai (angl. *Checklists*) – tai paprasčiausia duomenų surinkimo priemonė, padedanti rankiniu būdu rinkti ir klasifikuoti duomenis [18]. Kontroliniai sąrašai užima programinės įrangos kūrime svarbią vietą, nes apibendrina svarbiausius proceso punktus. Šie sąrašai dažnai įeina į proceso dokumentaciją. Be to, kontroliniai sąrašai naudojami bendroms klaidoms fiksuoti. Histograma – tai grafinė priemonė, padedanti apibendrinti didelės apimties duomenis parodant matavimų grupės pasiskirstymo dažnį.

Pareto analizė pagrįsta 80/20 principu, tai yra 80 proc. problemų kyla dėl 20 proc. priežasčių. Šia analize be daugelio kitų atskleidžiamos priežastys, darančios didžiausią poveikį. Šis aspektas yra svarbus nuolatiniam kokybės tobulinimui, nes jį galima pasiekti tik sprendžiant problemas. Pareto analizė lengvai pritaikoma programinės įrangos kokybės kontrolei, nes programinės įrangos defektai ar jų tankumas niekada nebūna vienodai pasiskirčiusios.

Priežasčių – pasekmių diagramos (angl. *Cause-and-effect diagrams*), dar vadinamos „žuvies kaulo“ ar Ishikawos, – tai kokybinė priemonė, apibendrinanti priežasčių – pasekmių analizę. Šio tipo diagramos retai naudojamos programinės įrangos kūrimo proceso metu.

Suskaidymo analizė (angl. *Stratification analysis*) atliekama norint išvengti tikslinių duomenų iš skirtingų šaltinių supainiojimo. Tai galinga priemonė leidžianti suskaidyti duomenis naudojant aplinkybių duomenis, surinktus kartu su tiksliniais duomenimis, ir ieškoti tam tikrų pataisos veikslių. Suskaidymas dažnai naudojamas kartu su kitomis priemonėmis.

Tendencijų diagramos (angl. *Scatter plot*) padeda nustatyti pavaizduotų duomenų tendas. Be to, šias diagramas galima panaudoti tiriant santykius (koreliacijos, regresijos, statistinio modeliavimo, priešasčių-pasekmių sąsajų) tarp kintamųjų dydžių. Lyginant su kitomis priemonėmis, šias diagramas sunku pritaikyti. Sklaidos diagramos dažnai siejamos su tiriamuoju darbu ir reikalauja tikslių duomenų.

Kontrolės diagrama – tai dinaminė statistinio proceso kontrolės priemonė, kurią galima panaudoti nustatant stebimo proceso atitikimą apibrėžtiems tikslams ir reikalingus pataisos veiksmus. Ši diagrama taip pat parodo, ar procesas yra stabilus. Jei procesas yra svyruojantis, patobulinimą galima pasiekti be sistemos lygio pakeitimo. Jei procesas stabilus, patobulinimas gali būti atliktas tik per sistemos lygio pakeitimus.

Aštuntajame dešimtmetyje atsirado septynios naujos kokybės kontrolės priemonės. Jos mažiau žinomos nei septynios senos kokybės kontrolės priemonės, nes yra nukreiptos į strateginį kokybės planavimą, yra labiau kokybinės ir sunkiai valdomos [18]. Šias priemones sudaro: panašumų (angl. *Affinity diagram*), santykių (angl. *Relations diagram*), medžio (angl. *Tree diagram*), matricinė diagramos (angl. *Matrix diagram*), matricinė duomenų analizė (angl. *Matrix data analysis*), proceso sprendimo programos (angl. *Process decision program chart*) ir procedūros diagramos (angl. *Procedure diagram*).

Naujos kokybės kontrolės priemonės pradėjo naują kokybės amžių, pagrįstą dviem fundamentaliais reikalavimais:

- papildomos vertės, viršijančios vartotojų poreikius, sukūrimas;
- klientų poreikių tenkinimo nesėkmės prevencija, o ne pataisymas.

Septynių naujų kokybės kontrolės priemonių pritaikymas programinės įrangos kūrimo procesui žymiai palengvina kokybės funkcijos diegimą ir sukuria teigiamą patirtį, renkant ir tikrinant klientų reikalavimus [29].

Panašumų diagrama – tai grafinė „smegenų šturmo“ pavaizdavimo priemonė, naudojama grupuojant faktus, nuomones, idėjas ir klientų pageidavimus, remiantis tam tikru panašumu.

Santykių diagrama yra grafinė priešasčių-pasekmių analizavimo priemonė, naudojama strateginio kokybės planavimo problemų nustatymo ir aprašymo fazėje, kai reikia išaiškinti ir suprasti sudėtingus santykius. Ši diagrama tvarko specifinę informaciją į bendrą ir parodo svarbiausias priešastis ir pagrindines pasekmes. Ji skiriasi nuo priešasčių-pasekmių diagramos tuo, kad parodo sudėtingas priešastis ir pasekmes, o priešasčių-pasekmių diagrama parodo vieno priklausomo kintamojo (pasekmės) ir jo priešasties struktūrą. Be to, santykių diagrama yra tinklo, o priešasčių-pasekmių – medžio struktūros.

Medžio diagrama, dar vadinama hierarchijos ar sisteminė diagrama, naudoja smulkinimo principą, klausimą skaidant į elementų seką iki įgyvendinimo. Pagrindinė problema, idėja ar pagrindinis klientų poreikis gali būti išskaidytas į sudedamąsias dalis. Ilgalaikis kokybės tobulinimo tikslas gali būti klasifikuotas į trumpalaikius tikslus. Šia diagrama galima sistemingai fiksuoti kelių nuo kokybės tikslo prie labiau detalizuotų uždavinių ir jų įvykdymo būdų.

Matricinė diagrama – tai lentelės formos priemonė, palengvinanti santykių tarp dviejų ar daugiau veiksnių rinkinių nustatymą. Ji tinka planuojant išteklius, nustatant seką, nesėkmių prevencijai ir kitais atvejais.

Matricinė duomenų analizė yra vienintelė iš septynių naujų kokybės kontrolės priemonių, skirta skaitmeninių duomenų analizei. Ji naudojama išmatuoti ir įvertinti skaičiais santykių tarp skirtingų veiksnių laipsnį. Tokios analizės rezultatai vaizduojami stačiakampėje ar trikampėje matricioje (koreliacijos tarp kiekvienos veiksnių poros viename duomenų rinkinyje). Santykių laipsnio kvantavimas nustatomas remiantis surinktų duomenų subjektiniais svoriais arba statistine analize.

Proceso sprendimo programos diagrama – tai planavimo priemonė, naudojama įvertinti proceso alternatyvas pradiname nustatyme ir kuriant procesus tam, kad būtų sukurtas geriausias aukšto lygio procesas. Šios diagramos naudojamos parodyti veiksmų ir sprendimų, reikalingų pasiekti norimą rezultatą, seką arba sutrukdyti nenorimą įvykį.

Procedūros diagrama (arba rodyklės diagrama) – tai planavimo ir komunikacijos priemonė, užtikrinanti tinkamiausią kiekvieno uždavinio planavimo laiką ir palengvinanti darbo kontrolę. Lyginant su Ganto diagrama, procedūros diagrama suteikia detalesnį nuoseklumą ir veiklos trukmę.

Aptartos 14 kokybės kontrolės priemonių pritaikymas programinės įrangos proceso tobulinimo sistemoje pateiktas 3 lentelėje.

3 lentelėje pateiktas kokybės kontrolės priemonių panaudojimas atskirose programinės įrangos proceso tobulinimo sistemos fazėse. Įvertinimo fazėje renkami duomenys ir vaizduojami kontroliniuose sąrašuose, histogramas ir suskaidymo analizę. Defektų analizė naudoja tendencijų ir kontrolės diagramas. Proceso brandos modelio tikslai, lygiai ir pagrindiniai procesai gali būti pavaizduoti medžio diagramoje. Kokybės funkcijos diegimas naudoja matricinę diagramą ir matricinę duomenų analizę, sujungiant organizacijos poreikius į proceso tobulinimo projektą.

Problemos fazėje ankstesnė fazė pateikiama kontrolinių sąrašų, histogramų, suskaidymo analizės, tendencijų ir kontrolės diagramose bei analizuojama Pareto ir suskaidymo analizių bei panašumų ir medžio diagramų pagalba. Priežasčių fazėje Pareto analizė, tendencijų diagramos ir santykių diagramos naudojamos rasti priežastis. Sprendimo fazėje kontroliniai sąrašai ir Pareto analizė naudojamos pasirenkant sprendimus, o kontrolės, santykių, medžio, matricinė diagramos, matricinė duomenų analizė ir proceso sprendimo programos bei procedūros diagramos naudojamos tikrinant ir planuojant sprendimą. Realizavimo fazės metu kontroliniai sąrašai, histogramos ir suskaidymo analizė

vaizduoja duomenis, o tendencijų ir kontrolės diagramos naudojamos stebint ir analizuojant proceso įgyvendinimą. Peržiūros fazėje kontroliniai sąrašai, histogramos ir suskaidymo analizė vaizduoja duomenis, Pareto analizė, tendencijų ir kontrolės diagramos naudojamos tikrinant patobulinimą, o prižasčių-pasekmių, panašumų ir santykių diagramos padeda nustatyti sprendimo pasisekimo ar žlugimo priežastis. Pakeitimo fazės metu einamojo proceso pakeitimai (kontrolės ribų pakeitimai (kontrolės diagramos), kitų procesų pakeitimai (panašumų, santykių ir medžio diagramos) turi būti planuojami naudojant proceso sprendimo programos ir procedūros diagramas.

3 lentelė

Kokybės kontrolės priemonės PĮ proceso tobulinimo sistemoje [18]

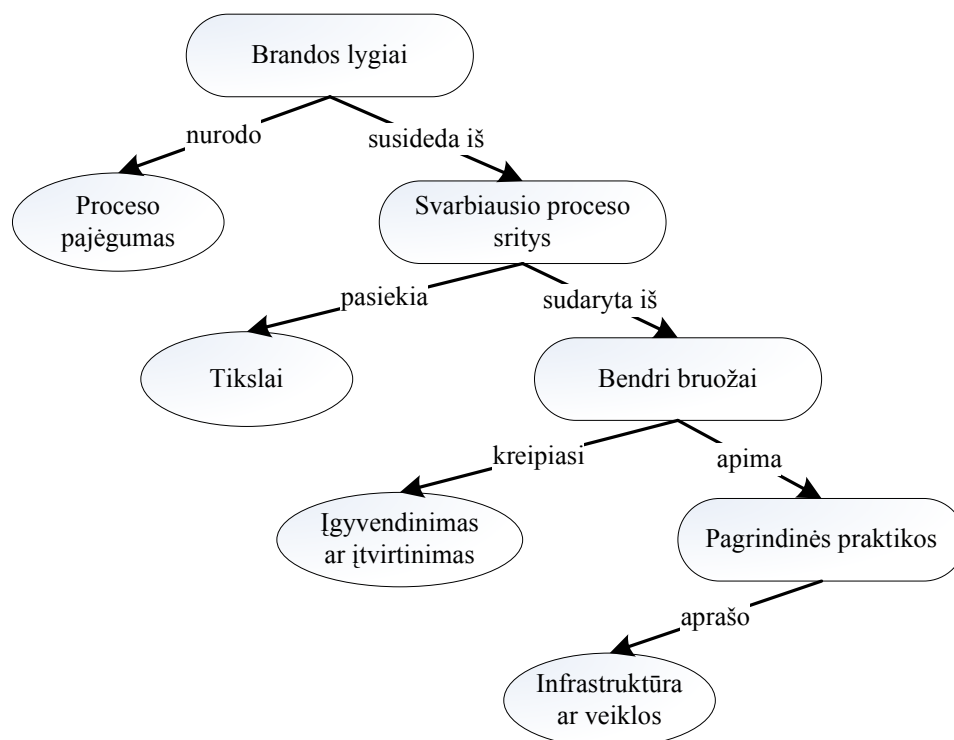
Eil. Nr.	Kokybės kontrolės priemonė	PĮ proceso tobulinimo sistemos fazė						
		Įvertinimas	Problemos	Priežastys	Sprendimai	Realizavimas	Peržiūra	Pakeitimas
1	Kontroliniai sąrašai	✓	✓		✓	✓	✓	
2	Histograma	✓	✓		✓	✓	✓	
3	Pareto analizė		✓	✓	✓		✓	
4	Priežasčių-pasekmių diagramos			✓			✓	
5	Suskaidymo analizė	✓	✓	✓		✓	✓	
6	Tendencijų diagrama	✓	✓	✓		✓	✓	
7	Kontrolės diagrama	✓	✓		✓	✓	✓	✓
8	Panašumų diagrama		✓				✓	✓
9	Santykių diagrama			✓	✓		✓	✓
10	Medžio diagrama	✓	✓		✓			✓
11	Matricinė diagrama	✓			✓			
12	Matricinė duomenų analizė	✓			✓			
13	Proceso sprendimo programos diagrama				✓			✓
14	Procedūros diagrama				✓			✓

Nors programinės įrangos kūrimo ir gamybos procesas turi daug skirtumų, pavyzdžiui, proceso ir produkto kaitojimasis, statistinių priemonių panaudojimas ir kt., gamybos proceso tobulinimo principus ir praktiką galima peržiūrėti ir įvesti į programinės įrangos proceso tobulinimą. Visos aptartos keturiolika kokybės kontrolės priemonių gali būti panaudotos programinės įrangos proceso tobulinimo sistemoje.

Programinės įrangos proceso brandos modelis (angl. *CMM – Capability Maturity Model for Software*) aprašo programinės įrangos proceso brandos principus ir praktikas. Numatyta, kad modelis padeda programinės įrangos kūrimo organizacijoms pagerinti savo programinės įrangos procesų brandą ir nueiti evoliucijos kelią nuo chaotiškų prie subrendusių, metodiškų programinės įrangos

kūrimo procesų [45]. Proceso brandos modelį sudaro penki brandos lygiai. Kiekvienas lygis – tai aiškiai nustatytas evoliucinis brandos stabilizavimas link programinės įrangos procesų brandos pasiekimo. Kiekvienas brandos lygis sudaro pagrindą nepertraukiam proceso tobulinimui.

Proceso brandos modelio struktūra pateikta 13 pav. Kiekvieną brandos lygį, išskyrus pirmąjį, sudaro kelios svarbios proceso sritys. Kiekvieną svarbiausią proceso sritį sudaro penkios dalys – bendrieji bruožai. Bendrieji bruožai nurodo pagrindines praktikas, kurios padeda įgyvendinti svarbiausius proceso srities tikslus.



13 pav. Proceso brandos modelio struktūra [46]

Išskiriami penki proceso brandos modelio lygiai.

1. Pradinis lygis (angl. *Initial Level*). Šiame lygyje programinės įrangos kūrimo procesas charakterizuojamas kaip chaotiškas. Pradiniame lygyje organizacijos paprastai nenumato stabilios programinės įrangos kūrimo ir palaikymo aplinkos. Tokio brandos lygio organizacijos dažnai sukuria veikiančius produktus viršydamos biudžetą ir pažeisdamos tvarkaraštį. Pradinio lygio organizacijų sėkmė priklauso nuo organizacijos darbuotojų kompetentingumo ir negali būti kartotinė, nebent tie patys darbuotojai dalyvauja kitame projekte. Šiame lygyje pajėgumas yra atskirų darbuotojų, o ne organizacijos bruožas.

2. Kartojamas lygis (angl. *Repeatable Level*). Šio lygio organizacijos sukuria pagrindinius projekto vadybos procesus, skirtus projekto kaštų, tvarkaraščio ir funkcionalumo stebėjimui. Naujų

projektų planavimas ir valdymas grindžiamas panašių projektų patirtimi. Proceso pajėgumas gali būti pagerintas vykdomų projektų pagrindu sukuriant bazinio proceso valdymo discipliną. Kartojamo lygio organizacijų projektai įdiegia pagrindines projekto valdymo priemones. Antro lygio organizacijos procesai priklausomai nuo vykdomų projektų gali skirtis. Norėdamos pasiekti antrą brandos lygį, organizacijos privalo sukurti tokią politiką, kuria būtų kuriami atitinkami projektų valdymo procesai. Antro lygio organizacijų programinės įrangos proceso pajėgumas gali būti apibendrintas kaip sistemingas (angl. *disciplined*), nes programinės įrangos projekto planavimas ir stebėjimas yra stabilus ir ankstesnės sėkmės gali pasikartoti.

3. Apibrėžtas lygis (angl. *Defined Level*). Šiame lygyje standartinis programinės įrangos kūrimo ir palaikymo procesas yra dokumentuotas, standartizuotas ir integruotas į organizacijos programinės įrangos kūrimo proceso standartą. Visi projektai naudoja patvirtintą ir pritaikytą organizacijos programinės įrangos kūrimo ir palaikymo proceso standarto versiją. Trečiame lygyje nustatyti procesai padeda programinės įrangos vadovams ir techniniam personalui efektyviau vykdyti savo darbus. Šio lygio organizacijų programinės įrangos pajėgumas gali būti apibendrintas kaip standartinis ir nuoseklus, nes programinės įrangos kūrimo ir valdymo veiklos yra stabilios ir pasikartojančios. Kontroliuojami kaštai, tvarkaraštis ir funkcionalumas bei stebima programinės įrangos kokybė. Šio proceso pajėgumas pagrįstas pagrindiniu visos organizacijos supratimu apie nustatyto programinės įrangos proceso veiklas, roles ir atsakomybes.

4. Valdomas lygis (angl. *Managed Level*). Šiame lygyje organizacija nustato kiekybinius programinės įrangos produkto ir procesų kokybės tikslus. Svarbių programinės įrangos proceso veiklų produktyvumas ir kokybė matuojami per visus projektus pagal organizacijos matavimo programą. Renkami detalūs programinės įrangos proceso ir produkto kokybės matavimai. Jie sudaro kiekybinį programinės įrangos projektų procesų ir produktų vertinimo pagrindą. Ketvirto lygio organizacijų programinės įrangos proceso pajėgumas gali būti apibendrintas kaip kiekybiškai įvertinamas ir prognozuojamas, nes procesas yra matuojamas ir funkcionuoja neperžengiant matavimo ribų. Šio lygio proceso pajėgumas leidžia organizacijai prognozuoti proceso ir produkto kokybės tendencijas jų kiekybinėse ribose.

5. Optimizavimo lygis (angl. *Optimizing Level*). Penktame lygyje visa organizacija koncentruojasi ties nuolatinio proceso tobulinimu. Organizacija turi priemones, nustatančias jos silpnybes ir sustiprinančias procesą, siekiant išvengti defektų atsiradimą. Programinės įrangos proceso efektyvumo duomenys panaudojami atliekant naujų technologijų ir siūlomų organizacijos programinės įrangos proceso pakeitimų išlaidų ir naudos analizę. Geriausios programinės įrangos kūrimo praktikų inovacijos nustatomos ir perkeliamos į visą organizaciją. Penkto lygio organizacijų programinės įrangos proceso pajėgumą galima apibūdinti kaip nuolat tobulinamą, nes penkto lygio organizacijos

nepertraukiamai siekia pagerinti jų proceso pajėgumo ribas, kartu tobulinant jų projektų veikimo procesą. Technologijos ir proceso tobulinimas planuojami ir valdomi kaip kiti įprasti verslo procesai.

Kiekvienas brandos lygis, išskyrus pirmą, išskaidomas į pagrindines proceso sritis (9 priedas), kurių turėtų laikytis organizacijos, tobulindamos programinės įrangos kūrimo procesą. Kiekviena pagrindinė proceso sritis nustato grupę susijusių veiklų, kurių tinkamas įvykdymas užtikrins brandos lygio pasiekimą.

Pagrindinės antro lygio proceso sritys koncentruojasi ties programinės įrangos projekto dalykais, susijusiais su bazinių projekto vadybos priemonių sukūrimu:

- *Reikalavimų valdymas* sukuria tarpusavio supratimą tarp klientų reikalavimų ir paties projekto.
- *Programinės įrangos projekto planavimas* sukuria pagrįstus programinės įrangos kūrimo ir valdymo projekto vykdymo planus .
- *Programinės įrangos stebėjimas ir vadovavimas* sukuria adekvatų faktinio progreso matomumą tam, kad vadovai galėtų imtis efektyvių veiksmų, kai projekto įgyvendinimas žymiai skiriasi nuo planų.
- *Programinės įrangos subrangos valdymas* kvalifikuotų programinės įrangos subrangovų pasirinkimas ir efektyvus jų valdymas.
- *Programinės įrangos konfigūracijų valdymas* sukuria ir palaiko programinės įrangos projekto produktų vientisumą visą programinės įrangos gyvavimo ciklą.
- *Programinės įrangos kokybės užtikrinimas* suteikia vadovybei tinkamą programinės įrangos proceso ir kuriamo produkto matomumą.

Trečias brandos lygis sprendžia tiek projekto, tiek organizacijos klausimus per jos sukurtą infrastruktūrą, kuri oficialiai patvirtina veiksmingus visų programinės įrangos kūrimo ir valdymo projektų procesus:

- *Organizacijos proceso pabrėžimas* nustato organizacinę programinės įrangos kūrimo proceso veiklų, kurios patobulina visą organizacijos programinės įrangos kūrimo proceso pajėgumą, atsakomybę.
- *Organizacijos proceso apibrėžimas* sukuria ir palaiko tinkamą naudoti programinės įrangos proceso įvertinimų rinkinį, kurio pagalba tobulinamas visų projektų procesų vykdymas, ir suteikia pagrindą augančiai ir ilgalaikiai naudai organizacijai.
- *Programinės įrangos produkto kūrimas* sistemingai atlieka gerai apibrėžtą kūrimo procesą, kuris jungia visas programinės įrangos kūrimo veiklas, efektyviai ir veiksmingai sukuriant teisingą ir nuoseklų programinės įrangos produktą.

- *Kompleksinis programinės įrangos valdymas* jungia programinės įrangos kūrimo ir valdymo veiklas į nuoseklų, apibrėžtą programinės įrangos procesą, kuris yra pagrįstas organizacijos programinės įrangos proceso standartu.
- *Tarpgrupinis koordinavimas* programinės įrangos kūrimo grupei nustato priemones, kad jie galėtų aktyviai bendradarbiauti su kitomis kūrimo grupėmis, kad projektas galėtų efektyviai ir veiksmingai tenkinti vartotojų poreikius.
- *Darbo įvertinimas* laiku ir efektyviai pašalina programinės įrangos produkto defektus. Padarinių efektas kelia supratimą apie programinės įrangos kūrimo darbus ir defektus, kurių būtų galima išvengti.
- *Apmokymo programos* gilina individų žinias ir įgūdžius tam, kad jie galėtų efektyviai ir veiksmingai vykdyti darbus.

Pagrindinės ketvirto brandos lygio proceso sritys kuria kiekybinį programinės įrangos proceso ir kuriamo produkto supratimą:

- *Kiekybinis proceso valdymas* kiekybiškai kontroliuoja programinės įrangos projekto proceso vykdymą.
- *Programinės įrangos kokybės valdymas* kuria kiekybinį supratimą apie programinės įrangos produktų projekto kokybę ir pasiekia konkrečius kokybės tikslus.

Pagrindinės penkto lygio proceso sritys apima klausimus, į kuriuos organizaciją turi atkreipti dėmesį įgyvendindama nuolatinį projektą ir įvertindama programinės įrangos proceso tobulinimą:

- *Defektų prevencija* nustato defektų priežastis ir neleidžia joms pasikartoti.
- *Technologijos pakeitimo valdymas* nustato naujas pelningas technologijas ir perkelia jas į organizaciją.
- *Proceso pakeitimo valdymas* – nepertraukiamai tobulina organizacijoje naudojamus programinės įrangos procesus, tobulinant programinės įrangos kokybę, didinant produktyvumą ir mažinant produkto kūrimo ciklo laiką.

Kiekviename proceso brandos lygyje pabrėžiama programinės įrangos kokybės svarba ir nustatomos nuolatinio kokybės tobulinimo priemonės, o kiekvieną pagrindinę proceso sritį sudaro bendri bruožai. Bendri bruožai – tai atributai, parodantys, ar pagrindinių proceso sričių įgyvendinimas yra efektyvus, pasikartojantis ir ilgalaikis. Išskiriami penki bendri bruožai [46]:

- *Įsipareigojimas įvykdyti* aprašo veiksmus, kuriuos turi atlikti organizacija, užtikrindama procesų įsteigimą ir įvykdomumą (apima organizacijos politiką ir aukščiausios vadybos finansavimą).

- *Sugebėjimas įvykdyti* aprašo projekto ar organizacijos prielaidas, reikalingas kompetentingai įvykdyti programinės įrangos kūrimo procesą (apima išteklius, organizacines struktūras ir apmokymus).
- *Įvykdyti veiksmi* aprašo būtinas roles ir atsakomybes, reikalingas įvykdyti pagrindines proceso sritis (planų, procedūrų, atliktų darbų, stebėjimo ir taisos darbų praktikos).
- *Matavimai ir analizė* aprašo proceso vertinimo ir matavimų analizės poreikį (apima matavimų pavyzdžius, kuriuos galima pritaikyti nustatant įvykdytų veiksmų efektyvumą ir būseną).
- *Įgyvendinimo patvirtinimas* aprašo žingsnius, kuriais užtikrinamas veiksmų atlikimas, laikantis nustatyto proceso (apima valdymo peržiūrų ir audito praktikas).

Įvykdytų veiksmų praktikos aprašo, kas turi būti įvykdyta, norint sukurti proceso pajėgumą. Visos kitos praktikos taip pat svarbios, nes formuoja pagrindą, kuriuo organizacija gali įtvirtinti praktikas, aprašytas įvykdytuose veiksmuose [46].

Proceso brandos modelis vaizduoja programinės įrangos proceso tobulinimo „sveiko proto projektavimo“ principą [46]. Proceso brandos modelis numato konceptualią metodiško ir nuoseklaus programinės įrangos produktų kūrimo ir valdymo struktūrą. Jis apibrėžia programinės įrangos brandos praktikas ir efektyvaus programinės įrangos proceso charakteristikas.

2002 m. programinės įrangos projektavimo institutas paskelbė naują CMMI (angl. *Capability Maturity Model Integration*) modelį, patikslinantį ir patobulinantį CMM modelį bei atitinkantį bendruosius programinės įrangos valdymo tarptautinių standartų reikalavimus. Šiame modelyje pabrėžiamas programinės įrangos kūrimo procesų valdymo tobulinimas, jų aukštos kokybės ir konkurencingumo užtikrinimas. Patobulinimų tikslas –projektų procesus padaryti labiau valdomus, o jų rezultatus – prognozuojamus. nemažas CMMI dėmesys skiriamas kūrimo procesams ir užsakovų reikalavimams bei jų projekto funkcijų, komponentų ir dokumentų atitikimui.

CMMI modelio pagrindą sudaro CMM brandos lygiai. Pirmas ir penktas lygiai pasižymi procesų nestabilumu ir neapibrėžtumu, todėl, detalizuojant procesų turinį, patariama apsiriboti trimis vidutiniais lygiais. Kiekviename aukštesniame lygyje rekomenduojama naudoti visus ankstesniojo lygio procesus.

IŠVADOS

Per pastaruosius penkiasdešimt metų išsivysčiusiose šalyse projektų vadyba reikšmingai išaugo, tapo pripažinta ir susiformavo kaip savarankiška mokslo sritis. **Projektų vadyba** – tai žinių, įgūdžių, priemonių ir technologijų visuma, padedanti projektui atitikti visus iškeltus reikalavimus ir patenkinti projekto dalyvių poreikius.

Projekto vykdymo metu funkcionuoja devynios žinių sritys, iš kurių kaip vieną svarbiausių reikėtų išskirti projektų kokybės vadybą, turinčią didelę įtaką galutiniam projekto rezultatui. **Projektų kokybės vadyba** – tai veiklos būdų ir metodų, naudojamų siekiant įgyvendinti užsakovo pateiktus kokybės reikalavimus visuma.

Įgyvendinant projektus, jų vykdytojai turi atsižvelgti į organizacijos kokybės politiką. Kokybės politikos užtikrinimas gali būti paremtas vienu iš kokybės užtikrinimo metodų: ISO kokybės standartais, Visuotinės kokybės vadyba, Šeši sigma metodologija ir kt. **ISO 9000 kokybės vadybos sistemos standartų šeimos** tikslas – skatinti kokybišką mąstymą ir naudotis geriausia patirtimi. **Šeši sigma metodologija** yra aukščiausios kokybės siekimo metodologija. **Visuotinės kokybės vadyba** – tai tokia vadybos filosofija ir metodai, kuriuos pasirinkusi organizacija nuolat tobulėja, įtraukdama į tobulinimo veiklą darbuotojus ir siekdama kuo geriau patenkinti vartotojų poreikius, gerindama produktų kokybę ir mažindama kaštus.

Programinės įrangos kokybės sąvokos apibrėžimas priklauso nuo daugelio aspektų, tokių kaip programinės įrangos tipas, naudojimo aplinkybės ir kt. **Programinės įrangos kokybė** – tai klientų ar vartotojų poreikių ar lūkesčių patenkinimo laipsnis. Aukštos kokybės programinė įranga apima tokias charakteristikas: **funktionalumą** (funkcijų, kurie programinei įrangai padeda patenkinti nustatytus ar numatomus vartotojų poreikius ir atitikti specifikacijas, visuma), **patikimumą** (atributų rinkinys, nusakantis, kad programinė įranga nustatytą laiką išlaiko savo funkcionavimo kokybės lygį pagal griežtai apibrėžtas sąlygas), **tinkamumą** (programinės įrangos dabartinių ir galimų vartotojų poreikių atitikimo laipsnis), **efektyvumą** (efektyvus atitinkamų išteklių panaudojimas vykdant nustatytas funkcijas apibrėžtomis sąlygomis), **palaikymo galimybę** (programinės įrangos pakeitimo darbų atlikimo galimybė) ir **mobilumą** (galimybė perkelti programinę įrangą iš vienos aplinkos į kitą).

Programinės įrangos kokybei įtaką daro įvairios vidinės produkto savybės, programinės įrangos ir jų vartotojų sąveika bei bendros produkto, kūrimo proceso ir aplinkos savybės. Išskiriamos keturios produkto kokybės metrikos: vidutinis gedimų intervalas, defektų tankis, problemos, su kuriomis susiduria klientai, ir klientų pasitenkinimas. Programinės įrangos kokybės parametrai gali kisti priklausomai nuo organizacijos požiūrio.

Sukurta daugybė modelių, aprašančių programinės įrangos kūrimą, pavyzdžiui, „krioklio“, spiralinis, prototipinis ir kiti modeliai. **„Krioklio“ modelis** formalizuoja programinės įrangos kūrimo fazes, pabrėžiant išankstinius reikalavimus ir kūrimo veiklas bei dokumentų kūrimą pradinėse fazėse. **Spiralinį modelį** sudaro santykinai tolygių ir organizuotų programinės įrangos kūrimo žingsnių sekos. **Prototipinis modelis** naudoja nustatytų reikalavimų vykdymo strategiją, kurios metu išskiriami, fiksuojami ir tenkinami vartotojų poreikiai naudojant galutinio programinės įrangos produkto darbo modelį.

Programinės įrangos kokybės užtikrinimą galima pasiekti naudojant šiam tikslui skirtus tarptautinius modelius ir standartus. Tokie modeliai yra ISO 9000 standartų šeima, visuotinės kokybės sistema, Programinės įrangos kūrimo proceso brandos modelis ir Programinės įrangos proceso tobulinimo sistema.

Programinės įrangos proceso tobulinimo sistema remiasi programinės įrangos įvertinimo procesu ir tobulinimo priemonėmis bei Demingo ciklu, kurį sudaro keturios tobulinimo proceso veiklos: planuok, daryk, tyrinėk, veik. Programinės įrangos proceso tobulinimo sistema apima šešias fazes – problemos, priežasčių, sprendimo, realizavimo, priežiūros ir pakeitimo. Nors programinės įrangos kūrimo ir gamybos procesai turi daug skirtumų, gamybos proceso tobulinimo principus ir praktiką galima peržiūrėti ir įvesti į programinės įrangos proceso tobulinimą.

Programinės įrangos proceso brandos modelis aprašo programinės įrangos proceso brandos principus ir praktikas bei padeda programinės įrangos kūrimo organizacijoms pagerinti savo programinės įrangos procesų brandą ir nueiti evoliucinį kelią nuo chaotiškų procesų prie subrendusių, metodiškų programinės įrangos kūrimo procesų. Jį sudaro penki brandos lygiai (pradinis, kartojamas, apibrėžtas, valdomas ir optimizavimo), kai kiekvienas iš jų yra aiškiai nustatytas evoliucinis brandos stabilizavimas link programinės įrangos procesų brandos.

Taigi organizacijos, norėdamos pasiekti aukštą programinės įrangos kokybės lygį, savo veikloje turi taikyti atitinkamą kokybės valdymo sistemą ar metodą, atsižvelgiant į specifinius programinės įrangos kūrimo procesus.

BIBLIOGRAFINIŲ NUORODŲ SĄRAŠAS

1. A Guide to the Project Management Body of Knowledge. PMBOK Guide. 2000 Edition. Project Management Institute. Pennsylvania: Project Management Institute, 2000. 210 p.
2. ABDALLAH, M. A quality assurance model for an information system development life cycle. International Journal of Quality & Reliability Management, Vol. 13 No. 7, 1996, p. 23-35.
3. ADOMĖNAS, V. Procesinio požiūrio taikymas kokybės vadybos sistemoje. Organizacijų vadyba: sisteminiai tyrimai: 2002. Nr. 20. 7-14 p.
4. ASSOCIATION OF PROJECT MANAGEMENT: Glossary of Project Management Terms [interaktyvus]. [Žiūrėta 2007 m. sausio 19 d.]. Prieiga per internetą: <http://www.apm.org.uk/PtoQ.asp>.
5. AZZOPARDI, S. The Evolution of Project Management. Project smart. [interaktyvus]. [Žiūrėta 2007 m. vasario 12 d.]. Prieiga per internetą: http://www.projectsmart.co.uk/pdf/evolution_of_project_management.pdf.
6. BABU, A., SURESH, N. Modelling and optimizing software quality. International Journal of Quality & Reliability Management, Vol. 13 No. 3, 1996, p.95-103.
7. BATE, R., KUHN, D., WELL, C. ir kt. A Systems Engineering Capability Maturity ModelSM, Version 1.1. Systems Engineering Capability Maturity Model Project. Software Engineering Institute. 1995.
8. BLUNDELL, J., HINES, M., STACH, J. The measurement of software design quality. Annals of Software Engineering 4 (1997) p. 235–255.
9. CARROLL, J. The application of total quality management to software development. Information Technology & People, Vol. 8 No. 4, 1995, p. 35-47.
10. CHARVAT, J. Project Management Methodologies: Selecting, Implementing, and Supporting Methodologies and Processes for Projects. New Jersey: John Wiley & Sons, 2003. 264 p.
11. European ICT market growing by a comfortable three percent in 2006. European Information Technology Observatory Press Release 2006. [interaktyvus]. [Žiūrėta 2007 m. kovo 19 d.]. Prieiga per internetą: <http://www.eito.com/download/Press%20release%20EITO%20Update%2012%2010%202006%20engl.pdf> >.
12. FUTRELL, R., Shafer, D., Safer, L. Quality Software Project Management. Prentice Hall PTR, 2002. 1677 p.
13. GAO, J., TSAO, H., WU, Y. Testing and Quality Assurance for Component-Based Software. Boston, London. Artech House. 2003.

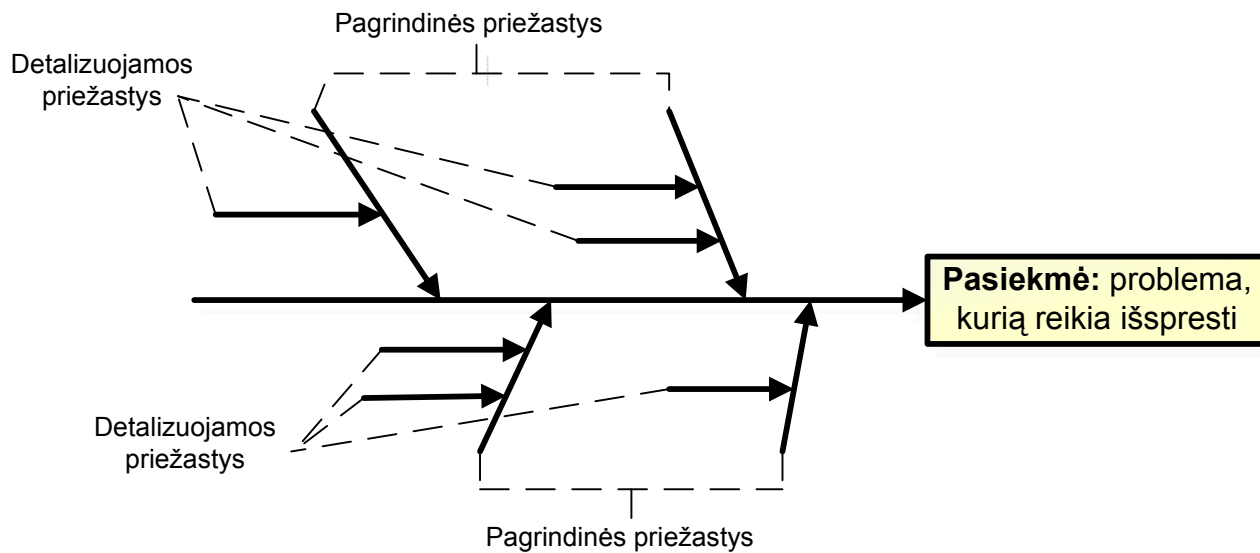
14. General systems theory. A Dictionary of Psychology. Andrew M. Colman. Oxford University Press, 2006. Oxford Reference Online. Oxford University Press. Vilnius University. 17 March 2007. [interaktyvus]. [Žiūrėta 2007 m. vasario 12 d.]. Prieiga per internetą: <<http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t87.e3445>>
15. GONG, B., YEN, D., CHOU, D. A manager's guide to total quality software design. Industrial Management and Data Systems. 98/3. 1998. p. 100-107.
16. Government-Industry – Academia Partnership in Software Process Improvement. Baltic IT&T review. A Business Journal for the Information Society. 2006, 1 (40), p 45-50.
17. GUPTA, P. Šešios sigmos verslo sėkmei valdyti. Iš anglų kalbos vertė Jonas Adolfas Patriubavičius. Vilnius : Vaga, 2005, 238, [2] p.
18. HE, Z., STAPLES. G., ROSS, M., COURT, I. Fourteen Japanese quality tools in software process improvement. The TQM Magazine. Volume 8 · Number 4 · 1996 · p. 40–44.
19. HELIO YANG, Y. Software quality management and ISO 9000 implementation. Industrial Management and Data Systems. 101/7. 2001. p. 329-338.
20. HOYLE, D. ISO 9000:2000: an A-Z guide. Oxford: Butterworth-Heinemann, 2003. 155 p. ISBN 0-7506-5844-4.
21. HONG, G., GOH, T. Six Sigma in software quality. The TQM Magazine Volume 15 · Number 6 · 2003 · p. 364-373.
22. HUMPHREY, W. The Personal Software Process. Technical Report. Team Software Process Initiative, 2000.
23. ICT markets, March 2006. European Information Technology Observatory. [interaktyvus]. [Žiūrėta 2007 m. kovo 19 d.]. Prieiga per internetą: <<http://www.eito.com/download/EITO%202006%20-%20ICT%20market%20March%202006.pdf>>.
24. ICT markets, March 2007. European Information Technology Observatory. [interaktyvus]. [Žiūrėta 2007 m. kovo 19 d.]. Prieiga per internetą: <<http://www.eito.com/download/EITO%202007%20-%20ICT%20markets,%20March%2020071.pdf>>.
25. ISO in brief. International Standards for a sustainable World. [interaktyvus]. [Žiūrėta 2007 m. sausio 6 d.]. Prieiga per internetą: <http://www.iso.org/iso/en/prods-services/otherpubs/pdf/isoinbrief_2005-en.pdf>.
26. YETTON, P., MARTIN, A., SHARMA, R., JOHNSON, K. A model of information systems development project performance. Info Systems J. 2000 Nr. 10, p. 263-289.
27. JOVANOVIC, V., SHOEMAKER, D. ISO 9001 standard and software quality improvement. Benchmarking for Quality Management & Technology, Vol. 4 No. 2, 1997, p.148-159.
28. JURKAUSKAS, A. Visuotinės kokybės vadyba: [mokomoji knyga]. Kaunas: Technologija, 2003. 226, [1] p.

29. KAN, S. Metrics and Models in Software Quality Engineering, Second Edition. Addison Wesley, 2002, p. 560.
30. KAUTZ, K., LARSEN, E. Diffusion theory and practice. Disseminating quality management and software process improvement innovations. Information Technology & People, Vol. 13 No. 1, 2000, p. 11-26.
31. KERZNER, H. Project Management. A System Approach to Planning, Scheduling, and Controlling. Seventh Edition. New York: John Wiley & Sons, 2001. 1232 p.
32. KHOSHGOFTAAR, T. A Multiobjective Module-Order Model for Software Quality Enhancement. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 8, NO. 6, DECEMBER 2004.
33. Kokybės vadybos poveikis šalies ūkiui, Lietuvai įstojus į Europos Sąjungą. Respublikinės konferencijos pranešimų medžiaga. 2004 m. lapkričio 24 – 25 d. Kaunas: Technologija, 2004.
34. Kokybės vadybos sistemos. Pagrindai, terminai ir apibrėžimai (ISO 9000:2000). Vilnius: Lietuvos standartizacijos departamentas, 2001. 84 p.
35. Kokybės vadybos sistemos. Reikalavimai (ISO 9001:2000). Vilnius: Lietuvos standartizacijos departamentas, 2001. 61 p.
36. Kokybės vadybos sistemos. Veiklos gerinimas. Rekomendacijos (ISO 9004:2000). Vilnius: Lietuvos standartizacijos departamentas, 2001. 123 p.
37. KRISHNAN, M. The role of team factors in software cost and quality. An empirical analysis. Information Technology & People, Vol. 11 No. 1, 1998, p. 20-35.
38. LANDRY, P. The ISO 9000:2000 Essentials. A practical handbook for implementing the ISO 9000 Standards. 3rd Edition. Canadian Standards Association. Toronto, Canada. 2001.
39. LEE, K. A Top-Down Approach to Quality Driven Architectural Engineering of Software Systems. IEICE TRANS. INF. & SYST., VOL.E88–D, NO.12 DECEMBER 2005.
40. MARASH, S. SIX SIGMA: A Quality Philosophy for the Next Millennium. Budapest: EQO Congress Proceedings, 2000.
41. MCADAM, R., FULTON, F. The impact of the ISO 9000:2000 quality standards in small software firms. Managing Service Quality. Volume 12. Number 5. 2002, p. 336-345.
42. MISRA, S., BHAVSAR, V. Object-oriented software development antecedents that influence product bug density. Lessons for quality managers. International Journal of Quality & Reliability Management. Vol. 22 No. 2, 2005. p. 162-182.
43. NEVERAUSKAS, B., STANKEVIČIUS, V., VILIŪNAS, V., ČERNIŪTĖ, I. Projektų valdymas. Mokomoji knyga. Kaunas: Technologija, 2003. 143, [1] p.
44. OGASAWARA, H., YAMADA, A., KOJO, M. Experiences of Software Quality Management Using Metrics through the Life-Cycle. IEEE, 1996.

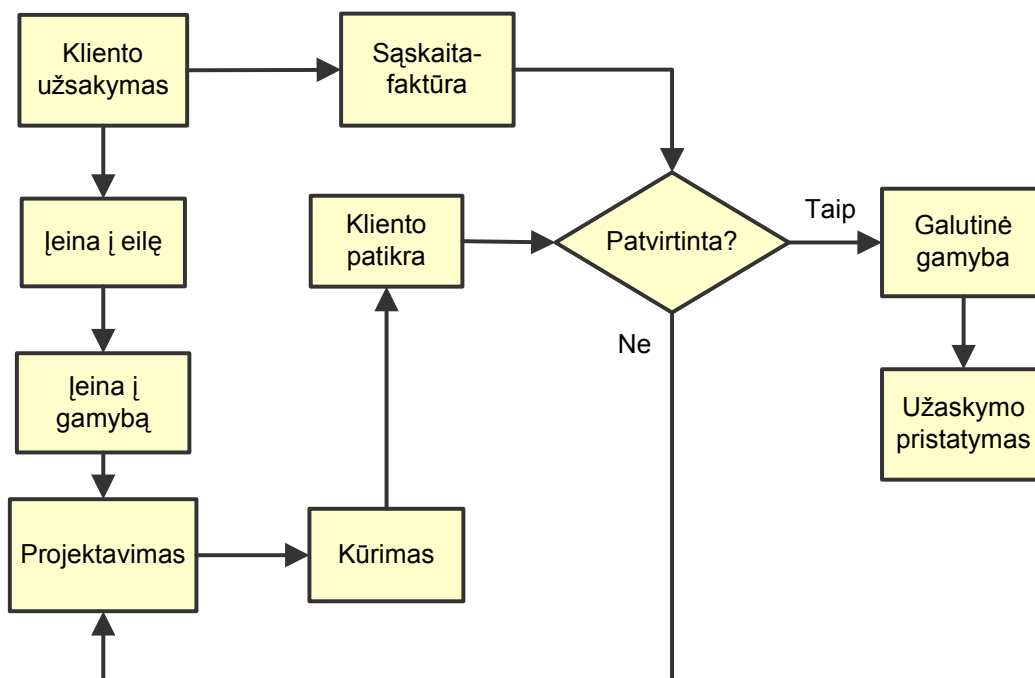
45. PAULK, M. A Comparison of ISO 9001 and the Capability Maturity Model for Software. Software Capability Maturity Model Project. Software Engineering Institute. Pittsburgh, Pennsylvania. 1994.
46. PAULK, M., WEBER, C., GARCIA, S., CHRISSIS, M., BUSH, M. Key Practices of the Capability Maturity Model SM, Version 1.1. Technical Report. Software Engineering Institute, Pittsburgh, Pennsylvania. 1993.
47. PESKIN, M., HART, J. Measuring the quality of computer systems development. Benchmarking for Quality Management & Technology, Vol. 3 No. 2, 1996, p. 68-84.
48. PFLEEGER, S., HATTON, L., HOWELL, C. Solid Software. Prentice-Hall, Upper Saddle River, NJ. 2002.
49. POCIŪTĖ, D., JANUŠAUSKIENĖ, V., VITKAUSKAS, R. Kokybės vadyba: mokomoji knyga. Vilnius: Technika, 2004. 218 [1] p.
50. RUBEŽIUS, J. Kokybės vadybos ir žinių vadybos sąsajų tyrimas. Informacijos mokslai. 2005. Nr. 35. 27 – 58 p.
51. RUTKAUSKAS, J. Šeši sigma koncepcijos taikymas kokybės vadyboje. Organizacijų vadyba: sisteminiai tyrimai: 2003. Nr. 23. 212-213 p.
52. TANAKA, T., AIZAWA, M., OGASAWARA, H., YAMADA, A. Software Quality Analysis & Measurement Service Activity in a Company. IEEE, 1998.
53. TENNANT, G. Six sigma. SPC and TQM in Manufacturing and services. Aldershot : Gower, 2001. 140 p.
54. TIAN, J. Software Quality Engineering. Testing, Quality Assurance and Quantifiable Improvement. Wiley-IEEE Computer Society Press, Hoboken, New Jersey. 2005, p.440.
55. TOMAŠAITIS, R. Projekto vadybos metodiniai nurodymai. Metodinė knyga. Vilnius: Technika, 2004. 35 p.
56. Top Players in the baltic ICT market. Baltic IT&T review. A Business Journal for the Information Society. 2006, 3 (42), p 45-47.
57. TRUSCOTT, W. Six Sigma: Continual Improvement for Businesses: A Practical Guide. Oxford: Butterworth-Heinemann, 2003. 272 p.
58. VANAGAS, P. Visuotinės kokybės vadyba: vadovėlis. Kaunas: Technologija, 2004. 426 [2] p.
59. WALKER, A., GEE, C. ISO 9001 model support for software process assessment. Logistics Information Management. Volume 13, Nr. 1, 2000, p. 39-44.
60. Термины, относящиеся к системе качества. Терминологический словарь [interaktyvus]. [Žiūrėta 2007 m. sausio 29 d.]. Prieiga per internetą: <<http://cert.stankin.ru/slovar/2.html>>.

PRIEDAI

1 priedas. Struktūrinių schemų pavyzdžiai

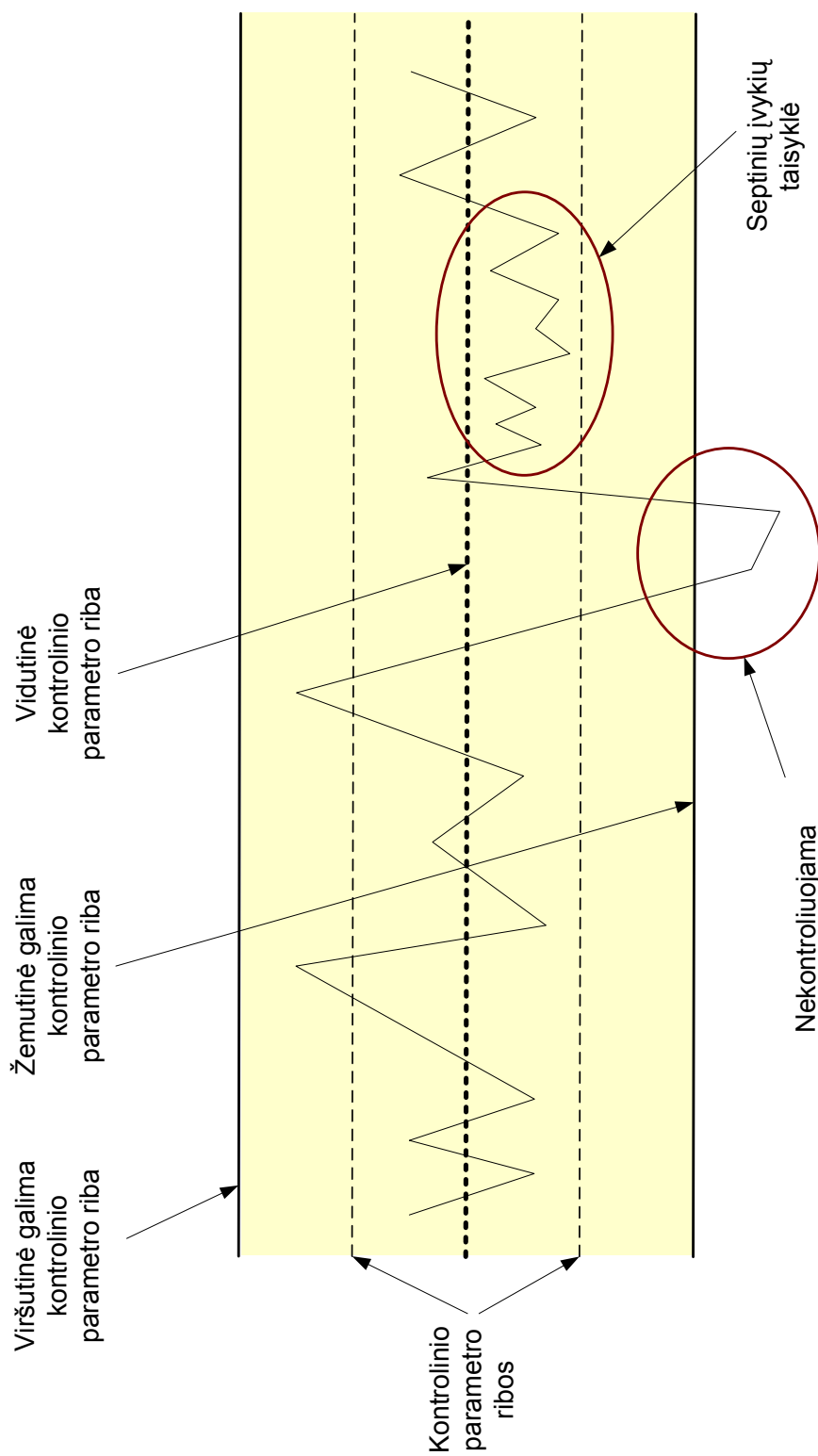


a. Priežasčių - pasekmių diagrama

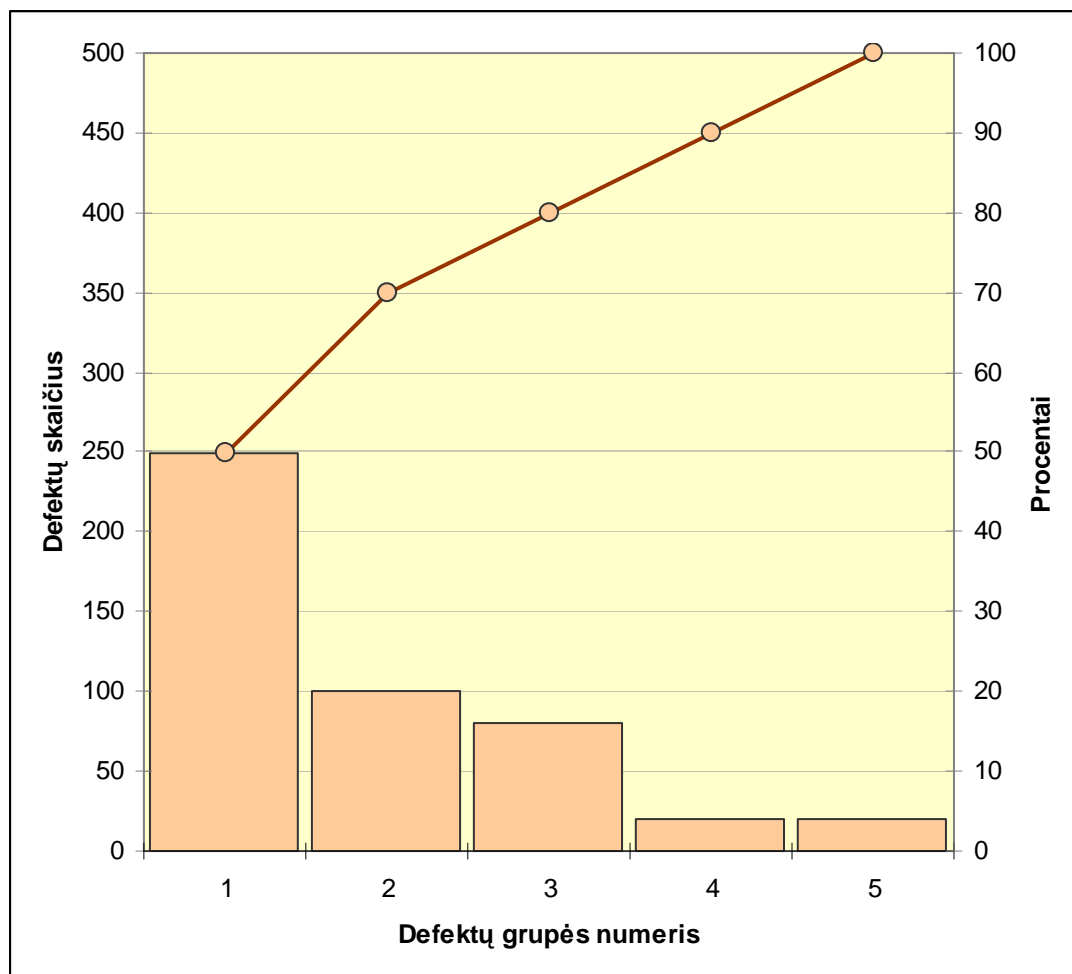


b. Proceso srauto diagrama

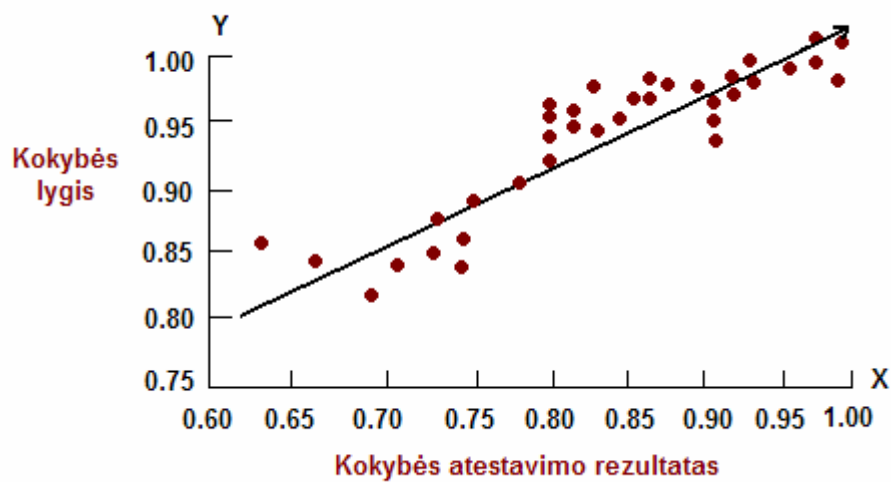
2 priedas. Kontrolinių diagramų pavyzdys



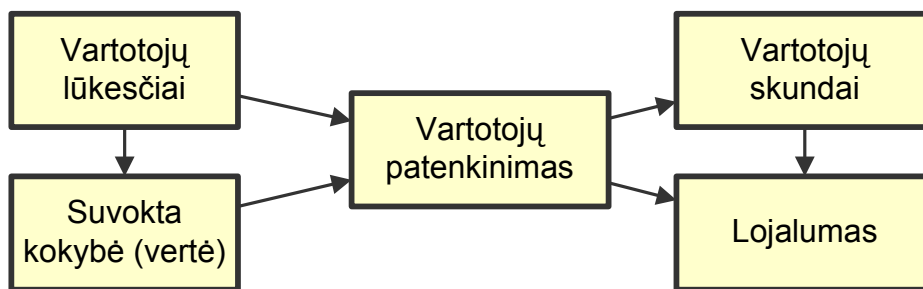
3 priedas. Pareto diagramos pavyzdys



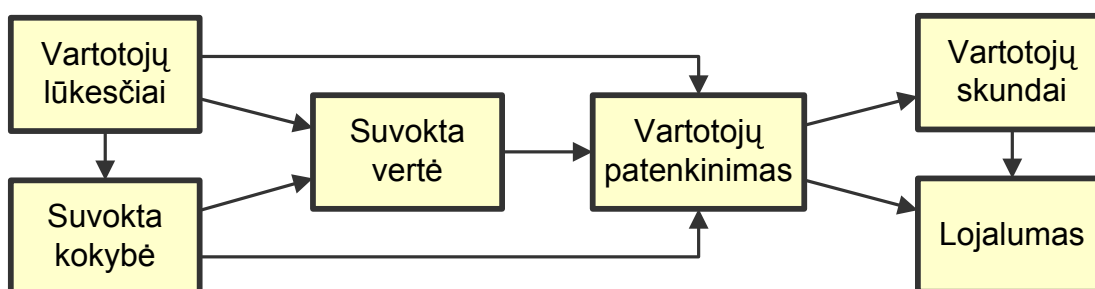
4 priedas. Trendo analizės grafinio vaizdavimo pavyzdys



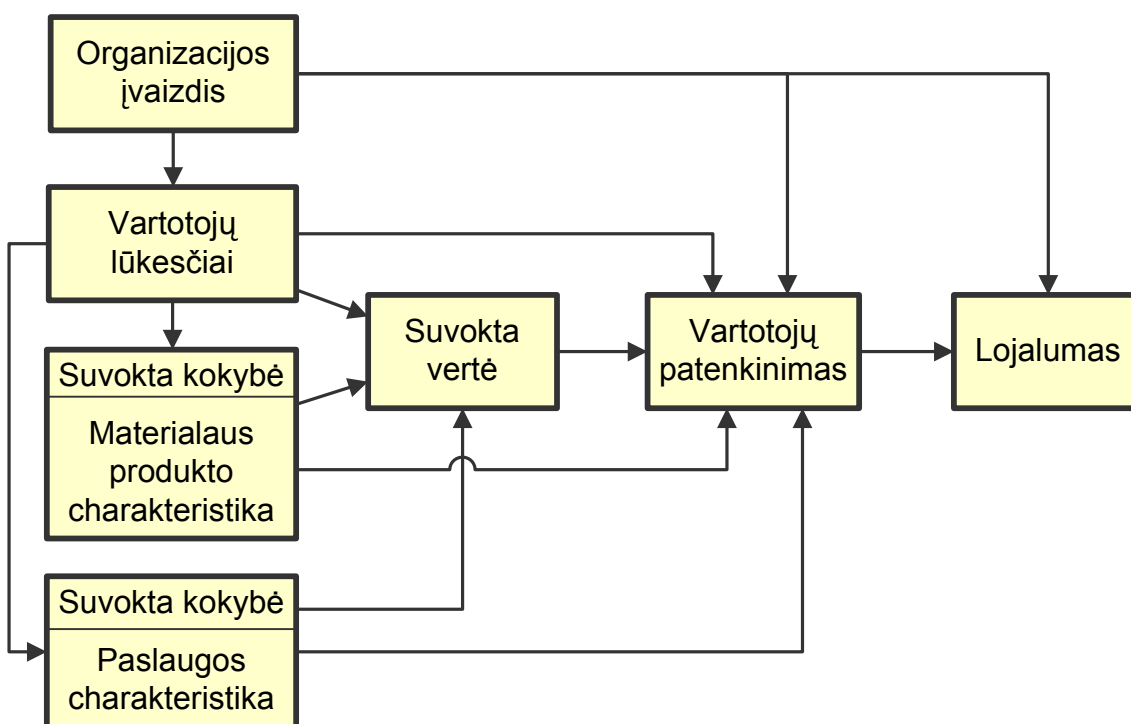
5 priedas. Vartotojų poreikių tenkinimo tyrimo modeliai



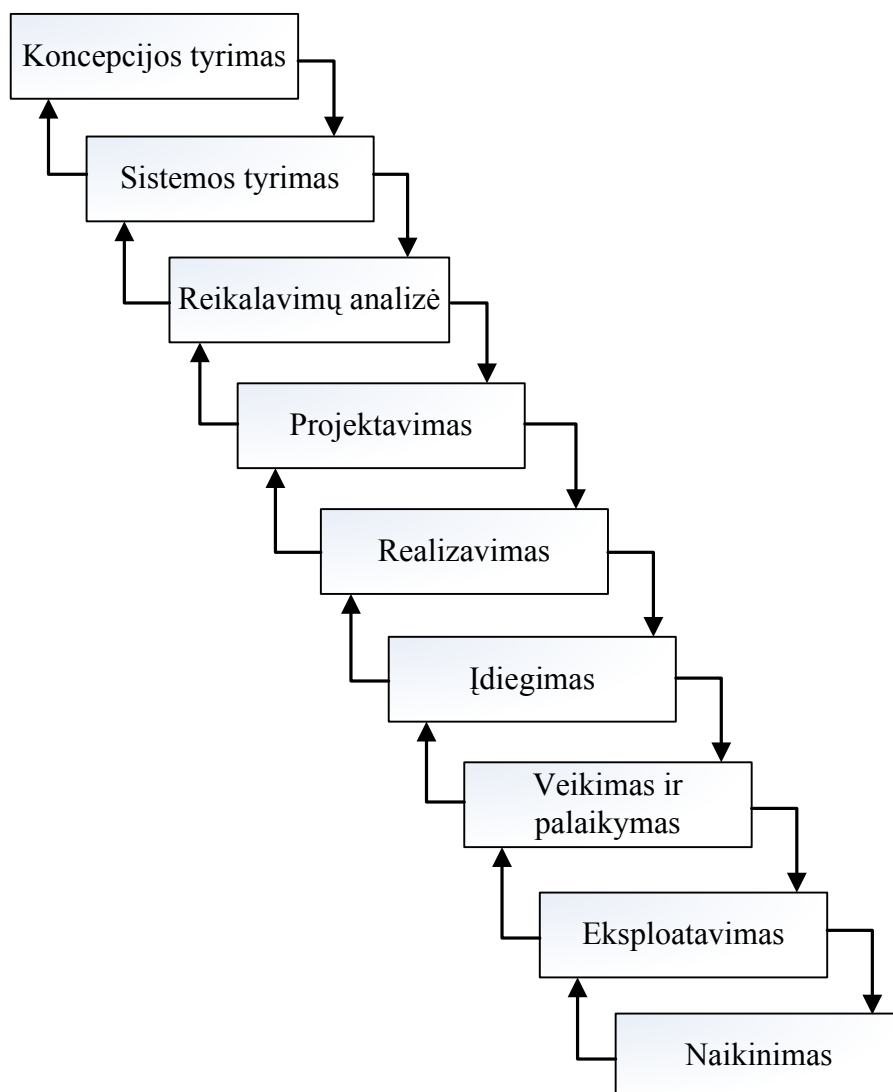
a. Švedijos vartotojų patenkinimo barometro modelis



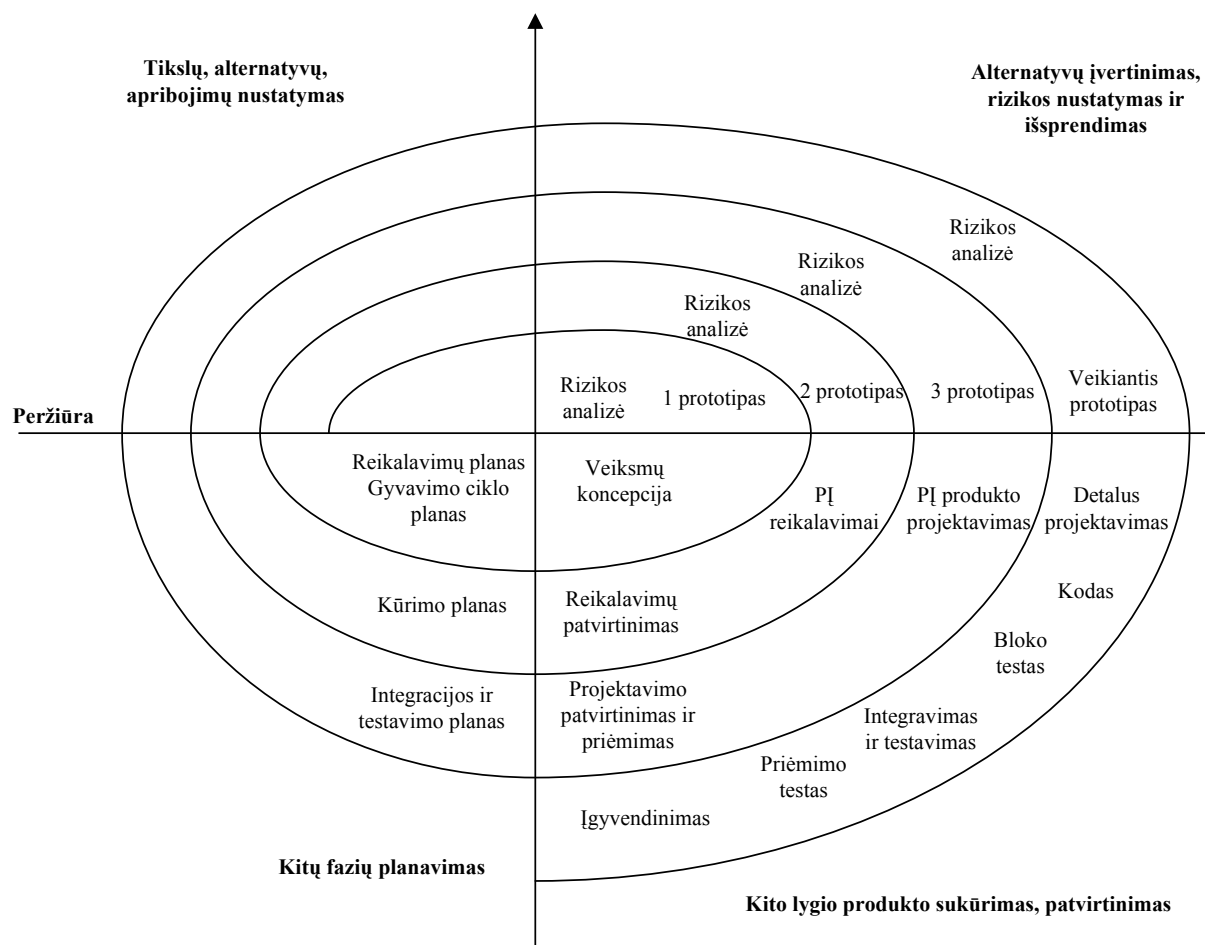
b. Amerikos vartotojų patenkinimo indekso modelis

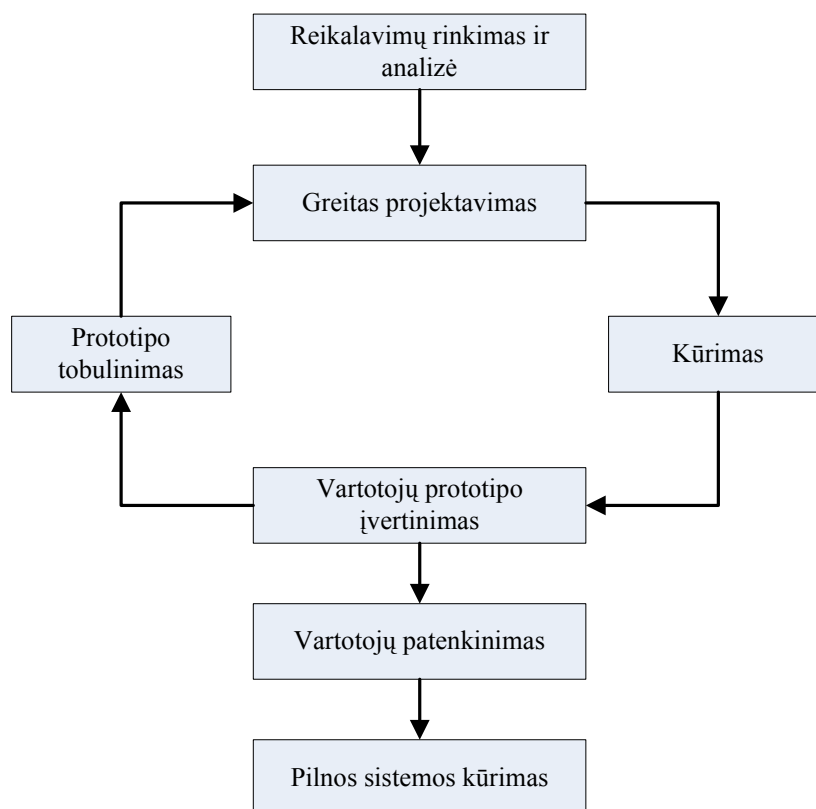


c. Europos vartotojų patenkinimo indekso modelis

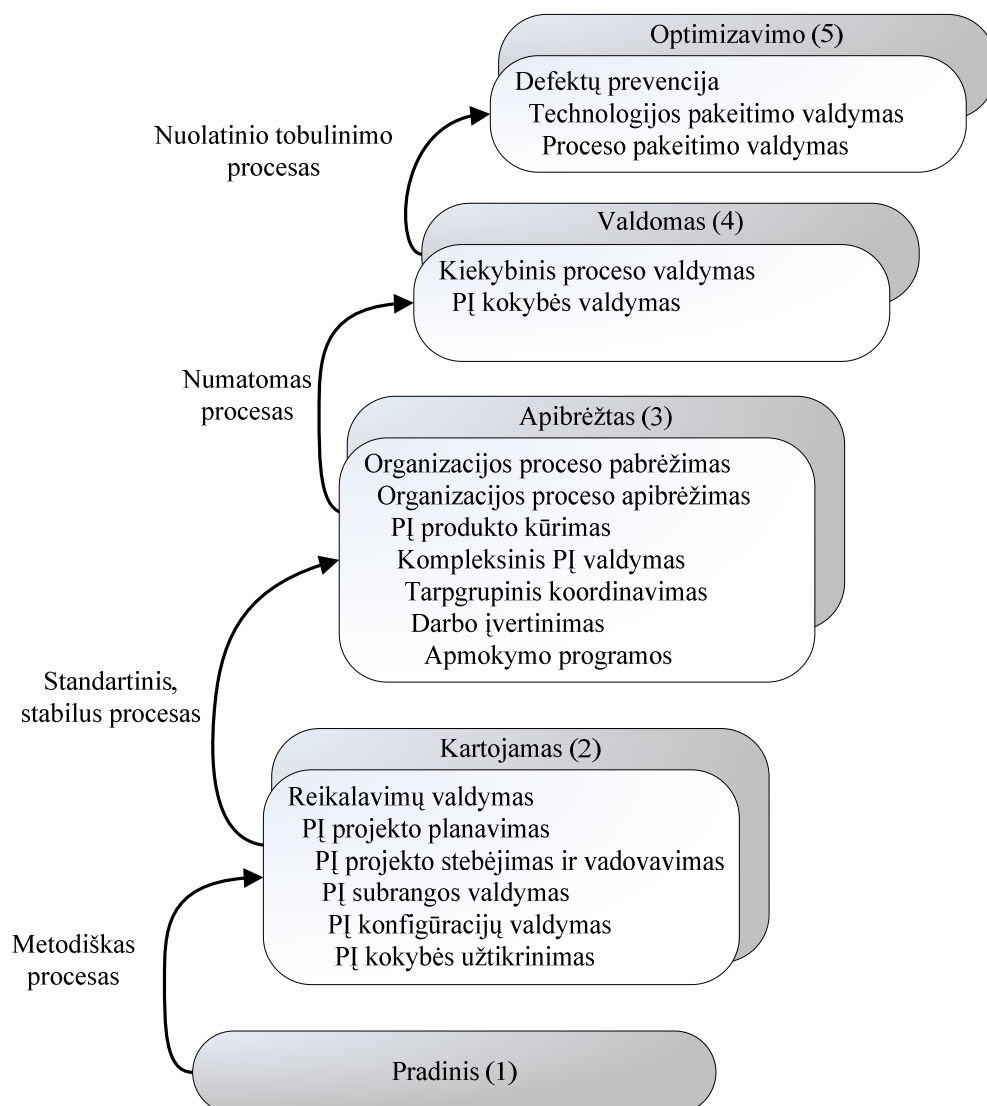
6 priedas. Krioklio modelis

7 priedas. Spiralinis modelis



8 priedas. Prototipinis modelis

9 priedas. Pagrindinės PBM brandos lygių proceso sritys



SUMMARY

Afanasjeva V. Software development projects quality management

Nowadays project management has been accepted as a project activities methodology in the whole world. Organizations can reach a high quality level, gain time and resources, understate risks and increase reliability of its product and services by using project management techniques and methods. The importance of project management is specifically important in software development projects. Without project management and its particular knowledge areas, such as quality management, the result of software development projects instead of benefit can cause significant direct and indirect costs or even fail.

Many organizations wishing to improve their business activities and processes are investing a lot of money into software development or procurement. The software industry has been growing at an accelerated rate for the past decade (in 1996, annual revenues of information technology market estimated \$120 billion, whereas in 2006 – €919 billion). In order to develop high quality software products, which can satisfy all users' needs and requirements at the maximum, software development process should be taken into consideration, due to that quality occurs at project initiation and must be planned for within the project. Project quality management combines interdependent methods, techniques and technologies, and helps to achieve high quality software products.

The recognition and growth of project management has changed meaningfully during last fifty years. Project management became an independent scientific field. Project management is the application of knowledge, skills, tools and techniques to project activities. It covers nine knowledge areas, such as project quality management, which includes processes required to ensure satisfaction of customers needs and requirements.

It is important for the organization to have a quality policy. The assurance of quality policy can be maintained by one of the quality assurance methods: ISO standards, Total quality management, Six sigma methodology. The aim of ISO quality management standards is to motivate quality thinking and to use best practices.

Software quality is the totality of characteristics (functionality, reliability, usability, efficiency, maintainability and portability) of an entity that bears on its ability to satisfy and implement needs. There are a lot of models describing the whole software process like waterfall model, spiral model, prototyping models, etc.

The assurance of software quality can be achieved with the help of international models and standards, like ISO 9000 standards, Capability Maturity Model for Software, Software process improvement system, etc. Regardless of which model or standard is used, the degree to which it is implemented varies from organization to organization and even from project to project.