

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Programų kūrimo proceso gerinimas labai mažose įmonėse

Software Process Improvement in Micro Companies

Magistro baigiamasis darbas

Darbą atliko: Oskaras Puipa (parašas)

Darbo vadovė: Laura Savičienė (parašas)

Recenzentas: Doc. dr. Saulius Ragaišis (parašas)

Vilnius – 2010

Santrauka

Šio darbo tikslas yra pateikti rekomendacijas labai mažų įmonių programų kūrimo proceso vertinimui ir gerinimui. Darbe apžvelgti mažoms įmonėms skirti proceso vertinimo metodai ir nagrinėjama, kaip jie galėtų būti adaptuoti labai mažos įmonės proceso vertinimui. Pasirinktas metodas ir siūlomi jo pakeitimai pritaikyti atliekant labai mažos informacinių technologijų įmonės programų kūrimo proceso vertinimą. Remiantis vertinimo metu sukaupta patirtimi apibendrintos rekomendacijos labai mažos įmonės programų kūrimo proceso vertinimui. Jos gali būti panaudotos kuriant naujus ar tobulinant esamus vertinimo metodus.

Išanalizavus vertinimo metodus, nuspręsta modifikuoti MARES metodą labai mažos įmonės vertinimui. Šis metodas yra grindžiamas ISO 15504 standartu ir skirtas mažoms įmonėms. Pasiūlyta kelis MARES procesus atlikti kitaip, papildyti tikslų-procesų modelį. Naujasis tikslų-procesų modelis gali būti naudojamas greitam vertinimo apimties nustatymui, tačiau jį dar reikėtų vystyti, kad būtų tinkamas platesniam labai mažų įmonių ratui.

Po vertinimo nustatytos silpnosios procesų vietos ir sudarytas gerinimo planas remiantis judriosiomis metodikomis. Įvykdytas pirmasis gerinimo žingsnis, siekiant parodyti, kad judriųjų praktikų naudojimas galimas su formaliumi įmonės procesų vertinimu.

Raktažodžiai: ISO 15504, proceso vertinimas, MARES, vertinimo metodai, judriosios praktikos.

Summary

The goal of this work is to provide recommendations for software process assessment and improvement in micro companies. Assessment methods aimed for small companies are reviewed and their applicability for micro companies is analyzed. One chosen method is modified and process assessment is performed in a micro software company. Based on the assessment experience, the recommendations for process assessment in micro companies are summarized. These recommendations might be used when creating new software process assessment methods or improving existing ones.

After method analysis it is decided to modify MARES method by adapting it for software process assessment of micro company. MARES method is based on ISO 15504 standard and is intended for small company assessment. Changes in MARES method are introduced, some processes are changed, goal-process model is complemented. New goal-process model might be used for quick definition of the assessment scope.

In the result of assessment, software process weaknesses are identified and a few processes chosen for improvement. Decision to use agile testing practices for improvement was made. First step of micro company software process improvement is completed demonstrating that usage of agile methods along with formal software process assessment is possible.

Keywords: ISO 15504, process assessment, process improvement, MARES, assessment methods, agile practices.

Turinys

Įvadas.....	6
1. Programų kūrimo proceso vertinimas	11
1.1. ISO/IEC 15504 standartas	12
1.2. ISO 15504 grindžiami vertinimo metodai mažoms įmonėms	14
1.2.1. MARES metodas	14
1.2.2. RAPID metodas	16
1.2.3. QuickScan metodas	17
1.2.4. TOPS metodas	18
1.2.5. Apibendrinimas	19
2. Judriosios PĮ kūrimo metodikos	20
2.1. Scrum.....	20
2.2. Ekstremalus programavimas (XP).....	21
2.3. Judrusis darbų vertinimas ir planavimas	23
2.4. Judrusis testavimas	24
2.5. Proceso gerinimas remiantis judriosiomis praktikomis.....	25
2.5.1. Reikalavimų išaiškinimas (ENG.1).....	27
2.5.2. Programinės įrangos projektavimas (ENG.5)	27
2.5.3. Programinės įrangos gamyba (ENG.6)	27
2.5.4. Programinės įrangos testavimas (ENG.8).....	28
2.5.5. Projektų valdymas (MAN.3)	28
2.5.6. Problemų sprendimas (SUP.9).....	28
2.5.7. Konfigūracijos valdymas (SUP.8)	28
3. Vertinimo metodų pritaikymas labai mažoms įmonėms	29
3.1. Vertinimo metodų analizė.....	29
3.1.1. Apimties nustatymas	29
3.1.2. Dydis ir sudėtingumas	30
3.1.3. Plečiamumas	30
3.1.4. Viešas prieinamumas	30
3.1.5. Vidinio vertinimo galimybė.....	30
3.1.6. Metodo naujumas	31
3.1.7. Apibendrinimas	31
3.2. Vertinimo metodo adaptavimas	32
3.2.1. Įmonės vertinimo planavimo procesas	33
3.2.2. Kontekstualizacija	34
3.2.3. Vertinimas	36
3.2.4. Apibendrinimas	38
4. Vertinimo metodo taikymo patirtis	39
4.1. Kontekstualizacijoje naudojamo modelio pakeitimas	39
4.2. Procesų vertinimas	46
4.2.1. Programinės įrangos gamybos proceso (ENG.6) vertinimas	46
4.2.2. Projektų valdymo proceso (MAN.3) vertinimas	47
4.2.3. Vertinimo rezultatai	49
4.3. Vertinimo patirtis	51
5. Rekomendacijos labai mažų įmonių programų kūrimo proceso vertinimui	53
6. Labai mažos įmonės programų kūrimo proceso gerinimas.....	54
6.1. Gerinimo planavimas	54
6.1.1. Detalus gerinimo planas	55
6.1.2. Gerinamų procesų padengimas pagal ISO 15504 naudojantis judriuoju testavimu	55
6.1.2.1. Programinės įrangos gamyba.....	55

6.1.2.2. Programinės įrangos integravimas	55
6.2. Pradinio gerinimo plano pakeitimas ir motyvai	56
6.3. Įrankių pasirinkimas	57
6.4. Automatinė integracija ir publikavimas	59
6.5. Bandomieji automatiniai testai	60
6.6. Gerinimo patirtis.....	61
Rezultatai ir išvados	62
Šaltiniai.....	64
Priedai.....	67
1 priedas. MARES vertinimo planavimo procesas.....	67
2 priedas. MARES kontekstualizacijos procesas	68
3 priedas. MARES vertinimo procesas	69
4 priedas. MARES tikslų-procesų modelis	70
5 priedas. Detalus gerinimo planas.....	72
6 Priedas. Programinių vienetų testavimo automatizavimas	73
7 priedas. Integracinio testavimo automatizavimas.....	75
8 priedas. Programinės įrangos gamybos padengimas po pagerinimo.....	76
9 priedas. Programinės įrangos integravimo padengimas po pagerinimo	78

Ivadas

Šio darbo objektas yra programų kūrimo proceso vertinimas ir gerinimas labai mažose įmonėse. Programų kūrimo proceso gerinimas yra svarbus visoms įmonėms, siekiančioms dirbti pelningai, tenkinti užsakovo reikalavimus ir išlikti rinkoje [RW07]. Proceso vertinimas – praktiškai neišvengiamas proceso gerinimo ciklo pirmasis žingsnis. Šiame darbe nagrinėjama, kaip programų kūrimo proceso vertinimo standartai ar metodai gali būti pritaikyti (angl. *scale down*) labai mažoms įmonėms, ir kaip galima gerinti labai mažos įmonės procesus. Į labai mažas įmones orientuojamasi dėl to, kad jos sudaro didelę rinkos dalį, o Lietuvoje – vyrauja tarp informacinių technologijų įmonių.

Nors netrūksta gerų pavyzdžių, kaip programų kūrimo proceso vertinimo ir gerinimo standartai ar metodai pritaikomi mažose įmonėse [AWSS04, ESP99, PGP07, RTCH00, WAS04, SKJ10], ši sritis dar nėra pakankamai išnagrinėta, nėra lanksčių metodų labai mažų įmonių vertinimui, nėra modelių, padedančių apibrėžti vertinimo apimtį.

Programų kūrimo proceso gerinimas

Programinė įranga yra paplitusi visose gyvenimo srityse. Nepaisant to, kad žmonės tampa vis labiau nuo jos priklausomi, programinės įrangos kūrimo projektai susiduria su tokiomis problemomis, kaip biudžeto viršijimas ar grafikų nesilaikymas [Hum92]. Kokybės valdyme besispecializuojančios „The Standish Group“ kompanijos duomenimis, nuo 1994 iki 2000 metų sėkmingų projektų padaugėjo nuo 16% iki 28% [TSG94, TSG00]. Taigi, tik mažiau negu trečdalis projektų baigiami sėkmingai. Tuo tarpu projektų, kurie turi problemų (viršytas biudžetas, vėlavimas, nepilnas funkcionalumas) procentas išlieka maždaug toks pats – apie 50% (pvz., 53% 1994 metais).

Programinės įrangos kokybė ir projektų sėkmė priklauso nuo programinės įrangos gamybos proceso. Kiek didesnio nei minimalaus sudėtingumo industrinio produkto kokybė gali būti užtikrinta, užtikrinant gamybos proceso kokybę [Hum89]. Kitaip tariant, norint pagerinti produkto kokybę, reikia gerinti proceso kokybę. Proceso gerinimas paprastai remiasi vertinimo ir gerinimo modeliais, standartais ir metodais (CMMI, ISO 15504, Bootstrap ir kt.).

Mažų ir vidutinių įmonių vaidmuo Europos ekonomikoje yra labai svarbus, jos sukuria 75 milijonus darbo vietų ir sudaro 99% visų įmonių [Eur05]. Programinės įrangos srityje šių įmonių vaidmuo taip pat yra labai svarbus [CS09]. 99,2% programinės įrangos įmonių yra mažos arba vidutinės [PGP07].

Labai mažos įmonės apibrėžimas

Bendru atveju nėra labai tikslaus apibrėžimo, kas yra labai maža įmonė. Dauguma autorių maža laiko įmonę, turinčią ne daugiau kaip 50 darbuotojų. MARES metodikos [WAS04] autoriai dar išskiria labai mažas (angl. *micro*) įmones, turinčias ne daugiau kaip 10 darbuotojų.

Europos komisijos publikuotame dokumente [Eur05] įmonių paskirstymas į kategorijas yra panašus:

- Mažiau nei 10 darbuotojų – labai maža įmonė.
- Mažiau nei 50 darbuotojų – maža įmonė.
- Mažiau nei 250 darbuotojų – vidutinė įmonė.

Toliau darbe maža įmone bus laikoma įmonė, turinti ne daugiau nei 50 darbuotojų, o labai maža įmone – turinti ne daugiau nei 10 darbuotojų. Dauguma autorių nenurodo, kaip skaičiuojami įmonės darbuotojai (ar į šį skaičių įtraukiama vadovybė, pagalbinių darbuotojai), o šiame darbe laikoma, kad labai maža informacinių technologijų įmonė turi iki 10 informacinių technologijų darbuotojų.

Asociacijos Infobalt duomenimis 2004 metais daugiau kaip 85% Lietuvos informacinių technologijų įmonių turėjo ne daugiau kaip 9 darbuotojus [IB04]. Taigi, galima teigti, kad dauguma Lietuvos informacinių technologijų įmonių patenka į „labai mažų“ kategoriją.

Dauguma metodų ir paminėtų tyrimų neišskiria labai mažų įmonių iš mažų įmonių aibės. Kadangi Lietuvoje labai mažos įmonės sudaro didelę informacinių technologijų įmonių dalį, labai mažų įmonių vertinimas ir gerinimas vertas atskiro dėmesio.

Programų kūrimo proceso gerinimas mažose ir labai mažose įmonėse

Tradiciškai laikoma, kad mažoms (kartu ir labai mažoms) įmonėms dauguma esamų proceso vertinimo ir gerinimo metodų yra per „sunkūs“, ir joms tinkamesni įvairūs judrieji metodai. Tačiau vertinimo metodų taikymas mažose įmonėse naudingas dėl kelių faktorių [AWSS04]:

- geresnis įmonėje vykstančių procesų supratimas dėl pateiktų vertinimo rezultatų ir įmonės darbuotojų diskusijų vertinimo metu,
- remiantis vertinimo modeliu identifikuotos silpnosios ir stipriosios procesų pusės,
- gerinimo veiksmai vykdomi tiems procesams, kurie įmonei aktualiausi,
- didesnė procesų gerinimo motyvacija todėl, kad yra geresnis procesų ir jų silpnųjų vietų supratimas.

[AWSS04] darbe buvo vertinamos 4 mažos įmonės, iš kurių 2 buvo labai mažos, todėl šie faktoriai tinkami ir labai mažoms įmonėms. Be to, vertinimas užima apie 80 žmogaus darbo

valandų, kurios apima įmonės darbuotojų ir vertintojų darbo valandas [AWSS04], o tiek gali skirti ir labai mažos įmonės. [Gar05] darbe įvardinami tokie vertinimo metodų taikymo privalumai:

- mažai įmonei augant didėja veiklos efektyvumo poreikis,
- jei įmonė gali parodyti, kad jos procesas atitinka modelio ar standarto reikalavimus, tai jai atsiranda galimybė gauti geresnius projektus/klientus.

Programų kūrimo proceso vertinimas

Įmonės turi atidžiai pasirinkti, kuriuos procesus gerinti. Kai įmonė apsibrėžia tikslus, kurie padėtų pasiekti sėkmę, reikia žinoti, kaip gerai įmonė vykdo savo veiklą, ir ką dar reikia padaryti sėkmei pasiekti. Kadangi šių laikų konkurenciniame pasaulyje sėkmė yra judantis taikynys (reikia visada būti greitesniu, geresniu, efektyvesniu), reikia nuolatos įvertinti iš naujo ir pagerinti veiklą, remiantis naujais tikslais [Loo07b]. Apibrėžiant ir vykdant procesus padedama įmonei siekti užsibrėžtų tikslų. Proceso vertinimo tikslas yra nustatyti esamą įmonės programų kūrimo proceso būseną, identifikuoti aukščiausio prioriteto proceso problemas ir padėti organizuojant proceso gerinimą [KH89].

Proceso vertinimas dažniausiai yra proceso gerinimo pirmasis žingsnis. Proceso vertinimas prieš gerinimą yra naudingas, kadangi [DM99]:

- padeda suprasti įmonės programų kūrimo praktikas ir kaip apskritai įmonė dirba;
- padeda surasti įmonės silpnąsias ir stipriąsias puses bei procesus, kuriems labiausiai reikalingas gerinimas;
- palengvina gerinimo pradžia;
- sukuria karkasą gerinimo veiksams;
- padeda gauti vadovybės palaikymą gerinimui.

Proceso vertinimui paprastai yra naudojami vertinimo modeliai, standartai ir metodai. Šiame darbe pasirinktas ISO 15504 standartas, kadangi šis standartas nenurodo, kaip turi būti įgyvendinami procesai, o padeda įvertinti, kaip gerai jie vykdomi [Loo07a]. Tai svarbu mažoms įmonėms, kadangi, mažose įmonėse procesų tikslai dažnai įgyvendinami nestandartinėmis priemonėmis, naudojamos alternatyvios praktikos. Be to, remiantis šiuo standartu yra sukurti keli vertinimo metodai, skirti mažoms įmonėms.

ISO 15504 pateikia vertinimo reikalavimus, tačiau nepateikia jų taikymo būdų mažose įmonėse [WAS04]. Siekiant pritaikyti standartą mažoms įmonėms, buvo sukurtos metodikos: MARES [WAS04], RAPID [RTCH00], TOPS [ESP99], QuickScan [Loo07b]. Nors šie metodai skirti mažoms įmonėms, jie gali būti taikomi ir labai mažose įmonėse.

Judriosios metodikos

Pastaruoju metu plinta judriosios (angl. *agile*) metodikos, besiremiančios iteratyviu programinės įrangos gyvavimo ciklo modeliu ir paprastomis „geromis praktikomis“. Šios metodikos didesnę dėmesį skiria individams ir bendravimui, o ne aiškiems procesams ir įrankiams, kadangi darniai dirbanti komanda, sudaryta iš motyvuotų asmenybių su vidutiniais įrankiais, gauna geresnį rezultatą nei komanda, sudaryta iš vidutinių žmonių ir naudojanti gerus įrankius ir procesus [Coh06].

Atrodytų, kad mažoms įmonėms puikiai tiktų judriosios metodikos, tačiau pilnas jų taikymas gali netikti dėl kelių priežasčių [Loo07b]:

- Augant įmonei judriąsias praktikas reikia keisti sunkesnėmis arba galvoti, kaip taikyti judriąsias praktikas didesniai darbuotojų skaičiui.
- Netinka projektams su didelėmis rizikomis.
- Programų sistemų kūrėjai gali ignoruoti verslo sistemą, kurioje programų sistemų gamyba yra tik viena iš veiklų.

Dėl šių priežasčių visiškai atsisakyti judriųjų metodikų neverta, tačiau jas reikia lanksčiau taikyti, gal būt kažko atsisakyti, galbūt kažką pridėti iš tradicinių metodų.

Judriąsias metodikas taikanti įmonė gali neišnaudoti visų judriųjų metodikų teikiamų privalumų dėl nesugebėjimo organizacijoje įtvirtinti šių metodikų ir jomis grįstų procesų. Formalus vertinimo ir gerinimo procesas gali padėti įtvirtinti šiuos procesus ir metodikas [KJ2008].

Nors judriosios metodikos, tinkamai jas taikant, įgalina pasiekti didelį produktyvumą, tačiau nenurodo, kaip būtų galima pasiekti dar didesnio. Formalus vertinimas padeda ieškoti „siaurų“ proceso vietų (angl. *bottle necks*) ir pagelbėti juos šalinant.

Formalaus vertinimo metu nustatoma, kuriems procesams turi būti skiriamas didžiausias dėmesys, tolimesnis jų gerinimas gali būti vykdomas remiantis praktiniais metodais (XP, Scrum, Crystal) ar naudojantis karkasais (RUP).

Darbo tikslas ir struktūra

Darbo tikslas yra pateikti rekomendacijas labai mažų įmonių programų kūrimo proceso vertinimui ir gerinimui.

Pirmame skyriuje trumpai apžvelgiami vertinimo standartai ir modeliai, supažindinama su ISO 15504 standartu ir juo paremtomis metodikomis, skirtomis mažoms įmonėms. Kadangi labai mažoms įmonėms metodikų nėra, nagrinėjamos mažoms įmonėms skirti metodai, siekiant nustatyti, kaip jei gali būti pritaikyti labai mažų įmonių vertinimui.

Antrame skyriuje apžvelgiamos judriosios praktikos, kurias galima panaudoti labai mažos įmonės procesų gerinimui.

Trečiame skyriuje išanalizuojamas mažoms įmonėms skirtų metodų pritaikymas labai mažoms įmonėms, pasirenkamas vienas metodas detalesnei analizei. Pateikiami pasirinkto metodo pakeitimai, pritaikant labai mažos įmonės vertinimui.

Ketvirtame skyriuje pristatytas modifikuoto metodo taikymas labai mažoje Lietuvos įmonėje ir vertinimo patirtis.

Penktame skyriuje apibendrinti metodo pakeitimai ir vertinimo patirtis.

Šeštame skyriuje pristatytas įmonės gerinimo planas ir parodyta, kad įgyvendinus planą, remiantis judriosiomis praktikomis, gerinami procesai pasieks reikalingą lygį. Aprašyta įmonės gerinimo proceso pradžia.

1. Programų kūrimo proceso vertinimas

Šiame skyriuje apžvelgiama trumpa ISO 15504 standarto ir kitų vertinimo metodų ir modelių vystymosi istorija, siekiant nurodyti kontekstą, kuriame toliau nagrinėjamas ISO 15504 standartas.

Per paskutinius tris dešimtmečius atsirado nemažai produktų ir procesų kokybės standartų [Loo07a]. Yra nemažai organizacijų, kuriančių tokius standartus: tarptautinė standartizacijos organizacija (ISO), Tarptautinė elektrotechnikos komisija (IEC), Elektrotechnikos ir elektronikos inžinierių institutas (IEEE), Nacionalinis standartų ir technologijos institutas (NIST). Taip pat daug organizacijų įtraukiama kuriant standartus: Jungtinės karalystės gynybos ministerija (MoD), Programinės įrangos inžinerijos institutas (SEI).

Yra nemažai standartų, kurie skirti procesų kokybės aspektams. Viena iš žinomiausių yra ISO 9000 standartų grupė. ISO 9000 pateikia tarptautinius standartus, apimančius organizacijos kokybės pasiekimus, ypatingai kokybės pasiekimus iš kliento perspektyvos. Tinkamai taikant ISO 9001 standartą, jis padeda organizacijai efektyviau gaminti kokybiškus produktus ir pateikti kokybiškas paslaugas [Loo07a].

1992 metais Jungtinės karalystės gynybos ministerija finansavo tyrimą, tuo metu vadintą „Improve-IT“. Britų standartų institutas (BSI) pasiūlė, kad programinės įrangos procesų vertinimas gali būti standartizuotas. 1992-1993 metais buvo sudaryta ISO/IEC JTC1/SC7 WG10 darbo grupė, kurios tikslas buvo sudaryti reikalavimus programų kūrimo proceso vertinimo tarptautiniam standartui. Programų kūrimo proceso vertinimo standartas turėjo pateikti proceso gebėjimo nustatymą, padėti gerinti procesą, būti lankstus, patikimas ir pakankamai paprastas, kad jį būtų galima suprasti ir taikyti. Šie reikalavimai tapo SPICE projekto pagrindu. 1995 metais buvo atlikti pirmieji „SPICE Process Assessment Model“ bandymai. Modelis buvo toliau tobulinamas ir 1996-1997 buvo atlikti papildomi bandymai ir jais remiantis buvo sudaryta galutinė standarto juodraščio versija.

Nuo 1997 iki 2003 metų ISO/IEC JTC1/SC7 WG10 darbo grupė vykdė standartizavimo procesą, kurio tikslas buvo atsižvelgti į kitus standartus ir transformuoti juodraščių į technines ataskaitas, kurios gavo pavadinimą ISO/IEC 15504. Penkių dalių standartas buvo baigtas 2005 metais, 6 dalis buvo patvirtinta 2007 metais.

Kadangi ISO 15504 buvo suderintas su kitais standartais (ISO/IEC 12207, ISO/IEC 15288, ISO 9000), jis tapo bendru proceso vertinimo standartu, o ne tik programinės įrangos proceso vertinimo standartu. Standartas nebėra grindžiamas vienu procesų modeliu, tačiau pateikia reikalavimus tokio modelio sudarymui [ISO02].

Programų kūrimo proceso vertinimui dar naudojami CMM, CMMI vertinimo modeliai.

JAV gynybos ministerija finansavo Programinės įrangos inžinerijos institutą (SEI) Carnegie Mellon universitete proceso vertinimo standarto sukūrimui. Standartas buvo skirtas įvertinti JAV gynybos ministerijos tiekėjus. Pradinė standarto versija buvo pateikta 1987 metais, vėliau tapo pagrindu „Programinės įrangos gebėjimų brandos modelio“ (SW-CMM) sukūrimui. SEI išleido SW-CMM v1.0 1993 metais. 1994 metais buvo pristatytas dokumentas: „A Software Process Framework for the SEI Capability Maturity Model“ kuris dar vadinamas „CMM vadovu“. 1998 metais buvo pateiktas SW-CMM v2.0. Tuo metu buvo sukurta nemažai modelių, paremtų CMM-SW. Tais pačiais metais buvo sukurtas CMMI-SW, kurio tikslas – integruoti CMM modelius į bendrą modelio karkasą. Kitaip sakant, buvo atsisakyta daugybės modelių ir pateiktas vienas. Šiuo metu naujausia CMMI versija yra CMMI-DEV 1.2.

Bell Canada ir Northern Telecom kompanijos finansavo Trilium standarto sukūrimą. Pirmoji jo versija buvo išleista 1991 metais. 1994 metais buvo išleista paskutinė versija ir šiuo metu standartas nebevystomas. Trilium buvo grindžiamas CMM 1.1 ir buvo skirtas vertinti minėtų kompanijų tiekėjus.

Europos Sąjunga finansavo Bootstrap metodikos sukūrimą. Bootstrap metodika išleista 1993 metais ir vystoma iki šiol. Pagrindiniai Bootstrap metodikos elementai: vertinimo procesas, vertinimo modelis, klausimynai, reitingų ir rezultatų pateikimas, proceso gerinimo gairės, Bootstrap duomenų bazė. Bootstrap metodika suderinta su ISO 15504 standartu.

1.1. ISO/IEC 15504 standartas

ISO 15504 yra tarptautinis procesų vertinimo standartas. Standartas gali būti naudojamas:

- Įvertinti tiekėjų gebėjimą.
- Padėti tiekėjams pagerinti procesų valdymą.
- Įvertinti ir pagerinti savo procesus [Loo04a].

ISO 15504-2:2003 dalyje pateikiami reikalavimai procesų vertinimui. Procesų vertinimui turi būti pateikti šie komponentai:

- Gebėjimo matavimo karkasas.
- Procesų modelis.
- Vertinimo modelis
- Vertinimo metodas [ISO02].

Gebėjimo matavimo karkase aprašyti 6 gebėjimo lygiai ir 9 su jais susieti atributai. Nulinis gebėjimo lygis apibrėžiamas kaip nepilnas, tai reiškia, kad nėra pasiekiamas pirmo gebėjimo lygio atributas. Kiekviename lygyje yra aiškus pasiekimo kriterijus. Pirmame gebėjimo lygyje

pasiekimo kriterijus yra susietas su proceso vykdymu. 2-5 lygių pasiekimo kriterijus priklauso nuo to, kaip gerai procesai yra valdomi, apibrėžiami, kiekybiškai valdomi ir gerinami.

Gebėjimo lygiai ir su jais susieti atributai:

1. Nepilnas.
2. Vykdomas:
 - PA 1.1 Proceso atlikimas.
3. Valdomas:
 - PA 2.1 Proceso vykdymo valdymas.
 - PA 2.2 Darbo produktų valdymas.
4. Apibrėžtas:
 - PA 3.1 Proceso apibrėžimas.
 - PA 3.2 Proceso įdiegimas.
5. Prognozuojamas:
 - PA 4.1 Proceso matavimas.
 - PA 4.2 Proceso kontrolė.
6. Optimizuojamas:
 - PA 5.1 Proceso inovatyvumas.
 - PA 5.2 Proceso optimizavimas.

Procesų modelis turi atitikti šiuos reikalavimus:

- Turėti aprašymą.
- Aprašyti suinteresuotą bendriją ir kaip gauti jos pripažinimą.
- Turėti unikalius procesų identifikatorius ir aprašymus (tikslas ir rezultatai).

Vertinimo modelis sudaromas remiantis procesų modeliu. Vertinimo modelis pateikia dviejų dimensijų procesų gebėjimo požiūrį. Vienoje dimensijoje yra procesų identifikatoriai, o kitoje tų procesų gebėjimo lygiai.

Vertinimo modelis atitinka standartą, jei patenkina šiuos reikalavimus:

- Turi aprašyti apimtį (kokie procesai vertinami ir iki kokio lygio).
- Turi aprašyti indikatorius, padedančius nustatyti procesų atributų pasiekimą.
- Pateikti vertinimo modelio ir proceso modelio susiejimą.

ISO 15504 standarto reikalavimai vertinimo modeliui turi kelis esminius privalumus:

- Galimybė naudoti įvairius procesų modelius.
- Tam pačiam procesų modeliui taikyti kelis vertinimo modelius [ISO02].
- Šalia gebėjimų ir procesų dimensijos pridėti papildomą dimensiją, pavyzdžiui,

produkto kokybės arba rizikos [Loo04a].

ISO 15504 standarto 5 dalis pateikia informacinių technologijų procesų vertinimo modelį, atitinkantį šiuos reikalavimus.

ISO 15504 standartas pateikia reikalavimus vertinimo procesui:

- Naudoti dokumentuotą vertinimo procesą, kuris atitiktų vertinimo tikslą.
- Dokumentuotas vertinimo procesas privalo turėti planavimą, duomenų rinkimą, duomenų validavimą, procesų atributų įvertinimą, duomenų pateikimą.
- Aprašyti vertintojo ir rėmėjo roles bei jų pareigas.
- Nurodyti vertinimo į eigą.
- Vykdyti vertinimą remiantis dokumentuotu vertinimo procesu.
- Aprašyti vertinimo rezultatus remiantis standarto reikalavimais.

1.2. ISO 15504 grindžiami vertinimo metodai mažoms įmonėms

Kadangi vertinimo metodų, skirtų labai mažoms įmonėms ir paremtų ISO 15504 standartu, nėra, toliau apžvelgiami mažoms įmonėms skirti metodai. 3 skyriuje detaliau nagrinėjamas šių metodų pritaikymas labai mažų įmonių vertinimui.

Mažų įmonių proceso vertinimo metodai yra keli: MARES, TOPS, RAPID, QuickScan.

1.2.1. MARES metodas

MARES metodas sukurtas Brazilijoje, Vale do Itajaí universitete, ir skirtas mažų įmonių programų kūrimo proceso vertinimui.

MARES metodas sudarytas iš šių dalių:

- Procesų vertinimo modelis. MARES vertina 21 procesą, kurių dažnai pakanka įvertinti mažą įmonę. MARES procesų modelis paremtas ISO 15504-5 dalimi, tik dalis procesų pašalinta, pavyzdžiui, nėra visos įsigijimo procesų grupės, tačiau šie procesai gali būti nesunkiai sugražinti į procesų modelį ir atliktas jų vertinimas.
- MARES vertinimo procesas. MARES vertinimas yra grindžiamas procesu, nurodytu ISO/IEC 15504 standarto 5 dalyje, išskyrus kontekstualizacijos veiklą [WAS04]. Ši veikla nenumatyta standarte ir naudojama, siekiant apibrėžti vertinimo apimtį. MARES vertinimo procesas atitinka standarto reikalavimus.
- Vertinimo akreditacijos metodas, kuris taikomas siekiant nustatyti, ar vertintojas tinkamas atlikti vertinimą.
- Vertinimo stebėjimo metodas, kuris naudojamas stebėti vertinimą ir užtikrinti nenutrūkstamą metodo gerinimą.

MARES vertinimo schema atrodo taip:

- Atliekama kontekstualizacija, kurios metu nusprendžiama, kokie procesai bus vertinami ir iki kokio gebėjimo lygio.
- Vykdomas pats vertinimas, kurio metu vykdomi šie žingsniai:
 - Pradžioje vykdomi neformalūs interviu su darbuotojais, kurių metu darbuotojai laisva forma pasakoja apie įmonėje vykdomus procesus. Kažkokio konkretaus klausimyno nėra. Taip pat gali būti analizuojami dokumentai. Vertintojas turi susirašyti rastus požymius.
 - Rasti požymiai susiejami su atributų indikatoriais: proceso rezultatais, bazinėmis praktikomis, bendrosiomis praktikomis, proceso įeigos produktais, proceso išeigos produktais.
 - Šis susiejimas turėtų būti peržiūrėtas įmonės atstovų, kurie turėtų patikrinti, ar surinkti ir suklasifikuoti požymiai atitinka įmonės procesus.
 - Pagal suklasifikuotus požymius, nustatomi procesų atributų pasiekimo lygiai iš skalės NPLF, aprašytos ISO 15504 standarte:
 1. Nepasiektas (0-15%). Yra mažai arba iš vis nėra įrodymų, kad proceso atributas buvo pasiektas.
 2. Pasiektas dalinai (16-50%). Yra įrodymų, kad atributas yra pasiektas, tačiau pasiekimas gali būti neprognozuojamas.
 3. Didžiąja dalimi pasiektas (51-85%). Yra žymūs atributo pasiekimo įrodymai, tačiau atributas nėra pilnai pasiektas.
 4. Pasiektas pilnai (86-100%). Yra pilni ir sistemingi atributo pasiekimo įrodymai.
- Šis nustatymas nėra griežtai formalizuotas ir remiasi daugiau vertintojo sprendimu.
- Pagal procesų atributus nustatomi procesų gebėjimo lygiai.
- Gautus procesų gebėjimo lygius lyginant su tiksliniais gebėjimo lygiais, gautais kontekstualizacijos procese, nustatomos procesų stipriosios ir silpnosios pusės.
- Naudojantis proceso-rizikos modeliu nurodomos galimos procesų rizikos ir gerinami procesai. Šis modelis susieja procesų atributų indikatorių silpnąsias vietas su galimomis rizikomis [WAS05], pavyzdžiui, „ENG.6.BP4 verifikuoti programinės įrangos programinius vienetus“ bazinės praktikos

nebuvimas gali sukelti žemo patikimumo ir padidėjusių testavimo pastangų riziką.

1.2.2. RAPID metodas

RAPID (The Rapid Assessment for Process Improvement for software Development) buvo sukurtas tam, kad mažos organizacijos galėtų gauti prasmingą ir patikimą gebėjimo įvertinimą skiriančiam labai ribotą resursų ir laiko kiekį [RTCH00]. RAPID yra paremtas ISO TR 15504 standarto versija ir neatitinka dabartinio ISO 15504 standarto. RAPID metodas orientuotas į proceso gerinimą, bet ne į subrangovo vertinimą.

RAPID yra grindžiamas šiais principais:

1. Vertinimas padaromas per vieną dieną, siekiant kuo labiau sumažinti vertinimo laiką ir kainą.
2. Vertinimas grindžiamas aštuonių procesų vertinimu.
3. Labai svarbi vertintojų patirtis ir kompetencija. Reikalingi du vertintojai, turintys vertinimo pagal ISO 15504-5 patirties.
4. Duomenų rinkimas grindžiamas diskusija, kurios dalyviai yra procesų vykdytojai, vadybos personalas.
5. Procesų gebėjimas nustatomas remiantis diskusijos dalyvių nuomone, o ne vertintojų sprendimu.

RAPID vertinimo modelis grindžiamas ISO 15504 standarto 5 dalimi, tačiau imami tik 8 procesai. Vertinimo modelis atitinka reikalavimus, pateiktus ISO 15504 standarto 2 dalyje. Vertinami šie procesai:

1. Reikalavimų surinkimas (CUS.3).
2. Programų sistemos gamyba (ENG.1).
3. Projektų valdymas (MAN.2).
4. Konfigūracijos valdymas (SUP.2).
5. Kokybės užtikrinimas (SUP.3).
6. Problemų sprendimas (SUP.8).
7. Rizikos valdymas (MAN.4).
8. Procesų įtvirtinimas (ORG.2.1).

Projektų valdymo, konfigūracijos valdymo, kokybės užtikrinimo, problemų sprendimo procesai įtraukti į šį metodą, todėl, kad yra susieti su kitų procesų antro gebėjimo lygio pasiekimu, o rizikos valdymo ir procesų įtvirtinimo procesai susieti su trečio gebėjimo lygio pasiekimu.

RAPID metode „programų sistemos gamyba“ laikomas procesas, kuris apjungia reikalavimų išaiškinimo, reikalavimų surinkimo ir programų sistemos gamybos procesus ISO 15504 standarto vertinimo modelyje.

Gebėjimo dimensija yra identiška pateiktajai ISO 15504 standarto 2 dalyje, tačiau vertinimas dažniausiai vykdomas iki 3 lygio.

RAPID pateikia klausimyną, kuris naudojamas diskusijos metu ir vertintojas padeda diskusijos dalyviams įvertinti procesų gebėjimą, todėl šiam metodui labai svarbi vertintojo kompetencija. RAPID pateikia išsamias procedūras, kaip reiktų vykdyti organizacijos vertinimą. Organizacijos vertinimas vykdomas šiais žingsniais:

1. Vykdoma diskusija, kurioje dalyvauja organizacijos žmonės, atsakingi už programinės įrangos procesų valdymą.
2. Pagal diskusijos rezultatus nustatoma, kiek yra pasiekti gebėjimo lygių atributai.
3. Nustatomi organizacijos procesų gebėjimo lygiai.
4. Sudaroma vertinimo ataskaita, kuri apima: stipriąsias puses, silpnąsias puses, atributų įvertinimus, gebėjimo lygius, gerinimo rekomendacijas.
5. Po 7-16 mėnesių daromas susitikimas, siekiant įvertinti, kiek pasiteisino gerinimo sprendimai. Prieš susitikimą gali būti atliktas pakartotinis vertinimas.

1.2.3. QuickScan metodas

QuickScan yra komercinis mažų įmonių vertinimo metodas, sukurtas „LC Consulting“. Šiuo metodu siekiama gauti kuo didesnę proceso vertinimo naudą su kuo mažesnėmis vertinimo pastangomis [Loo04b]. QuickScan vertinimas pilnai neatitinka ISO 15504 standarto, kadangi nevykdomas suderinamumo su standartu tikrinimas. Kitos metodo veiklos gali būti interpretuojamos, kaip atitinkančios standartą, pavyzdžiui, nėra vykdomas duomenų validavimas, tačiau interviu metu kompetetingas vertintojas turi užtikrinti, kad surinktų duomenų jam pakaks atributų pasiekimui įvertinti, taip pat vertintojas užtikrinti, kad surinkti duomenys nebus prieštaringi. Kadangi vertintojas atlieka duomenų validavimą interviu metu, tai yra viena iš priežasčių, kodėl vertinimui labai svarbi vertintojo kompetencija.

QuickScan metodo tikslai:

- Suteikti informacijos ISO 15504 standarto terminais, apie tai, kokie procesai vykdomi įmonėje.
- Nustatyti procesų vykdymo lygį (procesų gebėjimą).
- Remiantis trūkstamomis bazinėmis praktikomis, suteikti informacijos, kaip procesai gali būti pagerinti.

Vertinimas vykdomas interviu forma, jokios dokumentų peržiūros nėra vykdomos, todėl vertinimui svarbu gerai išrinkti darbuotojus, dalyvausiančius vertinimo interviu. Vertinimui reikalingi patyrę vertintojai, galintys greitai susieti darbuotojų pasakojimą su bazinėmis praktikomis.

QuickScan vertinimo procesas vykdomas taip:

- Vertinimo rėmėjas užpildo klausimyną, kuriame nurodomi asmenys, dalyvausiantys interviu. Nurodoma, kaip gerai jie galės įvertinti įmonėje vykstančius procesus.
- Vertintojai paruošia vertinimo klausimyną pagal tai, kokie procesai pasirenkami vertinimui. Klausimai pateikti vertinimo metode.
- Interviu metu, įmonės darbuotojai papasakoja, kaip dirba, vertintojas gali pateikti klausimus pagal interviu klausimyną. Siekiama, kad darbuotojai pasakotų apie įmonės veiklą savais terminais. Darbuotojų pasakojimai susiejami su interviu klausimais.
- Vertintojo asistentas pažymi, kaip procesai yra vykdomi, darbuotojų pasakojimai susiejami su vertinimo modelio praktikomis ir resursais.
- Vertinimo tikslumas labai priklauso nuo to, kokio tikslumo informacija buvo pateikta vertintojams interviu metu, todėl reikalingi patyrę vertintojai.
- Baigus interviu, klausimyno kopija paliekama vertinimo rėmėjui.
- Vertintojai turi kelias dienas sudaryti vertinimo ataskaitai ir nurodyti gerinimo galimybes.

1.2.4. TOPS metodas

TOPS (Toward Organised Process in SMEs) metodas yra ESPIRIT/ESPINODE iniciatyvos (proceso gerinimo iniciatyva, remiama Europos Sąjungos) centrinėje Italijoje dalis [ESP99]. Šis metodas paremtas ISO 15504 standartu ir orientuotas į įmonės gerinimą.

Vertinimas vykdomas remiantis klausimynu. Įvertinami 3 procesai:

- Programų sistemų reikalavimų sudarymas (ENG.2);
- Programų sistemos integracija ir testavimas (ENG.5);
- Peržiūros ir auditai (CUS.4).

TOPS vertinimas vykdomas taip:

1. Telefonu apklausiamas įmonės darbuotojas, siekiant išsiaiškinti esminius įmonės bruožus.
2. Vykdomas interviu su įmonės darbuotojais remiantis klausimynu, padedančiu įvertinti minėtus 3 procesus.

3. Atsakymai suvedami į Excell lentelę, kuri suskaičiuoja gebėjimo lygius.

Šis metodas yra labai paprastas, jei galima apsiriboti tik tais trimis procesais. Tačiau, turint tik tris procesus, yra nemaža tikimybė, kad jie nebus svarbiausi įmonės tikslams pasiekti, o norit plėsti metodą papildomais procesais reikėtų nemažai pastangų: reikėtų sugalvoti klausimus, įrodyti, kad teigiamas atsakymas padeda siekti proceso tikslų, nustatyti, kiek reikia teigiamų atsakymų, kad proceso tikslas būtų pasiektas.

1.2.5. Apibendrinimas

RAPID ir QuickScan metodai panašūs tuo, kad naudoja klausimynus kelių numatytų procesų vertinimui ir nevertina darbo produktų kaip bazinių praktikų įrodymų, taiko mažiau formalų vertinimą, remiantis interviu ir vertintojo sprendimu. Be to, visi metodai atvirai deklaruoja, kad jų vertinimas nepilnai atitinka ISO 15504 standartą. Šių metodų esminės vertinimo proceso dalys yra apklausiamų darbuotojų pasirinkimas, interviu pagal klausimyną ir vertinimo rezultatų pateikimas. Klausimynas šiuose metoduose naudojamas kaip pagalbinių priemonė interviu metu ir aiškiai nenurodo, kaip atsakymas siejamas su proceso atributų įvertinimu, įvertinimą turi atlikti vertintojas. Todėl šie metodai reikalauja aukštos vertintojo kompetencijos.

TOPS metodas taip pat naudoja klausimynus, aiškiai nurodo, kaip klausimai siejami su atributų indikatoriais (naudojamas Excell įrankis), metodas yra labai paprastas, lengvai naudojamas, nereikalauja patyrusio vertintojo, tačiau apima tik 3 procesus ir yra sunkiai plečiamas.

Tuo tarpu MARES vertinimo metodas yra pilnai atitinkantis ISO 15504 standartą, ir dėl to MARES procesas yra sudėtingesnis nei RAPID, TOPS ir QuickScan. MARES vertinimo, procesas apima vertinimo, kontekstualizacijos planavimus, kontekstualizacijos procesą. Pats gebėjimo lygių vertinimas grindžiamas ne klausimynais, o vertinimo modeliu, panašiu į ISO 15504-5 dalį.

RAPID ir TOPS procesų žymėjimas skiriasi nuo dabartinio standarto procesų žymėjimo, kadangi jie buvo kurti 1999 metais, tuo metu, kai standartas dar nebuvo patvirtintas. Pagrindinis RAPID ir TOPS metodų trūkumas yra tas, kad jie apima mažai procesų. Įmonės vertinami/gerinami procesai labai priklauso nuo įmonės veiklos pobūdžio ir veiklos tikslų, todėl vertinant kelis procesus yra didelė tikimybė, kad tikrai reikalingi procesai nebus įvertinti arba bus vertinami ne kritiškiausi procesai. Ši problema ryškiausia su TOPS metodu, kuris vertina tik 3 inžinerinius procesus. Su RAPID problema kiek mažesnė, kadangi jis apima daugiau procesų.

2. Judriosios PĮ kūrimo metodikos

Po įmonės vertinimo atliekamas įmonės procesų gerinimas. Gerinant įmonės procesus galima naudotis judriosiomis metodikomis. Šios metodikos skirtos mažoms programinės įrangos kūrimo komandoms, yra nesunkiai įgyvendinamos, remiasi bendravimu, o ne formaliu procesų įgyvendinimu. Tačiau tai nereiškia, kad formalių metodų naudojimas mažose komandose yra neįmanomas.

Formalių vertinimo ir gerinimo metodų taikymas padeda geriau išnaudoti judriąsias metodikas. Taikant CMMI, Scrum komandų produktyvumas yra dvigubai didesnis nei tradicinių komandų. Judriųjų metodikų taikymas keliuose projektuose sumažino defektų skaičių 38-40% vykdant galutinį testavimą [KJ2008]. Naudojant tik judriąsias metodikas šios statistikos nebūtų galima gauti, kadangi judriosios metodikos nereikalauja turėti matavimų.

Judriųjų metodikų „judėjimas“ prasidėjo po „Judrumo manifesto“ (angl. *Agile Manifesto*) paskelbimo 2001 metais. Manifestas buvo parašytas ir pasirašytas septyniolikos „lengvųjų metodininkų“. „Judrumo manifesto“ autoriai vertina:

- Žmones ir bendravimą labiau negu procesus ir įrankius.
- Veikiančią programinę įrangą labiau negu suprantamą dokumentaciją.
- Darbą su klientu labiau negu derybas dėl kontrakto.
- Reakciją į besikeičiantį planą nei plano vykdymą.

Remiantis šiais principais yra sukurta nemažai judriųjų metodikų (nors judriosios metodikos egzistavo gerokai prieš pasirašant „Judrumo manifestą“). Kelios žinomiausios: Scrum ir XP.

2.1. Scrum

Pirmą kartą pramonėje „Scrum“ terminas buvo panaudotas apibūdinti hiperproduktyvią gamybą. Šį terminą sugalvojo Ikujiro Nonaka ir Hirokata Takeushi 1987 metais.

Scrum metodika yra orientuota į valdymo praktikas, mažiau į inžinerines praktikas. Patys Scrum kūrėjai jį apibūdina kaip „orientuotą į paprastumą valdymo ir kontrolės procesą, kuris padeda sukurti tokią programinę įrangą, kuri atitinka verslo poreikius“ [SB02].

Pradedant Scrum projektą sudaromas naujos sistemos funkcionalumo, galimybių ir technologijų sąrašas. Šis sąrašas vadinamas produkto neužbaigtų darbų sąrašu (angl. *Product Backlog*). Darbų sąrašas yra prioritetizuotas produkto reikalavimų sąrašas. Jis nėra fiksuotas, o keičiasi ir evoliucionuoja produkto kūrimo metu. Darbai į produkto sąrašą gali patekti iš įvairių šaltinių: vartotojų, klientų, pardavimo skyriaus, marketingo skyriaus, klientų aptarnavimo

skyriaus. Tačiau tik produkto savininkas (angl. *Product Owner*) gali keisti prioritetus.

Mažos daugiafunkcinės komandos vykdo produkto kūrimą (angl. *Scrum Teams*). Šios komandos apsiima tiek funkcionalumo iš darbų sąrašo, kiek gali sukurti per 30 dienų. Šis 30 laikotarpis vadinamas sprintu (angl. *Sprint*). Visi sprintai baigiami pristatant veikiančią produkto funkcionalumą. Prieš pradėdant sprintą yra sudaromas sprinto darbų sąrašas (angl. *Sprint Backlog*). Produkto architektūra yra vystoma kiekvieno sprinto metu, o ne sukuriama pilna architektūra pirmųjų sprintų metu. Scrum leidžia kelioms komandoms dirbti prie to pačio produkto.

Sprinto metu vienas žmogus prižiūri Scrum praktikų laikimąsi, padeda komandai priimti sprendimus ir stengiasi užtikrinti, kad komanda turėtų visus reikalingus resursus. Tas žmogus vadinamas Scrum meistru (angl. *Scrum Master*). Taip pat, jis turi užtikrinti, kad niekas komandos netrukdytų ir nepriskirtų naujų užduočių tuo metu, kai vykdomas Scrum sprintas.

Scrum komanda susitinka kiekvieną dieną darbų būklės aptarimui. Tokie susitikimai vadinami kasdieniais Scrum susitikimais (angl. *Daily Scrum*). Susitikimo metu išaiškinamas komandos progresas, nustatomos kliūtys, trukdančios toliau dirbti.

Kai baigiamas sprintas, visa kūrimo komanda ir vadovybė surenkama sprinto aptarimui (angl. *Scrum Review Meeting*). Susitikimo tikslas – aptarti komandos sukurtą funkcionalumą. Sprendimas gali būti dvejopas: naudoti tai, kas buvo sukurta arba tiesiog „išmesti“ sprinto pakeitimus.

Po šio susitikimo, galima pakeisti produkto užduočių sąrašo prioritetus. Taip pat galima nuspręsti išleisti produktą, net jei išleidimas buvo planuojamas vėliau. Išleidimo sprendimas priklauso nuo empiriškai valdomos kainos, funkcionalumo ir kokybės.

Teigiama, kad Scrum, išmesdamas sudėtingas ir nereikalingas valdymo praktikas, palieka daugiau laiko darbui. Scrum procesas gali atrodyti labai paprastas, tačiau Scrum pateikia visas būtinas valdymo praktikas, kurios reikalingos sukurti kokybiškus produktus [SB02].

2.2. Ekstremalus programavimas (XP)

Skirtingai negu Scrum, XP metodika orientuota į inžinerines praktikas. „Metodika pabrėžia komandinį darbą, greitą ir ankstyvą programinės įrangos kūrimą, sumanaus kūrimo praktikas“ [Lar03]. Metodikoje naudojama 12 praktikų:

- Planavimo žaidimas.
- Maži ir dažni leidimai.
- Sistemos metaforos.
- Paprasta architektūra.

- Testavimas.
- Dažni pertvarkymai.
- Programavimas poromis.
- Produktas priklauso visai komandai.
- Pastovus integravimas.
- Tempo palaikymas.
- Visa komanda kartu.
- Kodavimo standartai.

Išleidimo planavimo žaidimo tikslas – nustatyti kito leidimo, suteikiančio didžiausią vertę klientui, apimtį. Dažniausiai tai yra pusdienio arba dienos susitikimų sesija, kurios metu klientas užrašo savo „istorijas“ ant kortelių. Šios „istorijos“ yra funkcionalumas, kurį programuotojams teks sukurti. Programuotojai tą funkcionalumą įvertina. „Istorijų kortelės“ gali būti likusios ir nuo praeito planavimo žaidimo. Toliau klientas pasirenka, koks bus sekantis leidimas.

Iteracijų planavimo žaidimo tikslas – nustatyti sekančios iteracijos „istorijas“, suplanuoti iteracijos užduotis. Tai daroma prieš kiekvieną 1-3 savaičių ilgumo iteraciją. Klientas parenka „istorijų korteles“ sekančiai iteracijai. Kiekvienai „istorijai“ programuotojai sudaro užduočių sąrašą, toliau kiekvienas programuotojas pasirenka užduotis, kurias darys, jis gali planuoti laiką, skiriamą kiekvienos užduoties įvykdymui. Jei užduotis įvertinama daugiau, kaip dviem dienom, ji išskaidoma.

Maži ir dažni leidimai yra evoliucinis programinės įrangos kūrimas. Tokie leidimai tinka ne visiems projektams.

Sistemos metaforos palengvina sistemos architektūros supratimą. Parenkama paprasta, klientui suprantama metafora ir ja naudojantis apibūdinama sistemos architektūra.

Sudarant paprastą architektūrą nereikia numatyti būsimų architektūros pakeitimų, vengiama bendrų komponentų, kurie iš karto nėra reikalingi, vengiama pasikartojančių išeitinių tekstų, stengiamasi išlaikyti minimalų klasių ir metodų skaičių.

XP pabrėžia testavimo svarbą. Visos funkcijos privalo turėti automatinius priėmimo testus. Visi testai baigiami sėkmingai arba ne, todėl nėra reikalinga žmogaus peržiūra. Priėmimo testai rašomi bendradarbiaujant su klientu.

Dažni pertvarkymai užtikrina, kad esant išeitinio kodo pakeitimams, testai baigiami sėkmingai. Taip pat tai yra pastovus architektūros gerinimas.

Visos programos rašomos dviejų programuotojų dirbančių vienu kompiuteriu. Tai yra realaus laiko išeitinių tekstų peržiūros. Poros gali būti keičiamos. Bet kuri programuotojų pora

gali keisti išeitinį tekstą bet kurioje programos vietoje. Tai reiškia, kad išeitinis tekstas priklauso ne vienam ar keliems asmenims, o visai komandai.

Kadangi išeitinis tekstas yra visos komandos nuosavybė, jis dažnai pertvarkomas, programuotojų poros dažnai keičiamos, reikalingas vieningas programavimo standartas.

Gaminamas produktas yra nuolatos integruojamas ir testuojamas.

Pastovaus tempo praktika užtikrina, kad žmonės nedirbtų viršvalandžių.

Visa projekto komanda (programuotojai ir klientai) dirba kartu toje pačioje aplinkoje. Vienas arba keli kliento atstovai dirba kartu su programuotojais, jie turėtų būti dalykinės srities ekspertai ir galėti priimti sprendimus, susijusius su reikalavimais ar jų prioritetais.

2.3. Judrusis darbų vertinimas ir planavimas

Judrusis planavimas nuo tradicinio skiriasi tuo, kad didesnę dėmesį sutelkia į planavimo procesą, o ne į planą [Coh06]. Judriojo planavimo esmė yra ta, kad planas visada sudaromas turint omenyje, kad jis greičiausiai pasikeis vykdant projektą. Plano pakeitimai galimi dėl įvairių priežasčių:

- Funkcionalumo įgyvendinimas užtruko ilgiau arba trumpiau nei tikėtasi.
- Sužinoma daugiau apie projektą.
- Pasikeitė kliento poreikiai.

Kadangi planas gali keistis, reikalingas toks planas, kurį būtų galima lengvai pakeisti. Tai reiškia, kad planavimas tampa svarbesnis už planą. Plano pakeitimas nereiškia, kad keičiamos datos. Bet, jei sužinoma, kad suklysta dėl kažkurio gaminamo produkto aspekto, kažką reikia keisti. Yra būdų pakeisti planą nekeičiant datos:

- Galima išmesti arba supaprastinti dalį funkcionalumo.
- Pridėti papildomų žmonių.

Kadangi projekto pradžioje apie galutinį produktą nežinoma viskas, todėl negalima iki smulkmenų suplanuoti viso projekto. Judrusis planavimas vykdomas viso projekto metu. Nustatomi programinės įrangos išleidimai (angl. *release*), kiekvieno išleidimo pradžioje atliekamas grubus išleidimo planavimas. Kiekviena programinės įrangos funkcija vertinama gamybos dienomis arba funkciniais taškais. Išleidimai suskaidomi į iteracijas, detalus planavimas vykdomas kiekvienos iteracijos pradžioje, iteracijos planavimo metu kiekviena įgyvendinama funkcija išskaidoma į užduotis ir kiekviena užduotis vertinama darbo valandomis.

Taigi judrusis planavimas yra:

- Sukoncentruotas į planavimą, o ne į planą.
- Skatina pokyčius.

- Planai nesunkiai keičiami.
- Vykdomas viso projekto metu.

2.4. Judrusis testavimas

Judriosiose metodikose kokybės užtikrinimui naudojama testais paremta gamyba (angl. *Test-driven development*, TDD). Programuotojas parašo dalį funkcionalumo, jei testas nepavyksta, programuotojas pakeičia kodą taip, kad testas pavyktų [CG09].

Judrusis testavimas remiasi dešimčia principų, kuriais turėtų vadovautis kiekvienas testuotojas:

- Palaikyti pastovų atgalinį ryšį.
- Sukurti vertę klientui.
- Naudoti žodinę komunikaciją.
- Turėti drąsos.
- Viską daryti kuo paprasčiau.
- Naudoti pastovų gerinimą.
- Reaguoti į pokyčius.
- Būti savarankiškam.
- Susikoncentruoti į žmones.
- Mėgautis.

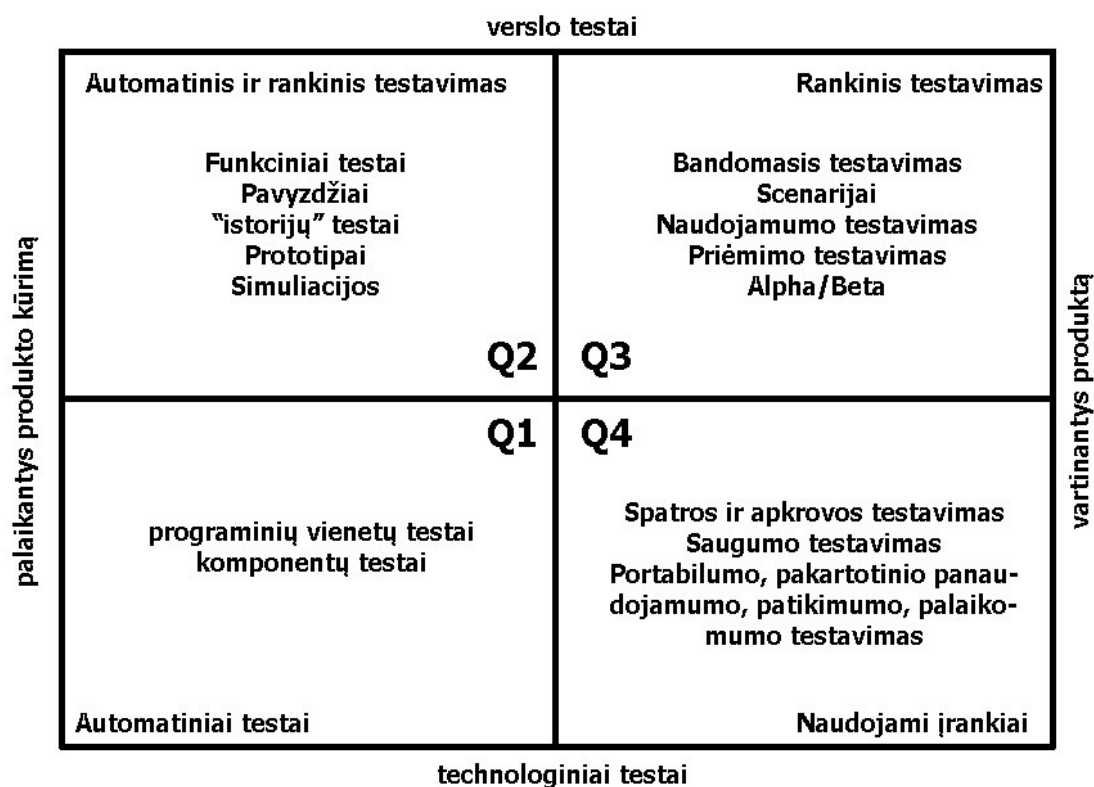
Pagrindiniai skirtumai lyginant su tradiciniu testavimu:

- Tradiciniuose metoduose testavimas yra gamybos pabaigoje, dėl to vėluojant projektui taupoma testavimo sąskaita. Judrusis testavimas vykdomas pabaigus kiekvieną iteraciją.
- Veikiantys testai atspindi reikalavimus, o reikalavimai nėra formuluojami atskirame dokumente. Tradicinių metodų atveju, rašoma reikalavimų specifikacija ir pagal ją sudaromi testai.
- Greičiau gaunami testavimo rezultatai, todėl galima greičiau sureaguoti į problemas, kol jos dar neįsisenėjo.
- Programinės įrangos kokybė yra ne vien tik testuotojų, bet visos komandos problema.

Pagal judriojo testavimo teoriją testai yra suskirstyti į kelias kategorijas pagal tai, kokio tikslo siekiama juos vykdant.

1 pav. pateikta diagrama, kurioje pavaizduotos testų kategorijos, sugrupuotos pagal tai,

kokiam tikslui testai vykdomi.



1 pav. Testų kategorijos

Du kvadrantai kairėje pusėje apima testus, kurie padeda komandai kurti produktą. Testavimo koncepcija, kai testavimas naudojamas padėti programuotojams, yra nauja. Ši koncepcija yra pagrindinis skirtumas, kuris skiria testavimą tradiciniuose metoduose nuo judriojo testavimo [CG09].

2.5. Proceso gerinimas remiantis judriosiomis praktikomis

Extreme programming praktikas įmanoma atvaizduoti į ISO 15504-5 procesus (1 lentelė).

1 lentelė. XP praktikų susiejimas su ISO 15504 procesais [Loo07b]

Extreme Programing praktika	Susietas ISO 15504 procesas
Planavimas „istorijomis“, klientas visada šalia, nenutrūkstama integracija	ENG.1 Reikalavimų išsiaiškinimas ENG.2 Reikalavimų valdymas
Planavimo žaidimas, istorijos, nedideli	MAN.3 Projektų valdymas

išleidimai, projekto greitis	
Programavimas poromis	SUP.4 Peržiūros
Kolektyvinė išeities tekstų nuosavybė, maži išleidimai ir nenutrūkstama integracija	SUP.8 Konfigūracijos valdymas
Metafora, paprastas projektavimas, pertvarkymai (angl. <i>refactoring</i>), programavimas poromis	ENG.5 Projektavimas
Testavimas (modulių ir funkcinis)	ENG.8 Testavimas
Nenutrūkstama integracija	ENG.7 Integravimas ir ENG.8 testavimas
Programavimo standartas	ENG.6 Programinės įrangos gamyba

Kaip matome iš 1 lentelės, daugumą mažoms įmonėms aktualiausių procesų (žr. toliau) galime susieti su XP praktikomis. Tačiau XP praktikų vertinimas labai priklauso nuo vertintojo sprendimo, dėl to gali atsirasti skirtingos praktikų interpretacijos [Loo07b]. Be to, judriųjų metodikų bazinės praktikos skiriasi nuo tradicinių bazinių praktikų, kurios naudojamos ISO 15504-5 dalyje. Taigi, gali tekti vertinti procesus ne pagal bazines praktikas, o pagal tikslus, kadangi pirmas gebėjimo lygis gali būti pasiektas bet kokiais įrankiais, metodais ar praktikomis [LF09].

Dažniausiai vertinami procesai programinę įrangą kuriančiose organizacijose yra [Loo07b]:

- ENG.1 Reikalavimų išaiškinimas.
- ENG.5 Programinės įrangos projektavimas.
- ENG.6 Programinės įrangos kūrimas.
- ENG.8 Programinės įrangos testavimas.
- MAN.3 Projektų valdymas.
- SUP.9 Problemų sprendimas.
- SUP.8 Konfigūracijos valdymas.

Šis procesų sąrašas buvo išskirtas naudojant QuickScan vertinimo metodą. Kadangi šis metodas skirtas procesų vertinimui siekiant sužinoti, kuriuos procesus reiktų gerinti, galima teigti, kad šių procesų gerinimas yra aktualiausias mažoms įmonėms.

Toliau aptariama, kaip mažoms įmonėms problematiškiausius procesus galima susieti su judriosiomis praktikomis.

2.5.1. Reikalavimų išaiškinimas (ENG.1)

Pagal ISO 15504-5 dalį šio proceso rezultatai turėtų būti:

1. Sukurta pastovi komunikacija su klientu.
2. Reikalavimai apibrėžti ir sukurtas reikalavimų bazinis komplektas.
3. Sukurtas mechanizmas, padedantis prisitaikyti prie besikeičiančių poreikių ir reikalavimų.
4. Sukurtas mechanizmas, padedantis stebėti kliento poreikius.
5. Klientai gali nesunkiai nustatyti savo reikalavimų įgyvendinimo būseną.
6. Nustatyti pagerinimai, įtakoti besikeičiančių technologijų ir kliento poreikių, valdomas jų poveikis.

Visi šie rezultatai gali būti pasiekiami taikant judrųjį vertinimą ir planavimą, kadangi kiekvienos iteracijos metu gali būti peržvelgiami įgyvendinti, nauji ir pasikeitę kliento reikalavimai. Taip pasiekiami 1, 3, 4, 5, 6 rezultatai. Reikalavimai projekto pradžioje gali būti apibrėžti labai abstrakčiai ir tolimesnis detalizavimas gali būti vykdomas pačio projekto metu, todėl galima sakyti, pasiekiamas ir 2 rezultatas.

2.5.2. Programinės įrangos projektavimas (ENG.5)

ISO 15504 standarte numatytas tikslas „sukurti programinės įrangos projektą, kuris būtų susietas su reikalavimais“ nėra pasiekiamas, kadangi judriosios praktikos mažai kreipia dėmesio į programinės įrangos projektavimą. XP praktika „metaforos“ gali būti susieta su pirmu rezultatu „sudarytas programinės įrangos architektūrinis projektas, atspindintis reikalavimus“, žinoma, kitų rezultatų nepadengia. Aišku, judriosios praktikos niekaip neapriboja šio proceso tikslų pasiekimo.

2.5.3. Programinės įrangos gamyba (ENG.6)

Programinės įrangos kūrimo rezultatai iš dalies pasiekiami taikant XP praktikas. Rezultatai pagal ISO 15504-5 dalį:

1. Sukurtas verifikavimo pagal reikalavimus kriterijus programinės įrangos moduliams.
2. Sukurti programinės įrangos moduliai.
3. Nustatytas darnumas tarp reikalavimų, projekto ir programinės įrangos.
4. Įvykdytas programinės įrangos verifikavimas pagal reikalavimus.

Kadangi XP metodikoje iš pradžių sudaromi testai, atitinkantys reikalavimus, ir pagal juos kuriama programinė įranga, galima sakyti, kad pasiekiami 1, 2, 4 tikslai. 3 tikslas pasiekiamas iš dalies, kadangi XP nenurodo, kaip turi būti užtikrinamas projekto darnumas su reikalavimais ir veikiančia programine įranga.

2.5.4. Programinės įrangos testavimas (ENG.8)

XP testavimo praktikas galima papildyti kitomis judriojo testavimo praktikomis. Judriosios praktikos stipriai orientuotos į testavimą, prieš kūrimą sudaromi veikiančys testai ir vykdomi viso kūrimo metu, todėl pasiekiami proceso rezultatai (užtikrinti, kad įvykdyti reikalavimai, įrašyti testavimo rezultatus, sukurti regresinio testavimo strategiją).

2.5.5. Projektų valdymas (MAN.3)

Scrum projekto valdymo praktikomis galima papildyti XP praktikas, taip pat dar labiau išplėsti judriojo vertinimo ir planavimo metodais. Tokiu atveju, šios judriosios praktikos apima šiuos projektų valdymo proceso rezultatus:

1. Numatyta darbo apimtis.
2. Numatyti reikalingi resursai ir apribojimai.
3. Užduotys yra įvertinamos.
4. Sudarytas įgyvendinimo planas.
5. Stebimas projekto progresas.
6. Daromi korekciniai veiksmai, jei projektas neatitinka plano.

1-3 rezultatai pasiekiami vykdant lankstaus planavimo praktikas, 5-6 rezultatai taip pat pasiekiami, kadangi projektai vykdomi iteracijomis, kiekvienos iteracijos pradžioje apžvelgiamas progresas, nustatomi nukrypimai nuo plano ir į juos atsižvelgiama planuojant kitas iteracijas. 4 rezultato įgyvendinimas nėra numatytas judriosiose praktikose.

2.5.6. Problemų sprendimas (SUP.9)

Problemų sprendimų procesu galima laikyti kasdienius rytinius pasitarimus, kuriuose dalyvių prašoma nurodyti problemas. Nėra tiksliai aprašoma, kaip vykdomas tolimesnis problemų sprendimo valdymas. Taigi judriosios praktikos tik iš dalies padengia šio proceso tikslą: „užtikrinti, kad visos surastos problemos būtų identifikuojamos, išanalizuotos, valdomos ir kontroliuojamos iki išsprendimo“.

2.5.7. Konfigūracijos valdymas (SUP.8)

XP padengia nemažai konfigūracijos valdymo veiklų, kadangi teigia, kad išeitinis tekstas turi būti versijuojamas, būtina naudoti konfigūracijos valdymo įrankį. Taip pat yra pakeitimų valdymas, kadangi klientas nusprendžia, ką keisti. Komanda integruoja pakeitimus į išeitinį tekstą, naudodama modulių testavimą [Kos03].

3. Vertinimo metodų pritaikymas labai mažoms įmonėms

Labai mažai įmonei skirtas metodas turėtų būti paprastas, pigus ir greitai įvykdomas.

Metode neturėtų būti mažai įmonei neaktualių veiklų, vertinimo veiklos turėtų būti specialiai optimizuotos labai mažos įmonės vertinimui. Vertinimo metodas turėtų vertinti tik tuos procesus, kurie reikalingiausi mažai įmonei, todėl reikia metodo, kuriuo remiantis būtų galima nustatyti vertinimo apimtį. Metodo vertinimo modelyje turėtų būti procesai, kurie padengtų vertinimo apimtį, arba vertinimo modelis turėtų būti sudarytas taip, kad jį būtų nesunku plėsti naujais procesais.

Labai mažai įmonei svarbi ir metodo kaina, kadangi labai mažos įmonės yra nedidelės pajamos. Reikėtų, kad pats metodas ir vertinimas kainuotų kuo mažiau. Kadangi išorinio vertintojo paslaugos yra brangios, metodas turėtų leisti vykdyti vidinį vertinimą.

3.1. Vertinimo metodų analizė

Nagrinėjant ISO 15504 modeliu paremtų vertinimo metodų tinkamumą labai mažoms įmonėms išskirti šie svarbūs aspektai:

1. Apimties nustatymas.
2. Dydis ir sudėtingumas.
3. Plečiamumas.
4. Viešas prieinamumas.
5. Vidinio vertinimo galimybė.
6. Metodo naujumas.

Tolimesniuose skyreliuose jie panagrinėti detaliau.

3.1.1. Apimties nustatymas

Vertinant labai mažą įmonę svarbu nustatyti įmonės tikslus ir, jais, nustatyti įmonei svarbiausius procesus. Taip užtikrinama, kad vertinimas bus efektyvus ir duos didžiausią įmanomą naudą įmonei. MARES metode tai daroma kontekstualizacijos procese, tuo tarpu RAPID, TOPS vertina fiksuotą skaičių procesų nepriklausomai nuo įmonės tikslų, todėl šie metodai neužtikrina, kad bus vertinami aktualiausi konkrečios įmonės procesai. Nors MARES vertinimo metode yra kontekstualizacijos procesas, jis nėra iki galo išvystytas, t.y. nepateikia pilno sprendimo, kaip remiantis įmonės veiklos tikslais, nustatyti vertinimo apimtį. QuickScan leidžia vertinti bet kokius ISO 15504-5 vertinimo modelio procesus, tačiau nepateikia jokių gairių, kaip išrenkami vertinami procesai, pateikia tik dažniausiai mažose įmonėse vertinamų procesų sąrašą.

3.1.2. Dydis ir sudėtingumas

Didesni ir sudėtingesni metodai yra sunkiau pritaikomi labai mažų įmonių vertinimui. Iš visų vertinimo metodų MARES yra sudėtingiausias, kadangi jis yra suderinamas su ISO 15504, pateikia detalius vertinimo procesų aprašymus ir vertinimo metu naudojamas formas. Mažesnis RAPID, TOPS ir QuickScan metodų sudėtingumas grindžiamas tuo, kad šie metodai yra paremti ISO 15504 standartu, tačiau nėra pilnai atitinkantys standartą. RAPID ir QuickScan yra paprastesni, kadangi reikalauja kompetetingo vertintojo su vertinimo patirtimi.

3.1.3. Plečiamumas

Svarbu, kad metodo vertinimo modelis būtų lengvai plečiamas, kadangi vertinant labai mažą įmonę gali pasirodyti, kad įmonei aktualūs procesai, susieti su įmonės veiklos tikslais, neįtraukti į metodo vertinimo modelį. TOPS metodas yra sunkiai plečiamas, kadangi paremtas klausimynais, norint pridėti naujus procesus, reikia sudaryti proceso klausimyną ir įrodyti, kad pateikti atsakymai į klausimus parodo, kad pasiekiamas reikalingas proceso gebėjimo lygis. RAPID taip pat turi klausimynus, kurie sunkiai plečiami, reikėtų sugalvoti tokius klausimus, kurie geriausiai apibūdintų vertinamų procesų gebėjimo lygius. QuickScan taip pat sunkiai plečiamas dėl tų pačių priežasčių kaip ir RAPID, tačiau klausimynai apima visus ISO 15504-5 procesus.

MARES vertinimo modelis nėra griežtai susietas su vertinimo metodu, vertinimas vykdomas renkant įrodymus ir susiejant juos su bazinėmis/bendrosiomis praktikomis, todėl MARES metode galima naudoti bet kokį vertinimo modelį suderinamą su ISO 15504 standartu.

3.1.4. Viešas prieinamumas

Geriau, kai labai mažai įmonei skirtas metodas viešai prieinamas ir nemokamas, kadangi labai mažoms įmonėms gana sunku skirti pinigų vertinimo metodui, kuris turi sunkiai pamatuojamą naudą.

MARES ar TOPS metodai yra nemokami ir viešai prieinami. RAPID ir QuickScan metodai mokami.

3.1.5. Vidinio vertinimo galimybė

Vertinimą galima supaprastinti, jei vertinimą atliktų įmonės darbuotojas, kuris gerai žino įmonės procesus. Labai mažose įmonėse yra didelė tikimybė, kad atsiras darbuotojas, kuris gerai žino įmonėje vykstančius procesus. TOPS, RAPID ir QuickScan metodus būtų sunku pritaikyti vidiniam vertinimui, kadangi šie metodai reikalauja aukštos vertintojo kompetencijos ir patirties vertinant pagal ISO 15504-5. MARES yra pakankamai metodiškas, kad jį atliktų žmogus

neturintis vertinimo patirties. Vis dėlto reikėtų, kad vertintojas gerai suprastų ISO 15504 standartą ir ypač 5 standarto dalį tam, kad galėtų tinkamai interpretuoti įmonėje vykdomas praktikas ir susieti jas su standarto bazinėmis praktikomis.

3.1.6. Metodo naujumas

Naujas metodas bus labiau pritaikytas šiuolaikiniams projektams ir jame bus atsižvelgta į naujesnius proceso vertinimo tyrimus. RAPID ir TOPS metodų vystymas baigėsi apie 1999 metus ir remiasi senesne, ISO TR 15504, standarto versija. Lyginant su RAPID ir TOPS, MARES ir QuickScan naujesni. MARES paskutiniai pakeitimai padaryti 2004 metais, kai standartas dar nebuvo patvirtintas, tačiau MARES jau rėmėsi ISO 15504 standartu, kuriame vertinimo modelis buvo išimtas iš antros dalies, atnaujintas pats vertinimo modelis. Standarte, lyginant su TR (*angl.* Technical Report), pasiremta didesne procesų vertinimo patirtimi. QuickScan, komercinis vertinimo metodas, iki šiol yra naudojamas ir remiasi patvirtinta standarto versija.

3.1.7. Apibendrinimas

2 lentelėje pateiktas vertinimo metodų analizės apibendrinimas.

2 lentelė. Vertinimo metodų analizė

	MARES	RAPID	TOPS	QuickScan
Apimties nustatymas	+	-	-	-
Dydis ir sudėtingumas	-	+	+	+
Plečiamumas	+	-	-	+
Viešas prieinamumas	+	-	+	-
Vidinio vertinimo galimybė	+	-	-	-
Naujumas	+	-	-	+

MARES metodas yra pakankamai sudėtingas ir sunkiai taikomas labai mažoms įmonėms. Norint MARES pritaikyti labai mažoms įmonėms, reikia padaryti metodo pakeitimus, tačiau po tokių pakeitimų MARES gali pilnai neatitikti ISO 15504 standarto. RAPID, TOPS ir QuickScan metodai yra pakankamai paprasti, kad tiktų labai mažų įmonių vertinimui, kadangi nereikalauja sudėtingų procedūrų.

Tačiau, nors ir didesnis bei sudėtingesnis negu kiti metodai, MARES metodas yra lankstesnis, turi vertinimo apimties nustatymą, yra nemokamas ir viešai prieinamas, pakankamai naujas ir įgalina vidinį vertinimą, ir, atlikus pakeitimus, galėtų būti tinkamas labai mažos įmonės

vertinimui. Toliau nagrinėjama, kaip būtų galima sumažinti MARES metodo sudėtingumą ir pritaikyti jį labai mažoms įmonėms, tuo pačiu neprarandant metodo privalumų lyginant su kitais metodais.

3.2. Vertinimo metodo adaptavimas

Toliau darbe nuspręsta nagrinėti MARES metodo taikymą labai mažoms įmonėms, kadangi šis metodas:

- Skirtas mažoms įmonėms.
- Turi apimties nustatymo procesą.
- Laisvai prieinamas.
- Palyginus naujas.
- Labiau struktūrizuotas ir todėl pritaikomas vidiniam vertinimui.
- Plečiamas. MARES galima naudoti bet kokį procesų vertinimo modelį, modifikuoti MARES vertinimo modelį keičiant, pridėdant arba išmetant procesus.

Pakeitimai MARES metode buvo įtakoti dviejų esminių faktorių:

- Taria, kad vertinama ne maža įmonė, kuriai MARES yra kurtas, o labai maža įmonė, kurioje dirba iki 10 darbuotojų.
- Įmonės vertinimą vykdo įmonės darbuotojas.

Labai mažose įmonėse, kuriose dirba iki 10 darbuotojų, darbuotojai dirba vienoje erdvėje, žino vienas kito atliekamus veiksmus, todėl vertinimas, atliekamas darbuotojo yra efektyvesnis. Nemaža tikimybė, kad bus bent vienas darbuotojas, žinantis visus įmonėje vykstančius procesus. Jei metodas skirtas mažai įmonei, tai dar nereiškia, kad jis bus tinkamas labai mažai įmonei. Mažoje įmonėje gali dirbti iki 50 darbuotojų, tokia įmonė jau gali būti skirstoma į organizacinius padalinius, yra galimybė, kad įmonėje bus nedaug darbuotojų, kurie žinos visus įmonėje veikiančius procesus, todėl vertinimo specifika skiriasi.

Kadangi įmonės vertinimą atlieka įmonės darbuotojas, reikia suplanuoti darbuotojo laiką skirtą vertinimui, galima paprasčiau vykdyti patį procesų gebėjimo vertinimą, lengviau surinkti indikatorius.

Be to, vidinis vertinimas yra pigesnis, tai yra aktualu mažoms ir labai mažoms įmonėms. Tačiau yra didesnė rizika, kad tokio vertinimo rezultatai gali būti ne tokie tikslūs dėl to, kad įmonės darbuotojas nebus patyręs vertintojas.

Jei įmonėje nėra darbuotojo, kuris žinotų visus įmonės procesus, vidinis vertinimas tampa mažiau efektyvus, kadangi vertintojas negalėtų daugumą vertinimo sprendimų priimti pats.

3.2.1. Įmonės vertinimo planavimo procesas

MARES planavimo procesas detaliau aprašytas 1 priede.

Pagal MARES metodą vertinimo planas sudaromas planavimo proceso (PROC 1.1 Planavimas) metu. Siūlomi vertinimo planavimo proceso pakeitimai aprašyti 3 lentelėje:

3 lentelė. Pakeitimai vertinimo planavime

Procesas	Pakeitimai	Motyvacija
PROC.1.1.1 Pradinis susitikimas	Paslaugų teikimo ir konfidencialumo sutartys nepasirašomos	Įmonės vertinimą vykdo įmonės darbuotojas
	Kelias roles priskirti tiems patiems asmenims	Įmonė labai maža ir, norint kiekvienai rolei priskirti atskirą asmenį, neužtenka žmonių
PROC.1.1.2 Kontekstualizacijos planavimas	Sujungti su pradiniu susitikimu	Kadangi pradinis susitikimas tapo paprastesnis, galima pradinį ir kontekstualizacijos planavimą vykdyti viename susitikime
PROC.1.1.3 Kontekstualizacijos plano peržiūra	Vykdoma kartu su kontekstualizacijos planavimu ir ją vykdo vertinimo rėmėjas	Peržiūra daroma planavimo metu ir ją atlieka vertinimo rėmėjas užtikrindamas, kad laikas bus rezervuotas kontekstualizacijos veikloms.
PROC.1.1.4 Vertinimo vykdymo planavimas	Pakeitimų nesiūloma	
PROC.1.1.5 Vertinimo vykdymo plano peržiūra	Vykdoma kartu su vertinimo planavimu ir ją vykdo vertinimo rėmėjas	Peržiūra daroma planavimo metu ir ją atlieka rėmėjas užtikrindamas, kad laikas bus rezervuotas vertinimo veikloms.

Pradinio susitikimo metu (atitinka MARES PROC.1.1.1 ir PROC.1.1.2 veiklas) įmonės vadovui pristatomas būsimas vertinimas (vertinimo tikslas, nauda ir vertinimo žingsniai) ir sudaromas pirminis vertinimo planas remiantis MARES metode pateikta pirminio planavimo forma, kurioje nurodomi pagrindiniai vertinimo darbai ir kiek įmonės darbuotojų ir vertintojo laiko reikės kiekvienam žingsniui atlikti. Pirminiame vertinimo plane nurodyti tie patys žingsniai, kurie rekomenduojami MARES metode. Tokį planą reikia sudaryti siekiant parodyti, kiek pastangų reikės iš vertinančio darbuotojo ir kitų įmonės darbuotojų vertinimo metu.

Nereikia pasirašyti paslaugų ir konfidencialumo sutarčių, kadangi įmonės darbuotojas vykdo įmonės vertinimą, o darbuotoją ir įmonę jau sieja juridiniai išsipareigojimai.

MARES metode naudojamos šios rolės:

- Pagrindinis vertintojas, vadovaujantis vertinimo procesui.

- Vertintojo asistentas, kuris paprastai atlieką techninę vertinimo dalį: užpildo formas, registruoja faktinius darbų laikus.
- Vertinimo rėmėjas, inicijuojantis vertinimą ir užtikrinantis paramą vertinimo veikloms.
- Įmonės darbuotojai, įtraukti į vertinimą.

Jei vertinama įmonė yra labai maža, tai gali nepakakti žmonių, norint užpildyti visas roles. Kadangi pagrindinis vertintojas gali pats atlikti asistento darbus, tai pagrindinio vertintojo ir asistento rolės priskirtos įmonę vertinančiam darbuotojui. MARES metode planų peržiūras vykdo pats vertintojas arba jo asistentas, tačiau MARES metode laikomas, kad vertintojas yra išorinis ir daugumoje veiklų negaišta įmonės laiko, pavyzdžiui, susiedamas įrodymus su bazinėmis praktikomis, įvertindamas procesų lygį. Vidinio vertinimo atveju, šioms veikloms naudojamas įmonės laikas ir jį būtina suplanuoti kitų įmonės darbų kontekste, todėl būtina vertinimo rėmėjo, turinčio įtakos įmonės veiklos planavimui, peržiūra. Vertinimo rėmėjas atlieka vertinimo rezultatų peržiūras. Kadangi rėmėjas yra atsakingas už peržiūras, kontekstualizacijos ir vertinimo planų peržiūros vykdomos juos sudarant.

Į vertinimą įtraukiami visi įmonės darbuotojai. Kadangi tų darbuotojų nėra daug ir remiamasi prielaida, kad vertinimą atlieka įmonės darbuotojas, gerai išmanantis įmonės procesus, todėl kitų įmonės darbuotojų dalyvavimas reikalingas tik greitam duomenų patikslinimui.

3.2.2. Kontekstualizacija

MARES kontekstualizacijos procesas detaliau aprašytas 2 priede.

Siūlomi kontekstualizacijos proceso pakeitimai pateikti 4 lentelėje.

4 lentelė. Pakeitimai kontekstualizacijos procese

Procesas	Pakeitimai	Motyvacija
PROC.2.1.1 Charakterizacijos klausimyno pildymas	Įmonės vadovybė turi pildyti tik dalį klausimyno, kitą dalį pildo vertintojas	Kadangi įmonės vertinimą atlieka darbuotojas, tai darbuotojas gali užpildyti didesnę dalį klausimyno, ypačingai tą dalį, kurioje aprašomi įmonės produktai.
PROC.2.1.2 Charakterizacijos klausimyno analizė	Pakeitimų nesiūloma	
PROC.2.1.3 Kontekstualizacijos interviu	Pakeitimų nesiūloma	
PROC.2.2.1 Įmonės proceso	Procesas nevykdomas	Jis reikalingas tik trumpam

aprašymas		bendram įmonėje vykstančių procesų surašymui, kad vertintojas turėtų įsivaizdavimą prieš vykdant vertinimą. Kadangi vertinimą atlieka įmonės darbuotojas, kuris turi gerą įsivaizdavimą, kaip įmonėje vykdomi procesai, proceso aprašymas nereikalingas. Kiti vertinimo metodai, skirti mažoms įmonėms, nevykdo tokios veiklos ir gerinimo sprendimai priimami analizuojant silpnąsias procesų puses.
PROC.2.2.2 Tikslinio profilio nustatymas	Pakeitimų nesiūloma	

Kontekstualizacijos proceso metu nustatomas tikslinis procesų profilis. Procesas pradedamas charakterizacijos klausimyno pildymu. Pagal MARES metodą visą klausimyną turėtų pildyti vertinimo rėmėjas ir perduoti vertintojui, kuris atlieka klausimyno analizę. Kadangi vertinimą atlieka įmonės darbuotojas, dalį klausimyno gali užpildyti jis pats, ypatingai tą dalį, kurioje nurodomos kuriamų produktų savybės.

Charakterizacijos klausimyno analizės metu pagal MARES metodą reikia surasti, kokios informacijos trūksta, kad būtų galima gerai atlikti vertinimą. Vertintojas tokią informaciją gali gauti iš sponsoriaus neformalaus pokalbio metu.

Kontekstualizacijos interviu metu pateikiamas galimų įmonės tikslų sąrašas, šis sąrašas yra MARES metodo dokumente:

- Plėsti rinką.
- Neviršyti projekto biudžeto.
- Sumažinti kaštus.
- Padidinti produktyvumą.
- Padidinti klientų pasitenkinimą.
- Geriau suprasti ir patenkinti klientų poreikius.
- Sumažinti produkto pateikimo į rinką laiką (angl. *time-to-market*).
- Suspėti laiku baigti projektus.
- Užtikrinti programinės įrangos funkcionalumą.
- Užtikrinti gerą programinės įrangos naudojamumą.
- Užtikrinti gerą programinės įrangos patikimumą.

- Užtikrinti gerą programinės įrangos našumą.
- Užtikrinti gerą programinės įrangos palaikomumą.
- Užtikrinti gerą programinės įrangos perkeliamumą.
- Užtikrinti gerą programinės įrangos pakartotinį naudojamumą.

Kontekstualizacijos interviu dalyvauja ir įmonės darbuotojai, įtraukti į vertinimą. Įmonės vadovybei ir darbuotojams pristatomas vertinimas pagal ISO 15504 standartą. Pristatymas darbuotojams naudingas, siekiant, kad įmonės darbuotojai geriau suprastų vertinimo modelį ir efektyviau padėtų vertintojui. Tokiu atveju reikėtų paruošti paprastą ir aiškia ISO 15504 vertinimo pristatymo medžiagą, kurios pristatymas netruktų ilgai ir būtų skirta pristatyti bendriems vertinimo principams ir terminologijai. Pristatymo medžiagos pavyzdys yra MARES metode.

Įmonės vertinimo planavimo proceso nesiūloma keisti, išskyrus peržiūrą, kuri atliekama plano sudarymo susitikime ir ją atlieka vertinimo rėmėjas, užtikrinantis, kad vertinimui bus skirti reikalingi resursai.

3.2.3. Vertinimas

MARES vertinimo procesas detaliau aprašytas 3 priede.

Įmonės vertinimo proceso pakeitimai ir jų motyvacija apibendrinti 5 lentelėje.

5 lentelė. Pakeitimai vertinimo procese

Procesas	Pakeitimai	Motyvacija
PROC.3.1.1 Pasiruošimas vertinimui	Pakeitimų nesiūloma	
PROC.3.2.1 Duomenų rinkimas	Vykdyti paprasčiau, nevykdomi interviu su pasiruošimu, trūkstami duomenys surenkami neformalių pokalbių su darbuotojais metu	Vertinimą vykdo įmonės darbuotojas, kuris gerai žino įmonės procesus
PROC.3.2.2 Duomenų konsolidacija	Pakeitimų nesiūloma	
PROC.3.2.3 Duomenų validavimas	Pakeitimų nesiūloma	
PROC.3.3.1 Procesų gebėjimų lygių nustatymas	Vietoj popierinių formų naudoti „Appraisal Assistant“ programą	Lengviau naudoti specializuotą programinę įrangą nei popierines formas
PROC.3.3.2 Galutinis įvertinimas	Pakeitimų nesiūloma	
PROC.3.4.1 Ataskaitos	Pakeitimų nesiūloma	

Procesas	Pakeitimai	Motyvacija
ruošimas		
PROC.3.4.2 Patikrinti vertinimo suderinamumą su standartu	Procesas nevykdomas	Šis procesas nėra labai svarbus vidiniam vertinimui. Vidinio vertinimo tikslas yra surasti gerinamas procesų vietas su kiek įmanoma mažesniais resursais
PROC.3.4.3 Peržiūrėti ataskaitą	Pakeitimų nesiūloma	
PROC.3.4.4 Rezultatų pristatymas	Pakeitimų nesiūloma	

Vertinimas turėtų būti vykdomas paprasčiau, nei nurodyta MARES metode, kadangi įmonės vertinimą vykdo įmonės darbuotojas, kuris gerai žino įmonėje vykstančius procesus. MARES nurodo formalių interviu vykdymą siekiant surinkti procesų atributų indikatorius, labai mažos įmonės vidinio vertinimo metu įrodymus gali surašyti pats vertintojas ir vertinimo metu pasitarti su kitais darbuotojais dėl rastų neaiškumų procesuose.

Duomenų konsolidacijos procesas yra vykdomas siekiant rasti trūkstamus duomenis, vertintojas peržiūri surinktus duomenis ir pateikia klausimus įmonės darbuotojams. Duomenų validavimo proceso metu įmonės darbuotojai bei vadovybė peržiūri surinktus duomenis ir nurodo pataisymus.

Duomenims suvesti ir analizuoti rekomenduojama naudoti vertinimo įrankį, kuris gerokai palengvintų duomenų rinkimą ir analizę bei panaikintų popierinių MARES formų naudojimo būtinybę. Naudojant tokį įrankį, lengviau įvertinti, ar naudojamos bazinės/bendrosios praktikos, nes aiškiai matomos įmonėje vykdomos veiklos susietos su bazinėmis/bendrosiomis praktikomis.

Kaip vienas iš galimų variantų gali būti naudojama programa „Appraisal Assistant“. Ši programa buvo pasirinkta iš kitų panašių programų kadangi:

- Yra nemokama.
- Paprastai naudojama.
- Pilnai palaiko ISO 15504 5 dalį.

Kadangi MARES vertinimo modelis identiškas ISO 15504-5, tik keli procesai pašalinti, „Appraisal Assistant“ vertinimo modelis tinkamas atlikti MARES vertinimui.

Galutinio įvertinimo proceso metu surašomos įmonėje atliekamos veiklos bei gaminami produktai. Veiklos ir produktai susiejami su bazinėmis ir bendrosiomis praktikomis ir įvertinama, ar įmonėje vykdoma veikla ar sukurtas produktas įrodo bazinės/bendrosios praktikos

naudojimą. Įvertinimas nurodomas pagal skalę: pilnai vykdomas, dalinai vykdomas, didžiąja dalimi vykdomas, nevykdomas.

Pagal vykdomų bazinių/bendrųjų praktikų kiekį nustatomas procesų atributų pasiekimas pagal NPLF skalę:

- Nepasiektas (angl. *not achieved*). Yra mažai arba iš vis nėra įrodymų, kad proceso atributas buvo pasiektas.
- Dalinai pasiektas (angl. *partially achieved*). Yra įrodymų, kad atributas yra pasiektas, tačiau pasiekimas gali būti neprognozuojamas.
- Didžiąja dalimi pasiektas (angl. *largely achieved*). Yra žymūs atributo pasiekimo įrodymai, tačiau atributas nėra pilnai pasiektas.
- Pilnai pasiektas (angl. *fully achieved*). Yra pilni ir sistemingi atributo pasiekimo įrodymai.

Ataskaitą turi peržiūrėti visi įmonės darbuotojai ir nurodyti netikslumus, praleistus faktus, papildomus gerinimo sprendimus. Ataskaitos pristatymo metu pristatomi vertinimo rezultatai ir diskutuojama dėl tolimesnių gerinimo veiksmų.

3.2.4. Apibendrinimas

Pritaikius siūlomus pakeitimus gautas vertinimo procesas yra paprastesnis nei originalus MARES metodas ir labiau tinkamas labai mažoms įmonėms. Kitaip vykdomos peržiūros, vertinimo pokalbiai yra neformalūs ir nereikalauja papildomo pasiruošimo. Planų peržiūras turi vykdyti vertinimo rėmėjas, turintys įtakos veiklų planavime įmonėje, siekiant užtikrinti, kad planai bus korektiški kitų įmonės darbų kontekste. Siūloma nevykdyti dalies veiklų:

- Formalaus duomenų validavimo ir konsolidacijos.
- Faktinių vertinimo veiklų laikų registravimo.

Taip adaptuotas metodas yra pigesnis, nes nereikalingas išorinis vertintojas ir vertinimo asistentas. Pritaikytas procesas nepilnai atitinka ISO 15504 standartą, kadangi nevykdomas atitikimo standartui tikrinimas, formalus duomenų validavimas.

4. Vertinimo metodo taikymo patirtis

Pagal ankstesniame skyriuje pateiktus pasiūlymus adaptuotas MARES vertinimo metodus buvo pritaikytas labai mažoje Lietuvos įmonėje, kurioje vertinimo metu dirbo 4 darbuotojai. Įmonė kuria programinę įrangą vienai dalykinei sričiai ir parduoda produktus Lietuvos ir užsienio rinkose.

Buvo vertinamas vienas įmonės projektas, kadangi šis projektas turėtų geriausiai atspindėti visas įmonėje vykdomas praktikas. Šis projektas vertinimo metu buvo ką tik pabaigtas ir jame buvo naudojamos geriausios įmonės praktikos. Vertinimo pradinis planavimas atliktas taip, kaip aprašyta ankstesniame skyriuje. Kontekstualizacijos vykdymo metu iškilo klausimas, kaip tiksliai susieti įmonės tikslus su vertinimo apimtimi.

4.1. Kontekstualizacijoje naudojamo modelio pakeitimas

Kontekstualizacijos interviu metu vertinimo rėmėjui buvo pateiktas galimų įmonės tikslų sąrašas ir paaiškinta, ką kiekvienas iš jų reiškia, ir iš jų rėmėjas išrinko keturis tikslus:

- Suspėti laiku baigti projektus.
- Plėsti rinką.
- Užtikrinti programinės įrangos funkcionalumą.
- Užtikrinti gerą programinės įrangos naudojamumą.

MARES metode yra nurodyti procesai, susieti su įmonės veiklos tikslais, tačiau nepateikta tokio susiejimo argumentacija. Pagal MARES metode pateiktą tikslų-procesų lentelę labiausiai reiktų kreipti dėmesį į šiuos procesus:

Suspėti laiku baigti projektus:

- SPL.1 Tiekėjo pasiūlymai – gerai vykdomas bendravimas su užsakovu sumažina projekto vėlavimo riziką.
- SUP.10. Pakeitimų prašymų valdymas – nevaldant pakeitimų projekto metu, atsiranda rizika, kad bus padaryti iš anksto nesuderinti darbai ir vėluos suderinti svarbūs darbai, arba nebus laiku įgyvendinti projekto vykdymo metu atsiradę svarbūs nauji pakeitimai.
- MAN.3 Projektų valdymas – užduočių valdymas siekiant, kad projektas būtų baigtas laiku ir neviršytų biudžeto.
- MAN.5 Rizikų valdymas – sumažinti rizikingų įvykių pasekmes projekto vykdymui.

Plėsti rinką:

- OPE.2 – Klientų palaikymas. Norint plėsti rinką pirmiausia reikia išlaikyti senus klientus, po to užtikrinti palaikymą naujiems, kad parodyti, jog iškilus problemoms, jos bus sprendžiamos
- ENG.12 – programinės įrangos ir sistemos priežiūra. Svarbu valdyti programinės įrangos pakeitimus po pirminio išleidimo, nes pakeitimai dažniausiai susieti su rinkos plėtimu ir dabartinės rinkos dalies išlaikymu.
- REU.3 – dalykinės srities inžinerija. Padeda valdyti dalykinių sričių žinias, jas pritaikyti kuriant programinę įrangą labiau atitinkančią klientų poreikius.

Užtikrinti programinės įrangos funkcionalumą:

- ENG.1 – Reikalavimų išaiškinimas. Sumažina klaidų (neatitikimų poreikiams) tikimybę. Vyksta viso programinės įrangos gyvavimo ciklo metu, turi užtikrinti, kad reikalavimai atitiks kintančius klientų poreikius.
- ENG.4 – Reikalavimų analizė. Sumažina klaidų (neatitikimų poreikiams) tikimybę vėlesniuose programinės įrangos kūrimo etapuose.
- ENG.8 – Testavimas. Užtikrina nustatytą kokybės lygį bei atitikimą reikalavimams.
- SUP.1 – Kokybės užtikrinimas, procesai ir produktai turi atitikti nustatytus vidinius standartus.
- MAN.4 – Kokybės valdymas. Produktų ir paslaugų kokybės planavimas ir stebėjimas.

Užtikrinti gerą programinės įrangos naudojamumą:

- ENG.1 – Reikalavimų išaiškinimas, interfeisas turi atitikti kintančius klientų poreikius.
- ENG.4 – Reikalavimų analizė. Sumažina naudojamumo klaidų tikimybę.
- ENG.8 – Testavimas. Padeda užtikrinti kad naudojamumas atitinka nustatytus reikalavimus.
- QUA.1 – Kokybės užtikrinimas. Padeda užtikrinti, kad naudojamumas atitinka nustatytus vidinius standartus.
- MAN.4 – Kokybės valdymas. Produktų ir paslaugų naudojamumo planavimas ir stebėjimas.

MARES metodo tikslų-procesų modelis per daug paprastas, nenurodo tokio susiejimo argumentacijos, nenurodo reikalingų gebėjimo lygių ir nėra tikslus, kadangi vertinamų procesų pasirinkimas gali priklausyti nuo daugelio faktorių: nuo organizacinės kultūros, projektų

valdymo ir produktų specifikos. Šiuo metu nėra metodo ar modelio, nurodančio, kaip būtų galima tiksliai sužinoti aktualiausius procesus remiantis įmonės veiklos tikslais. Toks metodas yra kuriamas MARES autorių, taip pat toks modelis gali būti šiuo metu kuriamoje ISO 15504 standarto devintoje dalyje.

Procesų sąrašą, gautą naudojantis MARES metodu, teko koreguoti remiantis įmonės veiklos bei kuriamų produktų specifika. Norint atlikti korekcijas, reikia nustatyti motyvaciją, kodėl kiekvienas procesas buvo priskirtas kiekvienam tikslui, o MARES aprašyme tas nebuvo nenurodyta.

Vertinimo apimtį įtakojantys kriterijai paremti įmonės veiklos bei produktų specifika:

1. Įmonė neturi tiekėjų ir visus darbus atlieka pati.
2. Produktas rinkai – įmonė gamina produktus ne konkrečiam klientui, o pardavimui rinkoje.
3. Paremti kitų procesų lygius – remiantis ISO 15504 standarto procesų modeliu (ir MARES metodu) įgyvendinus tam tikrus procesus palengvėja kitų procesų lygiu įgyvendinimas. Šiuo atveju įgyvendinus pirmą projektų valdymo ir konfigūracijos valdymo procesų lygį, lengviau pasiekti kitų procesų antrą lygį.
4. Produktų ir užduočių apimties pastovumas. Jei produktai ir užduotys nepastovios apimties, t.y. procese pagaminamų produktų bei atliekamų užduočių apimtis ir specifika stipriai skiriasi kiekvienam procesui, tai reikia antro proceso gebėjimo lygio. Jei produktai ir užduotys pastovūs skirtingiems projektams, užtenka pirmo proceso gebėjimo lygio, kadangi nereikia valdyti produktų ir užduočių apimties, nereikia stebėti ir koreguoti jų būsenos. Dažniausiai tokie procesai atlieka palaikančią arba organizacinę funkciją, todėl pradžioje užtenka, kad jie būtų tik vykdomi.

Remiantis šiais kriterijais koreguojama vertinimo apimtis, sudaryta pagal MARES tikslų-procesų modelį. Kriterijai susiejami su procesais ir įvertinama iki kokio lygio procesą reikia vertinti. Reikalingas proceso ENG.6 (programinės įrangos gamyba) lygis pagal kriterijus buvo įvertintas taip:

- Ar įmonė turi daug tiekėjų, ar jų neturi iš viso, šiam procesui tai įtakos neturi. Daugelio tiekėjų atveju programinės įrangos gamybos procesas neturėtų būti įtakojamas tiekėjų darbu.
- Kriterijus „produktas rinkai“ neįtakoja gamybos proceso, kadangi šio proceso veiklos (kurti programinių vienetų testus, gaminti programinius vienetus, vykdyti programinių vienetų testus, užtikrinti vientisumą) vykdomos vienodai kuriant produktą rinkai ir kuriant produktą konkrečiam užsakovui.

- Kadangi įmonei svarbu projektus baigti laiku, o veiklų apimtis ir specifika gali skirtis skirtingiems projektams, reikia užtikrinti, kad pats procesas būtų valdomas, t.y. stebimos ir daromos jų korekcijos siekiant užtikrinti, kad reikalingas rezultatas bus pasiektas laiku. Todėl procesui reikalingas 2.1 atributas.
- Produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui, todėl reikia, kad produktai būtų valdomi. Procesui reikalingas 2.2 atributas.
- Šis procesas neremia kitų procesų lygių pasiekimo.

Panašiai buvo įvertinti kiti pradinės apimties procesai:

1. ACQ.4 – tiekėjo stebėjimas. MARES tikslų-procesų lentelė nurodo, kad turėtų būti vertinamas SPL.1 (tiekėjo pasiūlymai), bet jį vertinti netikslinga, kadangi nėra konkretaus programinės įrangos užsakovo. Vietoj jo reiktų įtraukti ACQ.4 – tiekėjo stebėjimo (angl. *supplier monitoring*) – procesą. Mažose įmonėse paprastai neužtenka kompetencijų, kurios galėtų apimti visus darbus, kurių reikia sukurti sudėtingesnį produktą. Norint padaryti tokius darbus yra dvi galimybės: pasamdyti naują žmogų arba pirkti gamybos paslaugas iš įmonės arba privataus žmogaus. Neretai samdyti naują žmogų neapsimoka, nes po to tenka mąstyti, kaip užimti likusį jo laiką. Pirkti gamybos paslaugas tokiu atveju yra geresnis sprendimas, tačiau tokiu atveju svarbu stebėti ir valdyti darbų vykdymą, kadangi vėluojantis tiekėjas gali sukelti viso projekto vėlavimą. Reikia antro šio proceso lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

2. ENG.1 – reikalavimų išaiškinimas. Šis procesas nėra reikalingas toks, koks yra pateiktas standarte, kadangi konkretaus užsakovo nėra, todėl šis procesas pašalintas iš vertinimo apimties.

3. ENG.4 – programinės įrangos reikalavimų analizė. Šis procesas įvertina reikalavimų įtaką projektui (kaina, laikas) bei nustato testavimo kriterijus. Tai yra svarbus techninis procesas, sumažinantis klaidų tikimybę vėlesniuose projekto etapuose ir taip sumažinantis projekto kainą. Šiam procesui reiktų mažiausiai antro lygio, nes reikia užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

4. ENG.7 – programinės įrangos integracija. Svarbus techninis procesas užtikrinantis programinės įrangos dalių darnų veikimą. Šiam procesui reiktų antro lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

5. ENG.8 – programinės įrangos testavimas. Galima sakyti, kad šis procesas yra

programinės įrangos integravimo dalis, nes bazinės praktikos sutampa su ENG.7. Įgyvendinus ENG.7 procesą, greičiausiai tuo pačiu bus įgyvendintas ir ENG.8. Šiam procesui reikėtų antro lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

6. ENG.12 – programinės įrangos palaikymas. Įmonėje šis procesas yra sunkiai atsiejamas nuo gamybos, testavimo ir projektų valdymo, tačiau jis vis vien vertinamas, kadangi šio proceso bazinės praktikos yra reikalingos siekiant įmonės tikslo „plėsti rinką“. Šiam procesui reikėtų antro lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

7. MAN.3 Projektų valdymas. Svarbus procesas, kuris padeda kitiems procesams pasiekti antrą lygį (valdomas, angl. *managed*). Nevykdant proceso projektas tampa chaotiškas, labai nedidelė tikimybė, kad bus baigtas laiku ir už nustatytą biudžetą. Šiam procesui užtenka pirmo lygio, kadangi šis procesas palengvina kitų procesų antro lygio pasiekimą.

8. MAN.4 Kokybės valdymas. Šio proceso tikslas yra patenkinti kliento poreikius stebint produktų ir paslaugų kokybę. Svarbus procesas, siekiant plėsti rinką. Šiam procesui reikėtų pirmo lygio, kadangi proceso užduočių ir produktų apimtis ir specifika stipriai nesikeičia kiekvienam projektui.

9. MAN.5 Rizikos valdymas. Vykdamas šį procesą, sumažinama rizikų tikimybė bei jų poveikis, jis padeda siekti, kad projektas būtų baigtas laiku. Šiam procesui reikėtų antro lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

10. OPE.2 – Klientų palaikymas. Norint plėsti rinką pirmiausia reikia išlaikyti senus klientus, po to užtikrinti palaikymą naujiems, norint parodyti, jog iškilus problemoms, jos bus sprendžiamos. Reikėtų pirmo lygio, kadangi procesas vykdomas visiems projektams ir jo apimtis nesikeičia.

11. REU.2 Dalykinės srities inžinerija. Padeda valdyti dalykinių sričių žinias, jas pritaikyti kuriant programinę įrangą labiau atitinkančią klientų poreikius. Šiam procesui reikėtų pirmo lygio, kadangi procesas vykdomas visiems projektams ir jo apimtis nesikeičia.

12. SUP.1 Kokybės užtikrinimas. Procesai ir produktai turi atitikti nustatytus vidinius standartus. Šiam procesui reikėtų antro lygio, norint užtikrinti, kad pats procesas būtų valdomas bei valdomi jo produktai, kadangi veiklų ir produktų apimtis bei specifika gali stipriai keistis kiekvienam projektui.

13. SUP.8 – konfigūracijos valdymas. Valdomi darbo produktai, kitiems procesams padeda pasiekti 2 lygį. Procesui užtenka pirmo lygio, norint paremti kitų procesų antrą lygį.

14. SUP.10 – pakeitimų prašymų (angl. *change request*) valdymas. Užtikrinama, kad į projektą įtraukiami reikalingi pakeitimai jo vykdymo metu, taip pat užtikrina, kad tie pakeitimai yra tinkamai valdomi. Pakeitimų valdymas yra labai svarbus šiuolaikiniams produktams, skirtiems parduoti rinkoje, kadangi klientų poreikiai greitai keičiasi, keičiasi mados, konkurentai vis sugalvoja kažką naujo, projekto metu dažnai gaunama nauja marketingo informacija. Šiam procesui užtektų pirmo lygio, kadangi jis jau užtikrina, kad pakeitimai yra registruojamai ir stebimi (valdomi darbo produktai), taip pat įvertinama įtaka projekto vykdymui.

6 lentelėje pateiktas procesų lygių pasirinkimo apibendrinimas pagal kriterijus, joje nurodyta kokio gebėjimo lygio reikia kiekvienam procesui remiantis kriterijumi.

6 lentelė. Procesų pasirinkimas pagal kriterijus

Procesas	1 – daug tiekėjų	2 – produktas rinkai	3 – paremti kitų procesų lygius	4 – nepastovios apimtys	4 – pastovios apimtys
Suspėti laiku baigti projektus					
ACQ.4 – tiekėjo stebėjimas	0			2	1
SPL.1 – tiekėjo pasiūlymai		0		2	1
SUP.10 – pakeitimų prašymų valdymas				2	1
MAN.3 – projektų valdymas				2	1
MAN.5 – rizikos valdymas				2	1
Plėsti rinką					
OPE.2 – klientų palaikymas				2	1
ENG.12 – programinės įrangos palaikymas				2	1
REU.3 – dalykinės srities inžinerija				2	1
Užtikrinti programinės įrangos funkcionalumą					
ENG.1 – reikalavimų išaiškinimas		0		2	1
ENG.4 – programinės įrangos reikalavimų analizė				2	1
ENG.8 – programinės įrangos testavimas				2	1
SUP.1 – kokybės užtikrinimas				2	1
MAN.4 – kokybės valdymas				2	1
Užtikrinti gerą programinės įrangos naudojamumą					

Procesas	1 – daug tiekėjų	2 – produktas rinkai	3 – paremti kitų procesų lygius	4 – nepastovios apimties	4 – pastovios apimties
ENG.1 - reikalavimų išaiškinimas		0		2	1
ENG.4 – programinės įrangos reikalavimų analizė				2	1
ENG.8 – programinės įrangos				2	1
SUP.1 – kokybės užtikrinimas				2	1
MAN.4 – kokybės valdymas				2	1
Papildomi procesai					
MAN.3 – projektų valdymas			1	2	1
SUP.8 – konfigūracijos valdymas			1	2	1

6 lentelėje vertikalčiai kairėje pateikti procesai susieti su įmonės tikslais, horizontalčiai viršuje pateikti procesų gebėjimo lygių pasirinkimo kriterijai, 1-3 kriterijai susiję su įmonės veiklos specifika, 4 kriterijus susijęs su kiekvieno proceso vykdymo specifika.

Jei įmonės tikslas yra vienas iš nurodytų lentelėje, reikalingas visų su tikslu susietų procesų vertinimas, t.y. pradžioje šiems procesams reikalingas bent pirmas lygis. Toliau yra žiūrimi kriterijai iš kairės į dešinę. Jei kriterijus nurodo, kad procesui reikalingas 0 lygis, t.y. proceso vertinimas nereikalingas, toliau nedaroma nieko ir šis procesas šalinamas iš vertinimo apimties. Jei po pirmų 3 kriterijų peržiūros procesas nepašalintas iš vertinimo apimties, analizuojamas 4 kriterijus. Šis kriterijus vertinamas kiekvienam procesui atskirai, t.y. žiūrima, ar proceso darbų ir produktų apimtis kinta, ar lieka daugmaž vienodi kiekvienam įmonės projektui.

Papildomi procesai, nurodyti 6 lentelėje, turi būti vertinami iki 1 pirmo lygio, jei vertinimo apimtyje yra bent vienas procesas, kuris turi būti vertinamas iki 2 lygio.

Šį modelį galima panaudoti kitų įmonių, turinčių panašius tikslus ir veiklos specifika, vertinimo apimties nustatymui. Ketvirtas kriterijus gali būti naudojamas bet kokios įmonės atveju sprendžiant, ar procesas turi būti valdomas. Apibrėžiant vertinimo apimtį kitoms įmonėms (turinčioms kitokius tikslus, kitokią veiklos specifika) tokio modelio greičiausiai nepakaks. Kriterijų sąrašą greičiausiai reikės papildyti pagal įmonės veiklos tikslus bei specifika bei nustatyti kokio lygio reikia procesui pagal nustatytus kriterijus. Reikėtų nustatyti įmonės veiklos specifika, nustatyti, kokie kriterijai su tuo susiję ir įvertinti kaip šie kriterijai įtakoja vertinimo apimties procesų gebėjimo lygius. Esamas modelis gali būti panaudotas iš dalies, tiems

kriterijams, kurie atitiks tos įmonės tikslus.

4.2. Procesų vertinimas

4.2.1. Programinės įrangos gamybos proceso (ENG.6) vertinimas

Procesas laikomas pasiekęs atitinkamą lygį, jei žemesnių lygių atributai yra pilnai pasiekti, o atitinkamo lygio atributai pasiekti didžiąja dalimi.

Pavyzdžiui, programinės įrangos gamybos procesas buvo įvertintas nuliniu lygiu taip:

- Įrodymas „sukurtas programinės įrangos vienetas“ buvo priskirtas bazinei praktikai „ENG.6.BP2 gaminami programinės įrangos vienetai“. Šio vieno įrodymo užtenka, kad būtų galima teigti, jog ši bazinė praktika yra pilnai vykdoma.
- Kitoms trims šio proceso bazinėms praktikoms (ENG.6.BP1 sukurti programinių vienetų verifikavimo procedūras, ENG6.BP3 užtikrinti vientisumą, ENG6.BP4 verifikuoti programinius vienetus) įrodymų nerasta.
- Iš keturių bazinių praktikų vykdoma tik viena, todėl negalima teigti, kad procesas pasiekia tikslą „sukurti programinius vienetus atitinkančius numatytą projektą“, todėl vykdymo atributas pasiekiamas tik dalinai (angl. *partially achieved*), kadangi programiniai vienetai sukuriami, tačiau neužtikrinamas jų atitikimas projektui.
- Kadangi vykdymo atributas pasiekiamas tik dalinai, tai procesas nepasiekia pirmo lygio.
- Proceso vykdymo atributas pasiektas tik dalinai, todėl kiti proceso atributai nebuvo vertinami.
- Proceso įvertintas nuliniu lygiu.

7 lentelėje pateiktas bazinių praktikų įvertinimas.

7 lentelė. Programinės įrangos gamybos proceso (ENG.6) įvertinimas

Bazinė praktika	Įvertinimas
ENG.6.BP1 sukurti programinių vienetų verifikavimo procedūras	Vykdoma pilnai
ENG.6.BP2 gaminami programinės įrangos vienetai	Įrodymų nerasta
ENG6.BP3 užtikrinti vientisumą	Įrodymų nerasta
ENG6.BP4 verifikuoti programinius vienetus	Įrodymų nerasta
Atributas	Įvertinimas
PA 1.1 Proceso atlikimas	Pasiektas iš dalies

4.2.2. Projektų valdymo proceso (MAN.3) vertinimas

- Šiam procesui surinkti įrodymai:
 - Produkto savybių ir darbų sąrašas su priskirtomis iteracijomis ir laiko įvertinimais.
 - Sudaromas darbų sąrašas kiekvienai savybei ir įtraukiamas į produkto savybių ir darbų sąrašą.
 - Iteracijos pradžioje darbai priskiriami darbuotojams. Darbai surašomi į užduočių valdymo sistemą ir jiems priskiriami darbuotojai.
 - Iteracijos pradžioje peržiūrima, kas padaryta buvusioj iteracijoje, nustatomas progresas ir reikalingi korekciniai veiksmai, korekciniai veiksmai ir plano korekcijos įtraukiamos į produkto savybių ir darbų sąrašą.
 - Projekto pradžioje įvertinamos reikalingos kompetencijos iš įmonės darbuotojų ir tiekėjų, vertinimas vykdomas susitikimo metu.
 - Vykdomos darbų sąrašė nurodytos veiklos.
- Bazinė praktika „MAN.3.BP1 nurodyti darbo apimtį“ pilnai vykdoma, kadangi apimtis nurodoma savybių ir darbų sąrašė (1 įrodymas).
- Bazinė praktika „MAN.3.BP2 nurodyti projekto gyvavimo ciklą“ pilnai vykdoma, kadangi iteratyvus gyvavimo ciklas nustatomas pildant savybių ir darbų sąrašą (1 įrodymas).
- Bazinė praktika „MAN.3.BP3 įvertinti projekto įgyvendinamumą“ vykdoma pilnai, kadangi projekto pradžioje įvertinamos reikalingos kompetencijos iš įmonės darbuotojų ir tiekėjų (5 įrodymas) ir sudaroma projekto apimtis (1 įrodymas) ir taip įvertinamas įgyvendinamumas laiko atžvilgiu.
- Bazinė praktika „MAN.3.BP4 įvertinti projekto įgyvendinamumą“ vykdoma pilnai, kadangi iteracijos pradžioje peržiūrima, kas padaryta buvusioj iteracijoje, nustatomas progresas ir reikalingi korekciniai veiksmai, korekciniai veiksmai ir plano korekcijos įtraukiamos į produkto savybių ir darbų sąrašą.
- Bazinė praktika „MAN.3.BP5 nustatyti projekto užduotis“ yra vykdoma pilnai, kadangi projekto užduotys įtraukiamos į produkto savybių ir darbų sąrašą (1 įrodymas).
- Bazinė praktika „MAN.3.BP6 nustatyti projektui reikalingą patirtį ir kompetencijas“ vykdoma pilnai, kadangi projekto pradžioje įvertinamos reikalingos

kompetencijos iš įmonės darbuotojų ir tiekėjų (5 įrodymas).

- Bazinė praktika „MAN.3.BP7 sudaryti projekto grafiką“ vykdoma pilnai, kadangi savybių ir užduočių sąrašė (1 įrodymas) užduotys ir savybės priskiriamos iteracijoms, kurios išdėstomos laike.
- Bazinė praktika „MAN.3.BP8 identifikuoti ir stebėti projekto sąsajas“ yra nevykdoma, kadangi nerasta su ja susietų įrodymų.
- Bazinė praktika „MAN.3.BP9 priskirti atsakomybes“ vykdoma pilnai, nes kiekvienos iteracijos pradžioje darbai priskiriami darbuotojams (3 įrodymas).
- Bazinė praktika „MAN.3.BP10 nustatyti projekto planą“ vykdomas pilnai, kadangi sudarytas produktų savybių ir darbų sąrašas su priskirtomis iteracijomis ir įvertinimais (1 įrodymas).
- Bazinė praktika „MAN.3.BP11 vykdyti projekto planą“ vykdoma pilnai, kadangi darbų sąrašė nurodytos veiklos yra vykdomos (6 įrodymas).
- Bazinės praktikos „MAN.3.BP12 stebėti projekto atributus“, „MAN.3.BP13 peržiūrėti projekto progresą“, „MAN.3.BP14 vykdyti korekcinius veiksmus“ vykdoma pilnai, nes iteracijos pradžioje peržiūrima, kas padaryta buvusioje iteracijoje, nustatomas progresas ir reikalingi korekciniai veiksmai, korekciniai veiksmai ir plano korekcijos įtraukiamos į produkto savybių ir darbų sąrašą (4 įrodymas).
- Bazinė praktika „MAN.3.BP15 vykdyti projekto patirties apibendrinimą“ yra nevykdoma, kadangi nerasta su ja susietų įrodymų.
- Kadangi iš penkiolikos bazinių praktikų pilnai vykdoma trylika, galima teigti, kad pilnai pasiektas vykdymo atributas ir pirmas proceso lygis. Nevykdomos praktikos „vykdyti projekto patirties apibendrinimą“ ir „identifikuoti ir stebėti projekto sąsajas“ netrukdo pasiekti proceso tikslą „identifikuoti, koordinuoti ir stebėti veiklas, užduotis ir resursus, reikalingus projektui, kad būtų galima pagaminti produktą ar suteikti paslaugą projekto reikalavimų ir apribojimų kontekste“.

Tokiu pat būdu įvertinti kiti vertinimo apimtyje nurodyti procesai. 8 lentelėje pateiktas kitų procesų bazinių praktikų vertinimas.

8 lentelė. Įmonės procesų bazinių praktikų vertinimas

Procesas	Nevykdoma	Iš dalies vykdoma	Vykdoma didžiąja dalimi	Vykdoma
ACQ.4		BP3, BP4, BP5		BP1, BP2

Procesas	Nevykdoma	Iš dalies vykdoma	Vykdoma didžiąja dalimi	Vykdoma
ENG.4	BP4	BP1		BP2, BP5, BP6
ENG.6	BP1, BP3, BP4			BP2
ENG.7	BP1, BP2, BP5, BP6	BP3, BP4		
ENG.8	BP1, BP3	BP2		
ENG.12	BP5	BP1, BP3, BP4		BP2, BP6
MAN.3	BP8, BP15			BP1– BP7, BP9 – BP14
MAN.4	BP1 – BP8			
MAN.5	BP1 – BP7			
OPE.2	BP3 – BP6			BP1, BP2
REU.3	BP1 – BP6			
SUP.1	BP1 – BP5			
SUP.8	BP5, BP8, BP9			BP1 – BP4, BP6, BP7, BP10
SUP.10	BP6	BP9		BP1 – BP5, BP7, BP8

4.2.3. Vertinimo rezultatai

Atlikus įmonės vertinimą buvo nustatyta, kad šie procesai yra pirmo (vykdomas) lygio:

- ACQ.4 - tiekėjo stebėjimas.
- MAN.3 – projekto valdymas.
- SUP.8 – konfigūracijos valdymas.
- SUP.10 – pakeitimų prašymų valdymas.

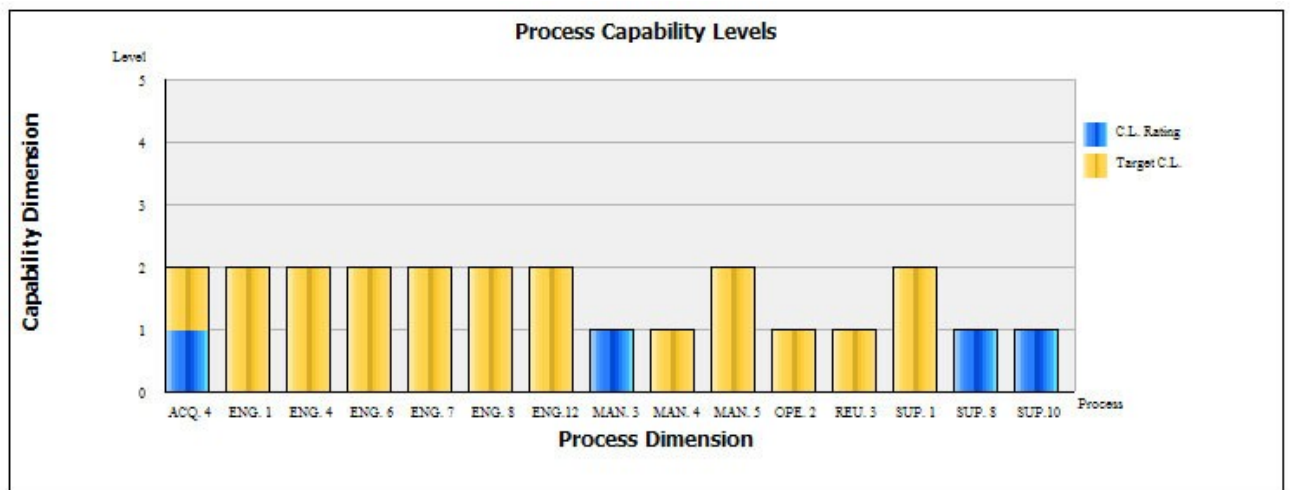
Kiti procesai yra nulinio lygio, tai reiškia, kad proceso bazinės praktikos nėra vykdomos arba vykdomos nepakankamai, kad būtų galima laikyti, kad vykdymo atributas yra pasiektas bent didžiąja dalimi. Vykdam vertinimą buvo išryškintos procesų silpnosios ir stipriosios vietos (9 lentelė):

9 lentelė. Įmonės proceso silpnosios ir stipriosios vietos

Procesas	Silpnosios vietos	Stipriosios vietos
ACQ.4 – tiekėjo stebėjimas	Nėra vykdomos tiekėjo darbo progreso peržiūros.	
ENG.4 – reikalavimų analizė	Nedaromi reikalavimų priėmimo testai	
ENG.6 – programinės įrangos	Nedaromi formalūs	

Procesas	Silpnosios vietos	Stipriosios vietos
gamyba	programinių vienetų (angl. <i>unit</i>) testai, testavimas vykdomas chaotiškai ir priklauso tik nuo programuotojo noro vykdyti testavimą	
ENG.7 - programinės įrangos integravimas	Formalūs integraciniai testai nėra vykdomi	
ENG.8 – programinės įrangos testavimas	Tos pačios problemos kaip ir ENG.7	
MAN.3 – projektų valdymas		1 lygis. Naudojamos judriojo planavimo ir vertinimo (angl. <i>agile estimating and planning</i>) praktikos suteikiančios lankstumo ir paprastumo projektų valdyje ir tuo pačiu užtikrina pirmą proceso lygį
MAN.4 – kokybės valdymas	Nėra vykdomas	
MAN.8 – rizikos valdymas	Nėra vykdomas	
ENG.12 – programinės įrangos palaikymas	Vykdomas iš dalies	Yra gamybos, testavimo, projektų valdymo dalis, kas savaime užtikrina kai kurių bazinių praktikų vykdymą
OPE.2 – klientų palaikymas	Klientų pasitenkinimas produktu nėra matuojamas	
REU.3 – dalykinės srities inžinerija	Procesas nėra vykdomas	
SUP.1 – kokybės užtikrinimas	Procesas nėra vykdomas	
SUP.8 – konfigūracijos valdymas		Procesas pasiekia pirmą lygį todėl, kad naudojama versijavimo sistema, nustatomos išleidimų versijos,

Procesas	Silpnosios vietos	Stipriosios vietos
		valdomi visi projekto darbo produktai.
SUP.10 – pakeitimų prašymų valdymas		Judriojo planavimo ir vertinimo praktikos užtikrina, kad bus pasiektas pirmas šio proceso lygis. Šios praktikos leidžia keisti projekto darbus bei planą projekto vykdymo metu. Taigi, pakeitimų prašymų valdymas vertinamoje įmonėje yra projekto valdymo proceso dalis.



2 pav. Tikslinis įmonės procesų profilis ir faktiniai lygiai

Paruošta vertinimo ataskaita buvo pristatyta įmonės vadovybei. Galutinėje vertinimo ataskaitoje buvo pateiktos gerinimo rekomendacijos, gerinimo gairės ir detalus gerinimo planas.

4.3. Vertinimo patirtis

Vertinimo metu buvo sukaupta ir apibendrinta ši patirtis:

1. ISO 15504 standarto 5 dalyje pateiktas vertinimo modelis yra pritaikomas labai mažos įmonės proceso vertinimui. Šiuo modeliu paremtas MARES metodas, kuris reikalauja, kad iš visų modelio procesų, vertinimui būtų parinkta nedidelė procesų aibė. MARES pateikia gaires, kaip tuos procesus išrinkti. Modifikuoto MARES vertinimo apimtis yra

- apie 20-30 valandų, neskaitant darbuotojo paruošimo vertinimui, tačiau paruošti darbuotoją reikia tik vieną kartą.
2. Labai mažą įmonę vertinti efektyviausia, kai tai daro įmonės darbuotojas, kadangi tokiam vertinimui reikia mažiau pastangų, galima taikyti mažiau formalius metodus. Tokiu atveju galima naudoti neformalius interviu, vertintojas gali pats užpildyti dalį klausimynų, surašyti dalį bazinių/bendrujų praktikų indikatorių, nereikia pasirašyti formalių dokumentų, kaip paslaugų ir konfidencialumo sutarčių, nevykdyti proceso stebėjimo. Jei įmonėje nėra darbuotojo, kuris žinotų visus įmonės procesus, vidinis vertinimas nėra efektyvus. Antra vertus, mažai tikėta, kad įmonės darbuotojas bus patyręs vertintojas, turintis didelę vertinimo patirtį, todėl vertinimo kokybė gali būti žemesnė.
 3. Labai mažos įmonės vertinimas turi būti kuo lengvesnis, todėl tikslų įvertinimas yra būtinas, kadangi padeda sumažinti vertinimo apimtį iki svarbiausių procesų, MARES vertinimo procesas reikalauja tą padaryti. Įmonės vertinimo metu buvo pasiremta MARES tikslų-procesų modeliu, kuris buvo patikslintas pagal nustatytus kriterijus. Sudarytas procesų lygių pasirinkimas pagal kriterijus gali būti pavyzdys apibrėžiant panašios specifikos įmonių vertinimo apimtį.
 4. Nustatant įmonės tikslus buvo naudojamas MARES tikslų-procesų modelis, todėl įmonės tikslų nustatymas ir vertinimo apimties apibrėžimas buvo gana paprastas, tačiau apimtį vėliau teko patikslinti.
 5. Tikslaus modelio, susiejančio įmonės tikslus su procesais ir jų lygiais nėra. Šiuo metu tokie modeliai kuriami. Todėl išskirtų procesų sąrašą reikia koreguoti pagal įmonės veiklos ir produktų specifiką. Tam reikia gerai suprasti naudojamą vertinimo modelį.
 6. Vertinimui reikėtų naudoti įrankį, padedantį surinkti ir apibendrinti duomenis, kadangi toks įrankis atlieka rutininius vertinimo darbus ir sutaupo laiko. Nereikia pildyti formų, lengviau įvertinti, ar naudojamos bazinės praktikos, nes aiškiai matomos įmonėje vykdomos veiklos susietos su bazinėmis praktikomis.

5. Rekomendacijos labai mažų įmonių programų kūrimo proceso vertinimui

Apibendrinant pateiktus pasiūlymus programų kūrimo proceso vertinimo metodo adaptavimui labai mažoms įmonėms, bei patirtį, gautą pritaikius tokį adaptuotą modelį konkrečiai įmonei, suformuluotos šios rekomendacijos:

1. Vertinimą turėtų vykdyti įmonės darbuotojas, tuomet procesų gebėjimų lygių nustatymas supaprastėja.
2. Reikia užtikrinti, kad įmonės darbuotojas įgaus pakankamą kompetenciją vykdyti įmonės vertinimą ir taip sumažinti riziką, kad vertinimo metu bus padaryta klaidų.
3. Vertinimo rėmėjas, turintis įtakos įmonės darbo laiko paskirstymui, turi atlikti planų peržiūras, siekiant užtikrinti, kad vertinimui bus skirtas reikalingas laikas.
4. Vertintojas ir vertintojo asistentas turėtų būti tas pats asmuo.
5. Jei reikia, į vertinimą galima įtraukti visus įmonės darbuotojus.
6. Proceso gebėjimo lygių įvertinimui naudoti specializuotą programą. Rašant darbą geriausias pasirinkimas buvo „Appraisal assistant“.
7. Naudinga supažindinti darbuotojus su vertinimu pagal ISO 15504 standartą.
8. Nustatant procesų gebėjimo lygius užtenka remtis neformaliais pokalbiais su įmonės darbuotojais, o ne vykdyti formalius interviu.
9. Nerekomenduojama vykdyti vertinimo proceso faktinių laikų stebėjimo.
10. Nustatyti vertinimo apimtį reiktų remiantis paprastu ir lengvai suprantamu tikslų-procesų modeliu.
11. Galima nevykdyti vertinimo suderinamumo su ISO 15504 standartu tikrinimo.

Šios gairės gali būti panaudotos kuriant naują labai mažų įmonių vertinimo metodą arba tobulinant esamus metodus. Tokiame vertinimo metode taip pat galima būtų pateikti greitus gerinimo sprendimus, paremtus judriosiomis praktikomis parodant, kad tokios praktikos užtikrins reikiamą proceso lygį.

6. Labai mažos įmonės programų kūrimo proceso gerinimas

6.1. Gerinimo planavimas

Pristačius įmonės vertinimo rezultatus buvo aptartos galimos gerinimo galimybės. Nuspręsta prieš gerinant valdymo procesus (rizikų valdymas, kokybės valdymas) pirmiausia pagerinti technologinius procesus, kadangi šie procesai sukuria realią vertę. Jei šie procesai nėra gerai vykdomi, iš pagalbinių ir valdymo procesų nėra daug naudos. Buvo pasirinkta viena technologinė gerinimo sritis: testavimas ir programinės įrangos kokybės užtikrinimas. Ši sritis susieta su įmonės veiklos tikslu: užtikrinti programinės įrangos funkcionalumą. Šie procesai yra susieti su pasirinkta gerinimo sritimi:

- ENG.6 – programinės įrangos gamyba.
- ENG.7 – programinės įrangos integravimas.
- ENG.8 – programinės įrangos testavimas.

Programinės įrangos gamybos proceso kokybę užtikrina programinių vienetų kokybę, sumažina klaidų kiekį vėlesniuose programinės įrangos kūrimo etapuose ir sumažina pastangas, reikalingas vėlesniuose programinės įrangos etapuose.

Kokybiškas programinės įrangos integravimo procesas užtikrina mažesnę verslo logikos klaidų tikimybę bei apsaugo nuo regresinių klaidų.

Programinės įrangos testavimas yra integravimo proceso dalis.

Šie procesai bus gerinami naudojantis judriojo (angl. *agile*) testavimo praktikomis [CG09]. Judriojo testavimo praktikos nereikalauja didelio kiekio dokumentacijos, reikalavimų specifikavimo raštu, tačiau reikalauja aukšto testavimo disciplinos lygio ir bendravimo komandos viduje. Todėl šios praktikos tinka mažos įmonės arba komandos veikloje. Vis dėlto šios praktikos reikalauja investuoti laiką į testų automatizavimą ir į gaminamo produkto testuojamumą didinimą.

Gerinimas bus vykdomas keliais žingsniais:

1. Pradėti naudoti formalius automatizuotus programinių vienetų testus.
2. Pradėti naudoti formalius automatizuotus integracinius verslo logikos testus.
3. Pradėti vykdyti apkrovos testavimą.

Testavimo gerinimas bus vykdomas vienam produktui, po to pagerinimas bus pritaikytas kitiems produktams, todėl vykdant gerinimą reiktų numatyti, kad priimti sprendimai bus panaudojami kituose projektuose. Pateiktos programinių vienetų (6 priedas) ir integracinių testų (7 priedas) automatizavimo gairės.

6.1.1. Detalus gerinimo planas

Pagal gerinimo gaires sudarytas detalus gerinimo planas su įvykdymo datomis. Detalus gerinimo planas pateiktas prieduose. Pagal šį planą vykdomas įmonės gerinimas keliais žingsniais:

- pasirinkti ir įdiegti reikiamus įrankius;
- sudaryti testavimo procedūrą;
- padaryti techninius pakeitimus produktuose;
- užtikrinti automatinę testinės ir produkcinės versijos sukūrimą;
- parašyti bandomuosius testus;
- įdiegti matavimus;
- įtvirtinti pagerinimą.

6.1.2. Gerinamų procesų padengimas pagal ISO 15504 naudojantis judriuoju testavimu

Reikia įvertinti, ar numatytas gerinimas leidžia pasiekti 2 lygį šiems procesams:

- ENG.6 – programinės įrangos gamyba,
- ENG.7 – programinės įrangos integravimas,
- ENG.8 – programinės įrangos testavimas.

6.1.2.1. Programinės įrangos gamyba

Šio proceso tikslas yra sukurti programinius vienetus, kurie atitiktų programinės įrangos projektą.

Proceso rezultatai:

1. Nustatytas programinių vienetų atitikimo reikalavimams verifikavimo kriterijus.
2. Pagaminti programinius vienetus pagal projektą.
3. Darnumas ir trasuojamumas nustatyti tarp reikalavimų, projekto ir programinių vienetų
4. Atliktas programinių vienetų verifikavimas pagal reikalavimus.

8 priede parodyta, kaip senasis įmonės procesas ir pagerintas procesas padengia pirmo ir antro gebėjimo lygių bazines ir bendrąsias praktikas.

Įgyvendinus programinės įrangos gamybos proceso gerinimą remiantis judriojo testavimo praktikomis proceso valdymo atributas bus pilnai pasiektas, o produktų valdymo atributas bus pasiektas didžiaja dalimi, todėl bus pasiektas antras šio proceso lygis.

6.1.2.2. Programinės įrangos integravimas

Šio proceso tikslas – sujungti programinius vienetus į veikiančią programinę įrangą.

atitinkančią reikalavimus.

Proceso rezultatai:

1. Sukurta integravimo strategija, atitinkanti programinės įrangos reikalavimus ir architektūrą.
2. Sukurti integruotos programinės įrangos verifikavimo kriterijai.
3. Programinė įranga verifikuota remiantis sudarytais kriterijais.
4. Programinė įranga sukurta remiantis integravimo strategija.
5. Integracinio testavimo rezultatai užfiksuoti.
6. Regresinio testavimo strategija sukurta, kad būtų galima pakartotinai verifikuoti programinę įrangą po pakeitimų.

9 priede parodyta, kaip senasis įmonės procesas ir pagerintas procesas padengia pirmo ir antro gebėjimo lygių bazines ir bendrąsias praktikas.

Įgyvendinus programinės įrangos integravimo proceso gerinimą remiantis judriojo testavimo praktikomis proceso valdymo atributas bus pilnai pasiektas, o produktų valdymo atributas bus pasiektas didžiąja dalimi, todėl bus pasiektas antras šio proceso lygis.

Trys programų sistemos testavimo proceso bazinės praktikos sutampa su integravimo proceso praktikomis, todėl integravimo procesui pasiekus antrą lygį, testavimo procesas irgi pasieks antrą lygį.

6.2. Pradinio gerinimo plano pakeitimas ir motyvai

Atlikus įmonės vertinimą buvo sudarytas pradinis gerinimo planas. Jis buvo pristatytas įmonės vadovybei. Iš karto pasidarė aišku, kad įgyvendinami pakeitimai užimtų per daug resursų, kurie labai mažai įmonei yra per daug svarbūs. Gerinimas užimtų 152 valandas, o tai yra per daug įmonei, kurioje dirba 5 darbuotojai. Todėl pirmąjį gerinimo ciklą nuspręsta sumažinti remiantis tuo, kad gerinimo veiksmai duos apčiuopiamą ir nesunkiai paskaičiuojamą naudą.

Pakeistas planas pateiktas 10 lentelėje.

10 lentelė. Pakeistas gerinimo planas

	Veiksmas	Pabaigos data	Laikas(val.)
1	Pasirinkti testavimo įrankius programinių vienetų ir integraciniam testavimui	2010-03-01	8
2	Įdiegti pasirinktus įrankius	2010-03-20	8
3	Susieti naujus įrankius su dabar naudojamais įrankiais (pagrinde su SVN)	2010-03-22	2
4	Įdiegti pasirinktus įrankius taip, kad būtų galima automatiškai sukurti testinę ir produkcinę produkto versijas	2010-03-28	24

	Veiksmas	Pabaigos data	Laikas(val.)
5	Parašyti kelis bandomuosius programinių vienetų ir integracinius testus ir užtikrinti automatinį jų vykdymą	2010-04-04	16

Supaprastintas gerinimas užimtų apie 58 valandas ir orientuotas į automatinį testinės ir produkcinės versijos sukūrimą. Įmonė vykdo naujos versijos išleidimą kas 1-2 savaites, kiekvienam išleidimui reikia padaryti mažiausiai 4 testinės versijos sukūrimus ir 1 produkcinės platformos sukūrimą. Testinės ir produkcinės platformos sukūrimas užima apie 1 valandą neskaičiuojant žmogiškų klaidų taisymo po sukūrimo. Taigi po 12 išleidimų investuotas laikas atsipirks.

Programinių vienetų testų ir integracinių testų automatizavimo atsipirkimą suskaičiuoti gerokai sunkiau. Reikia įvesti testavimo matavimus, nustatyti kiek laiko užima testų sudarymas, apmokymas tokių testavimų vykdyti, ir kiek laiko sutaupo jų vykdymas. Testų vykdymo naudą apskaičiuoti dar sudėtingiau: reikia kiek laiko sutaupo rečiau atsirandančios regresinės klaidos, kiek laiko sutaupo programinių vienetų klaidų nebuvimas prieš integracinį testavimą, kiek laiko sutaupo iš karto rastos klaidos. Skaičiuoti programinių vienetų atsiperkamumą tiesiog per daug sudėtinga, kad iš viso vertėtų tą daryti. Todėl šią pagerinimo dalį nuspręsta daryti atskirai, tik pirmoje gerinimo dalyje numatyti įvykdyti kelis bandomuosius testus, siekiant parodyti, kad tokie testai veikia ir apytiksliai nustatyti, kiek gali užimti tokių testų sudarymas ir kiek reikės architektūrinių programų sistemos pakeitimų.

6.3. Įrankių pasirinkimas

Prieš gerinimą įmonėje buvo naudojamas Zend karkasas, skirtas palengvinti PHP programų kūrimą, naudojama išeitinio kodo valdymo sistema SVN.

Reikėjo kelių įrankių:

- Nenutrūkstamos integracijos serverio.
- Integravimo įrankio (angl. *build*).
- Testavimo įrankio.

Buvo nagrinėjamos dvi nenutrūkstamos integracijos serverio ir integravimo įrankio derinio alternatyvos: CruiseControl+Ant ir Xinc+Phing. Nagrinėjami įrankių deriniai, kadangi nenutrūkstamos integracijos serveris CruiseControl diegiamas kartu su integravimo įrankiu Ant, o nenutrūkstamos integracijos serveris Xinc – kartu su integravimo įrankiu Phing, todėl reikalauja mažiausiai pastangų norint pradėti naudoti. Nagrinėjant įrankius buvo naudojami šie kriterijai:

- Vystymo laikas – seniau vystomi produktai yra stabilesni ir labiau naudojami ir

didesnis spaudimas įrankį vystyti toliau.

- Patogi vartotojo aplinka, kurioje galima stebėti vykstančias integracijas, paskutinės integracijos laiką ir būseną.
- Vartotojo aplinka turi suteikti galimybę aktyvuoti integraciją rankiniu būdu, tai reikalinga produkcinės versijos sukūrimui.
- Galimybė kviesti komandinės eilutės programas – svarbu, nes suteikia daug lankstumo.
- Galimybė vykdyti SSH komandas nutolusiame serveryje – reikalinga sukuriant testinę ir produkcinę versijas.
- Duomenų bazės versijavimo įrankis – reikalingas, norint užtikrinti, kad į duomenų bazę būtų rašomi tik nauji pakeitimai.
- Automatinis integravimo sėkmės/nesėkmės raportavimas el.paštu – svarbu gauti pranešimus apie nepavykusias integracijas, kad būtų galima greitai ištaisyti padarytas klaidas.

11 lentelėje pateikta įrankių analizė pagal pasirinktus kriterijus.

11 lentelė. Automatinio integravimo įrankio pasirinkimas

Kriterijus	CruiseControl+Ant	Xinc+Phing
Vystymo laikas	seniai	vystomas nuo maždaug 2006 metų
Patogi vartotojo aplinka	taip	taip
Galima lengvai aktyvuoti integraciją rankiniu būdu	taip	taip
Galimybė kviesti komandinės eilutės programas	taip	taip
Galimybė vykdyti SSH komandas nutolusiame serveryje	taip	Ne, bet galima naudoti SSH iš komandinės eilutės
Patogi integracija su SVN	taip	taip
Duomenų bazės versijavimo įrankis	dbdeploy įrankis nesunkiai integruojamas su Ant	dbdeploy įrankis sunkiau integruojamas su Phing, alternatyvų nelabai yra
Automatinis integravimo	taip	taip

rezultato raportavimas el. paštu		
----------------------------------	--	--

Skirtumas tarp šių įrankių yra labai nedidelis, tačiau CruiseControl+Ant įrankių rinkinys yra vystomas gerokai ilgiau, todėl ir buvo pasirinktas.

Įmonėje kuriamiems produktams naudojama PHP kalba, o šia kalba parašytoms programų sistemoms testuoti yra dvi galimos alternatyvos: PHPUnit ir SimpleTest. Renkantis įrankį buvo atsižvelgta į šiuos kriterijus:

- Lengva naudoti su CruiseControl+Ant – reikia kuo mažiau pastangų įdiegti.
- Naudojimo paprastumas.
- Patirtis dirbant su įrankiu – lengviau pradėti naudoti įrankį, su kuriuo darbuotojams yra tekę dirbti.

12 lentelė. Testavimo įrankių pasirinkimas.

Kriterijus	PHPUnit	SimpleTest
Lengva integracija	Per komandinę eilutę	Per komandinę eilutę
Naudojimo paprastumas	Abiejų įrankių naudojimo paprastumas panašus	Abiejų įrankių naudojimo paprastumas panašus
Patirtis dirbant su įrankiu	yra	nėra

Įrankiai yra labai panašūs. Kadangi su PHPUnit įmonės darbuotojai yra dirbę, pasirinktas šis įrankis.

6.4. Automatinė integracija ir publikavimas

Automatinė integracija ir publikavimas buvo įgyvendintas naudojant CruiseControl ir Ant įrankius. Papildomai teko suprogramuoti kelis scenarijus (angl. *script*) atliekančius produktui specifinius veiksmus:

- Vertimų konstantų sinchronizacija.
- Atsarginės atminties valymas.

Visi kiti veiksmai buvo įgyvendinti standartinėmis priemonėmis arba kviečiant įrankius iš komandinės eilutės.

Visas darbas užtruko apie 66 valandas vietoj planuotų 48. Ilgesnį laiką nei planuota galima paaiškinti tuo, kad toks darbas daromas pirmą kartą ir buvo sudėtinga įvertinti veiksmų įgyvendinimo sudėtingumą. Taip pat automatinis sukūrimas buvo baigtas savaite vėliau nei

planuota, kadangi šiai užduočiai buvo priskirtas žemesnis prioritetas.

Įgyvendinus automatinį testinės ir produkcinės versijos sukūrimą gauti šie rezultatai:

- Eliminuoja žmogiškos klaidos galimybę.
- Testavimo metu neberandama klaidų susijusių su integracija ir testinės versijos sukūrimu.
- Sutaupytas laikas, automatinis sukūrimas nereikalauja papildomų pastangų.
- Nenutrūkstama integracija užtikrina, kad integracijos problemos bus ištaisytos iš karto po to, kai jos atsiradusios.
- Techninis pagrindas pradėti daryti automatinį testavimą.
- Teigiamas vadovybės požiūris į gerinimo veiksmus ir didesnė parama tolimesniuose veiksmuose.

6.5. Bandomieji automatiniai testai

Į gerinimo planą įtraukta užduotis, kurios tikslas parašyti kelis bandomuosius integracinius ir programinių vienetų testus.

Parašyti keli bandomieji programinių vienetų testai, testuojantys verslo logikos objektus. Tokius testus parašyti buvo nesudėtinga, kadangi produktui naudojamas Zend karkasas yra gerai struktūrizuotas ir leidžia nesunkiai izoliuoti verslo logikos komponentus. Be to, jis pateikia įrankius patogiam programinių vienetų testavimui, pavyzdžiui, menamos duomenų bazės objektą, leidžiantį izoliuoti verslo logikos objektus nuo realios duomenų bazės ir tuo pačiu lengvai testuoti, kokius duomenų bazės pakeitimus vykdo verslo logikos objektai.

Bandomuosius integracinius testus įvykdyti buvo gerokai sunkiau. Remiantis judriosiomis testavimo praktikomis, šie testai turi veikti programų sistemos „API sąsajos lygyje“. Tai reiškia, kad testavimas vykdomas veikiančiai sistemai be vartotojo sąsajos. Kadangi testuojamas programų sistema iš pradžių nebuvo kuriama taip, kad būtų lengvai testuojama, be to, programų sistemoje buvo naudojami keli techniniai sprendimai, kurie nėra įprasti Zend karkasui, sunku visiškai atskirti vartotojo sąsają.

Remiantis judriojo testavimo praktikomis (testavimo automatizaciją pradėti nuo žemiausio lygio – programinių vienetų testų) ir bandomaisiais automatiniais testais, nuspręsta kitu gerinimo įgyvendinimo žingsniu:

- Užtikrinti automatinį programinių vienetų paleidimą ir rezultatų raportavimą įgyvendinti su jau esančiais automatinio integravimo įrankiais.
- Nustatyti reikiamą programų sistemos padengimą programinių vienetų testais.
- Užtikrinti reikiamą programų sistemos padengimą programinių vienetų testais.

naudojant padengimo metrikas ir užtikrinti automatinį tokių metrikų surinkimą.

- Įvykdyti reikalingus architektūrinius programų sistemos pakeitimus leidžiančius lengvai vykdyti automatinius integracinius testus.

6.6. Gerinimo patirtis

Įvykdžius pirmąją įmonės gerinimo ciklą galima, apibendrinti tokią patirtį:

1. Labai sunku įvertinti gerinimo technologinę riziką, kai tenka dirbti su įmonei naujomis technologijomis, todėl reikia lanksčiai žiūrėti į gerinimo planą, jį koreguoti, jei reikia.
2. Nepatartina planuoti viską padaryti iš karto, geriau planą išskaidyti į mažesnius žingsnius ir po kiekvieno žingsnio koreguoti veiksmų planą ir grafiką.
3. Pradėti nuo lengvai įgyvendinamų gerinimo veiksmų, kurie teikia lengvai pamatuojamą naudą, kadangi tai užtikrina vadovybės paramą tolimesniems gerinimo veiksams.

Rezultatai ir išvados

Programų kūrimo proceso gerinimas yra aktualus tiek didelėms, tiek labai mažoms įmonėms. Šiuo darbu buvo siekiama pateikti rekomendacijas labai mažų įmonių programų kūrimo proceso vertinimui ir gerinimui. Darbe remiamasi prielaida, kad net ir labai mažos įmonės gali turėti naudos vertindamos programų kūrimo procesą standarto atžvilgiu.

Darbo rezultatai:

1. Išnagrinėtas vertinimo metodų, paremtų ISO 15504 standartu ir skirtų mažoms įmonėms, pritaikymas labai mažose įmonėse.
2. Pasirinktas MARES metodas detalesnei analizei ir sudarytas metodo pakeitimų sąrašas, pritaikant metodą labai mažos įmonės programų kūrimo proceso vertinimui.
3. Modifikuotas metodas panaudotas labai mažos Lietuvos įmonės programų kūrimo proceso vertinimui.
4. Aprašytas patikslintas tikslų-procesų modelis, padedantis nustatyti vertinimo apimtį.
5. Remiantis metodo analize ir vertinimo patirtimi, pasiūlytos labai mažos įmonės proceso vertinimo rekomendacijos.
6. Po vertinimo pasiūlyti įmonės gerinimo sprendimai, atliktas pirmasis įmonės programų kūrimo proceso gerinimo ciklas remiantis judriosiomis praktikomis ir surinkta gerinimo patirtis.

Darbe išnagrinėti ISO 15504 standartu grindžiami vertinimo metodai skirti mažoms įmonėms ir analizuota, kaip jie gali būti pritaikyti labai mažoms įmonėms. Nustatyta, kad iš visų metodų skirtų mažoms įmonėms tinkamiausias labai mažos įmonės proceso vertinimui yra MARES metodas. Šis metodas turi apimties nustatymo procesą, yra laisvai prieinamas ir nemokamas, palyginus naujas, labiau struktūrizuotas ir todėl lengviau pritaikomas vidiniam vertinimui, vertinimo modelis lengviau plečiamas, tačiau MARES yra sudėtingesnis nei kiti metodai ir reikalingi pakeitimai, norint metodą pilnai pritaikyti labai mažos įmonės programų kūrimo proceso vertinimui.

MARES metodo pakeitimai daugiausiai susiję su tuo, kad tokį vertinimą atlieka įmonės darbuotojas, gerai žinantis įmonės procesus. Atsisakyta kelių veiklų, siekiant supaprastinti vertinimo metodą.

Pakeistas MARES metodas pritaikytas labai mažos Lietuvos informacinių technologijų įmonės programų kūrimo proceso vertinimui ir surinkta vertinimo patirtis. Taip parodyta, kad

vertinimo metodas tinkamas vertinti labai mažos įmonės procesą. Vertinimo metu buvo sudarytas procesų tikslinis profilis pagal nustatytus kriterijus. Tikslinio profilio sudarymo tikslų-procesų modelis galėtų būti naudojamas kitų labai mažų įmonių proceso vertinimui. Tiesa, darbe pateiktas modelis yra gana siauras ir buvo sudarytas vienos labai mažos įmonės vertinimo apimčiai nustatyti, tačiau modelį galima papildyti papildomais įmonių veiklos tikslais ir gebėjimo lygių kriterijais ir pritaikyti platesniam naudojimui.

Remiantis vertinimo rezultatais ir metodo analize, pasiūlytos labai mažos įmonės proceso vertinimo rekomendacijos, kurios gali būti naudojamos kuriant naujus metodus arba pritaikant esamus labai mažų įmonių programų kūrimo proceso vertinimui.

Po įmonės proceso vertinimo, remiantis nustatytomis silpnosiomis procesų pusėmis, atliktas įmonės proceso gerinimo ciklas ir surinkta gerinimo patirtis. Gerinimas atliktas remiantis judriosiomis praktikomis, taip įrodyta, kad judriųjų praktikų naudojimas galimas kartu su formaliu vertinimu. Parodyta, kad gerinimas, remiantis judriojo testavimo praktikomis leis pasirinktiems procesams pasiekti reikiamą lygį.

Šaltiniai

- [AWSS04] Alessandra Anacleto, Christiane Gresse von Wangenheim, Clenio F. Salviano, Rafael Savi, Experiences Gained from Applying ISO/IEC 15504 to Small Software Companies in Brazil, 4th International SPICE Conference on Process Assessment and Improvement, Portugal, 2004.
- [CG09] Lisa Crispin, Janet Gregory, Agile Testing, Addison-Wesley, 2009.
- [Coh06] Mike Cohn, Agile Estimating and Planning, Prentice Hall, 2006.
- [CS09] Gerhard Chroust, Erwin Schoitsch, Motivating SMEs to Software Process Improvement. A retrospective at the SPIRE-project of 1998, Shaker Verlag, 2009.
- [DM99] Tore Dyba, Nils Brede Moe, Rethinking the Concept of Software Process Assessment, EuroSPI99, 1999.
- [ESP99] ESPINODE for Central Italy, Toward Organised software Processes in SMEs, SME & SPI Questionnaire, 1999.
- [Eur05] European Commission, The new SME definition. User guide and model declaration, 2005. [žiūrėta 2008-06-10]. Prieiga per internetą: <http://ec.europa.eu/enterprise/enterprise_policy/sme_definition/sme_user_guide.pdf>
- [Gar05] Suz Garcia, Thoughts on Applying CMMI in Small Settings, Carnegie Mellon University, 2005.
- [GM97] Jennifer Gremba, Chuck Myers, The IDEAL(SM) Model: A Practical Guide for Improvement, Software Engineering Institute, 1997.
- [GT06] Suzanne Garcia, Richard Turner, CMMI Survival Guide: Just Enough Process Improvement, Addison-Wesley Professional, 2006.
- [Hof00] Douglas Hoffman, The Darker Side of Metrics, Software Quality Methods, 2000.
- [Hum89] Watts S. Humphrey, Managing the Software Process, Addison-Wesley, 1989.
- [Hum92] Watts S. Humphrey, Introduction to Software Process Improvement, Software Engineering Institute, Carnegie Mellon University, 1992, p. 5.
- [IB04] Informacinių technologijų sektoriaus studija, Profesinio mokymo metodikos centras, 2004. Prieiga per internetą: <<http://www.kpmc.lt/PMIT/doc/IT-%20studija.pdf>>.
- [ISO02] ISO 15504 Part 2: Performing an assessment, 2003.
- [ISO05] ISO 15504 Part 5: An exemplar Process Assessment Model, 2005.
- [Kau99] Karlheinz Kautz, Making Sense of Measurement for Small Organizations,

- Copenhagen Business School, 1999.
- [KH89] David H. Kitson, Watts S. Humphrey, The Role of Assessment in Software Process Improvement, Software Engineering Institute, Carnegie Mellon University, 1989.
- [KJ2008] Margaret K. Kulpa, Kent A. Johnson, Interpreting the CMMI. A Process Improvement Approach. Second Edition, CRC Press, 2008.
- [Kos03] Juha Koskela, Software configuration management in agile methods, VTT Publications, 2003.
- [Lar03] Craig Larman, Agile and Iterative Development: A Manager's Guide, Addison Wesley, 2003.
- [LF09] Giuseppe Lami, Fabio Falcini, Is ISO/IEC 15504 Applicable to Agile Methods?, Springer-Verlag Berlin Heidelberg, 2009
- [Loo04] Han van Loon. Process Assessment and ISO/IEC 15504. A reference Book, Springer, 2004, p. 187-190.
- [Loo07a] Han van Loon. Process Assessment and ISO/IEC 15504. A reference Book. Second Edition, Springer, 2007, p. 279-283, p. 17-22.
- [Loo07b] Han van Loon. Process Assessment and Improvement. A Practical Guide. Second Edition, Springer, 2007, p. 8, p. 11, p. 98.
- [NIST02] National Institute of Standards and Technology, The Economic Impacts of Inadequate Infrastructure for Software Testing, 2002.
- [PGP07] Francisco J. Pino, Felix Garcia, Mario Piattini, Software process improvement in small and medium enterprises: a systematic review, Springer Science.
- [RTCH00] T.P. Rout, A. Tuffley, B. Cahill and B. Hodgen, The Rapid Assessment of Software Process Capability, Software Quality Institute, Griffith University, Queensland, Australia, 2000.
- [RW07] Ita Richardson, Christiane Gresse von Wangenheim, Why Are Small Software Organizations Different?, IEEE Software, 2007-1, p. 18-20.
- [SB02] Ken Schwaber, Mike Beedle, Agile Software Development with Scrum, Prentice Hall, 2002, pp. 7-10.
- [SKJ10] M. Sivashankar, A. M. Kalpana, Dr. A. Ebenezer Jeyakumat, A framework approach using CMMI for SPI to Indian SME'S, Innovative Computing Technologies (ICICT), 2010 International Conference, 2010.
- [TSG94] The Standish Group, The CHAOS Report, 1994.
- [TSG00] The Standish Group, Extreme Chaos, 2000.
- [WAS04a] Christiane Gresse von Wangenheim, Alessandra Anacleto, Clenio F. Salviano,

MARES - A Method for Software Process Assessment in Small Software Companies, LPQS, 2004.

[WAS05] Chistiane Gresse von Wangenheim, Alessandra Anacleto, Clenio F. Salviano, Rafael Savi, Documentacao do Metodo de Avaliacao de Pprocesso de Software MARES/11504MPE v1.0, LPQS, 2005.

Priedai

1 priedas. MARES vertinimo planavimo procesas

PROC.1.1.1 Pradinis susitikimas, kurio metu:

- 1) Vertintojas susitinka su vertinimo rėmėju, kurio metu pristatomas būsimas darbas.
- 2) Vertintojas ir vertinimo rėmėjas pasirašo paslaugos sutartį. Šis dokumentas aprašo darbą, kurį turės padaryti vertintojas bei vertinamos kompanijos teises ir pareigas.
- 3) Pasirašoma konfidencialumo sutartis tarp vertintojo ir vertinimo rėmėjo. Ši sutartis užtikrina, kad informacija, surinkta vertinimo metu nebus atskleista be vertinimo rėmėjo sutikimo.
- 4) Nustatoma vertinimo pabaigos data remiantis pirminiu planavimu.
- 5) Nurodyti vertinamos įmonės darbuotojus, kurie dalyvaus konteksto sudarymo procese.

PROC.1.1.2 Kontekstualizacijos planavimas, kurio metu:

- 1) Vertintojas pasirenka vertinimo komandos narius.
- 2) Sudaromas kontekstualizacijos planas remiantis pirminiu planu.
- 3) Kontekstualizacijos planas dokumentuojamas.

PROC.1.1.3 Kontekstualizacijos plano peržiūra, kurio metu paskirtas žmogus peržiūri kontekstualizacijos plano nuoseklumą ir tinkamumą.

PROC.1.1.4 Vertinimo vykdymo planavimas, kurio metu:

- 1) Vertintojas išrenka žmones, kurie dalyvaus vertinime.
- 2) Remiantis pirminiu planu sudaromas vertinimo grafikas.
- 3) Vertinimo planas dokumentuojamas.

Vertinimo vykdymo planavimas atliekamas po kontekstualizacijos.

PROC.1.1.5 Vertinimo vykdymo plano peržiūra, kurios metu paskirtas žmogus peržiūri vertinimo vykdymo planą įvertindamas jo nuoseklumą tikslumą ir informacijos tinkamumą. Siekiant įvertinti šiuos kriterijus naudojamas pirminis planas ir kontekstualizacijos rezultatai.

Šis planavimo procesas naudojamas, kai įmonės vertinimą atlieka įmonė ar grupė iš šalies.

2 priedas. MARES kontekstualizacijos procesas

PROC.2.1.1 Charakterizacijos klausimyno pildymas, kurio metu:

- 1) Charakterizacijos klausimynas siunčiamas rėmėjui.
- 2) Rėmėjas užpildo klausimyną.
- 3) Vertintojas turi užtikrinti, kad klausimynas gautas laiku, laikantis kontekstualizacijos plano.

PROC.2.1.2 Charakterizacijos klausimyno analizė, kurios metu:

- 1) Vertintojas patikrina ar visi klausimyno laukai užpildyti.
- 2) Kilus klausimams, vertintojas susisiekia su klausimyną užpildžiusiu asmeniu.
- 3) Vertintojas išsiunčia klausimyną visiems vertinime dalyvaujantiems asmenims.
- 4) Klausimai, susiję su charakterizacija išsprendžiami kontekstualizacijos interviu metu.

PROC.2.1.3 Kontekstualizacijos interviu, kurio metu:

- 1) Vykdomas vertinimo pristatymas.
- 2) Nustatomi įmonės tikslai, užpildoma kontekstualizacijos forma.
- 3) Išsprendžiami klausimai, kilę charakterizacijos metu.

PROC.2.2.1 Įmonės proceso aprašymas, kurio metu:

- 1) „Aukštu lygiu“ aprašomi įmonėje vykdomi procesai.

PROC.2.2.2 Tikslinio profilio nustatymas, kurio metu:

- 1) Remiantis kontekstualizacijos klausimynu nustatomi aktualiausi procesai.
- 2) Nustatomas reikiamas procesų lygis.

3 priedas. MARES vertinimo procesas

PROC.3.1.1 Pasiruošimas vertinimui, kurio metu:

- 1) Vertintojas surašo vertinamų procesų atributų indikatorius į vertinimo dokumentą.
- 2) Visi vertinimo dalyviai turi išstudijuoti atributų indikatorius

PROC.3.2.1 Duomenų rinkimas, kurio metu:

- 1) Įmonės darbuotojai informuojami, kad bus vykdomi interviu
- 2) Vertintojas, besiremdamas informacija surinkta pasiruošimo metu, vykdo neformalius interviu su įmonės darbuotojais.

- 1) Vertinime dalyvaujantys įmonės darbuotojai interviu metu pristato įmonėje vykdomas veiklas.
- 2) Vertintojas dokumentuoja interviu rezultatus

PROC.3.2.2 Duomenų konsolidacija, kurios metu:

- 1) PROC.3.2.1 proceso metu surinkti duomenys priskiriami bazinėms/bendrosioms praktikoms.
- 2) Surinkti duomenys peržiūrimi ir sudaromas klausimynas įmonės darbuotojams dėl duomenų patikslinimo.

PROC.3.2.3 Duomenų validavimas, kurio metu:

- 1) Duomenų patikslinimo klausimai pateikiami įmonės darbuotojams, atsakymai dokumentuojami
- 2) Įmonės darbuotojai peržiūri surinktus duomenis ir nurodo, ko trūksta ar buvo suprasta neteisingai.
- 3) Darbuotojų pastebėjimai dokumentuojami.

PROC.3.3.1 Procesų gebėjimų lygių nustatymas, kurio metu:

- 1) Nustatoma ar kiekviena įmonėje vykdoma veikla padeda pasiekti bazinę/bendrąją praktiką.
- 2) Remiantis bazinėmis/bendrosiomis praktikomis proceso atributai įvertinami pagal NLPF skalę.
- 3) Įvertinami procesų lygiai: kad, procesas pasiekia nustatytą lygį, jei lygio atributai pasiekia bent L įvertinimą, o žemesnių lygių atributai pasiekia F įvertinimą.

PROC.3.3.2 Galutinis įvertinimas, kurio metu:

- 1) Remiantis skirtumu tarp tikslinio ir faktinio procesų gebėjimo lygių profilio, nustatomos gerintinos vietos.
- 2) Dokumentuojamos stipriosios ir silpnosios proceso vietos.

PROC.3.4.1 Ataskaitos ruošimas, kurios metu:

- 1) Dokumentuojamas galutinis įvertinimas, gerintinos vietos, silpnosios/stipriosios procesų pusės.

PROC.3.4.2 Patikrinti vertinimo suderinamumą su standartu, kurio metu

- 1) Vertintojas turi nustatyti vertinimo atitikimą ISO 15504 standarto reikalavimams pagal atitikimo matricą.

PROC.3.4.3 Ataskaitos peržiūra, kurios metu:

- 1) Vertinimo ataskaita išsiunčiama žmogui atsakingam už peržiūras.
- 2) Ataskaita peržiūrima tikrinant ar joje pateikta informacija korektiška ir nuosekli.
- 3) Peržiūrėta ataskaita grąžinama vertintojui.
- 4) Vertintojas įtraukia pakitimus į ataskaitą.

PROC.3.4.4 Rezultatų pristatymas, kurio metu:

- 1) Vertinimo ataskaita pristatoma darbuotojams ir vadovybei susitikimo metu.
- 2) Diskutuojama dėl gerinimo variantų
- 3) Vertinimo ataskaita perduodama vertinimo rėmėjui.

4 priedas. MARES tikslų-procesų modelis

Įmonės tikslai	Susieti procesai
Plėsti rinką	OPE.2 – klientų palaikymas ENG.12 – programinės įrangos ir sistemos priežiūra REU.3 – dalykinės srities inžinerija
Patenkinti klientų poreikius	SPL.1 – tiekėjo pasiūlymai CFG.4 – pakeitimų prašymų valdymas MAN.3 projektų valdymas MAN.5 rizikų valdymas
Sumažinti kaštus	MAN.3 projektų valdymas Bet kuris kitas procesas turintis didelę įtaką kainai
Padidinti produktyvumą	MAN.3 projektų valdymas Bet kuris kitas procesas turintis didelę įtaką produktyvumui
Atitikti nustatytą biudžetą	MAN.3 projektų valdymas SUP.1 – kokybės užtikrinimas MAN.4 – kokybės valdymas ENG.1 – reikalavimų išaiškinimas OPE.2 – klientų palaikymas CFG.4 – pakeitimų prašymų valdymas
Geriau atitikti klientų poreikius	ENG.1 – reikalavimų išaiškinimas ENG.4 – reikalavimų analizė SUP.1 – kokybės užtikrinimas MAN.4 – kokybės valdymas
Sutrumpinti laiką tarp išleidimų	MAN.3 – projektų valdymas Bet kuris kitas procesas turintis didelę įtaką išleidimų laikui
Suspėti laiku baigti projektus	SPL.1 – tiekėjo pasiūlymai SUP.10 – pakeitimų prašymų valdymas MAN.3 – projektų valdymas MAN.5 – rizikų valdymas
Kokybės aspektai	
Funkcionalumas	ENG.1 – reikalavimų išaiškinimas ENG.4 – reikalavimų analizė ENG.8 – testavimas SUP.1 – kokybės užtikrinimas MAN.4 – kokybės valdymas
Naudojamumas	ENG.1 – reikalavimų išaiškinimas ENG.4 – reikalavimų analizė

	<p>ENG.8 – testavimas</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p>
Patikimumas	<p>ENG.1 – reikalavimų išaiškinimas</p> <p>ENG.4 – reikalavimų analizė</p> <p>ENG.8 – testavimas</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p> <p>Bet kuris kitas procesas, kuriame atsiranda daug klaidų</p>
Efektyvumas	<p>ENG.5 – programinės įrangos projektavimas</p> <p>ENG.6 – programinės įrangos gamyba</p> <p>ENG.8 – testavimas</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p>
Prižiūrimumas	<p>ENG.5 – programinės įrangos projektavimas</p> <p>ENG.6 – programinės įrangos gamyba</p> <p>ENG.12 – programinės įrangos priežiūra</p> <p>SUP.7 – dokumentavimas</p> <p>SUP.8 – konfigūracijos valdymas</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p>
Portabilumas	<p>ENG.5 – programinės įrangos projektavimas</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p>
Pakartotinis panaudojamumas	<p>ENG.5 – programinės įrangos projektavimas</p> <p>ENG.6 – programinės įrangos gamyba</p> <p>SUP.1 – kokybės užtikrinimas</p> <p>MAN.4 – kokybės valdymas</p> <p>REU.1 – inventoriaus valdymas</p> <p>REU.2 – pakartotinio panaudojimo programos valdymas</p> <p>REU.3 – dalykinės srities inžinerija</p>

5 priedas. Detalus gerinimo planas

	Veiksmas	Gerinimo gairės	Pabaigos data	Laikas(val.)
1	Pasirinkti testavimo įrankius programinių vienetų ir integraciniam testavimui	2.1 skyrius, 4 a,b, c, e dalys. 2.2 skyrius, 4 a,b, c dalys.	2010-03-01	8
2	Sudaryti testavimo procedūrą vykdomą kiekvieno projekto metu ir reikalingus testavimo matavimus	2.1 skyrius, 1a dalis. 2.2 skyrius, 1a dalis.	2010-03-07	24
3	Aptarti ir suderinti testavimo procedūrą ir matavimus įmonės susitikimo metu	2.1 skyrius, 1b dalis. 2.2 skyrius, 1b dalis.	2010-03-10	2
4	Įdiegti pasirinktus įrankius	2.1 skyrius, 4 dalis. 2.2 skyrius, 4 dalis.	2010-03-20	16
5	Susieti naujus įrankius su dabar naudojamais įrankiais (pagrindė su SVN)	2.1 skyrius, 4d dalis. 2.2 skyrius, 4a dalis.	2010-03-22	8
6	Suderinti su programuotojais techninius pakeitimus, reikalingus programinių vienetų ir integracinių testų vykdymui	2.1 skyrius, 3 dalis. 2.2 skyrius, 3 dalis.	2010-03-25	2
7	Padaryti architektūrinius produkto pakeitimus	2.1 skyrius, 3 dalis. 2.2 skyrius, 3 dalis.	2010-03-28	16
8	Sukonfiguruoti pasirinktus įrankius taip, kad būtų galima automatiškai sukurti testinę ir produkcinę produkto versijas	2.2 skyrius, 4a dalis.	2010-03-28	24
9	Parašyti kelis bandomuosius programinių vienetų ir integracinius testus ir užtikrinti automatinį jų vykdymą	2.1 skyrius, 4g dalis. 2.2 skyrius, 4d dalis.	2010-04-04	16
10	Sudaryti reikalingų matavimų sąrašą ir juos dokumentuoti	2.1 skyrius, 2 dalis. 2.2 skyrius, 2 dalis.	2010-04-11	8
11	Automatizuoti reikalingus matavimus (kiek įmanoma)	2.1 skyrius, 5 dalis. 2.2 skyrius, 5 dalis.	2010-04-18	32
12	Užtikrinti reikalingų neautomatizuotų matavimų vykdymą	2.1 skyrius, 5 dalis. 2.2 skyrius, 5 dalis.	2010-04-22	8
13	Įtvirtinti pagerinimą – padaryti gerinimo susitikimą, kurio metu su darbuotojais bus sutariama, kaip nuo šiol bus dirbama.	2.1 skyrius, 6 dalis. 2.2 skyrius, 6 dalis.	2010-05-01	2
14	Stebėti vykdomas veiklas.	2.1 skyrius, 6 dalis. 2.2 skyrius, 6 dalis.	Neribotai	-

6 Priedas. Programinių vienetų testavimo automatizavimas

Kadangi programinių vienetų testavimas užtikrina programuotojų rašomo kodo kokybę ir stipriai sumažina klaidų tikimybę vėlesniuose programinės įrangos kūrimo etapuose, reiktų pradėti nuo šių testų naudojimo. Šie testai taip pat užtikrina, kad klaida bus surasta ir ištaisyta labai greitai po to, kai tik atsiranda. Šių testų automatizavimas užtikrina, kad bus vykdomas greitas regresinis testavimas programinių vienetų lygyje. Be to, automatizavimas padeda griežtai formalizuoti šį testavimą bei lengviau rinkti reikalingas metrikas.

Programinių vienetų testavimo automatizavimas bus įgyvendintas atlikus šiuos veiksmus:

1. Sudaryti testavimo kontrolinį sąrašą, kuris bus naudojamas užtikrinti, kad bus įvykdytos reikalingos testavimo užduotys.
 - a) Į kontrolinį sąrašą reikia įtraukti šias užduotis:
 1. Nuspręsti kokios programinės įrangos dalys bus testuojamos.
 2. Nuspręsti kaip bus testuojama.
 3. Nuspręsti kokio detalumo testai bus reikalingi.
 4. Nuspręsti kokių programinės įrangos gyvavimo ciklo metu reikės vykdyti šį testavimą. Testai gali būti rašomi prieš programavimą, po programavimo, testavimo fazėje. Testai gali būti vykdomi iš karto po programavimo, integravimo metu, testavimo fazės metu.
 5. Nuspręsti kaip bus vykdomas regresinis testavimas.
 6. Nuspręsti kokie testai bus įtraukiami į regresinį testavimą. Skirtingos programinės įrangos dalys gali pareikalauti skirtingų technologinių testavimo įgyvendinimų.
 7. Paskirstyti atsakomybės, kas kurs programinių vienetų testus.
 8. Nuspręsti, kaip bus valdomos testavimo užduotys.
 9. Nuspręsti, kaip bus valdomi testavimo proceso produktai.
 - b) Testavimo kontrolinis sąrašas turi būti aptartas ir suderintas su vadovybe ir darbuotojais, kurie bus atsakingi už jos vykdymą.
2. Numatyti matavimus:
 - a) Reikalingus įvertinti programinių vienetų testavimo teikiamą naudą,
 - b) Nustatyti testavimo efektyvumą
 - c) Testavimo pabaigos sąlygas.
 - d) Kiekvieno matavimo naudojimą reikia motyvuoti ir motyvaciją dokumentuoti, kad nebūtų pamiršta, kam jis naudojamas.
3. Padaryti architektūrinius testuojamos programinės įrangos pakeitimus, kad būtų galima įgyvendinti programinių vienetų testus.
 - a) Paprastai, įgyvendinant automatinį programinių vienetų testavimą jau sukurtai programinei įrangai, reikia atlikti testuojamos programinės įrangos pakeitimus, kad toks testavimas veiktų. Norint įgyvendinti tokį testavimą, reikia atskirti architektūrinius programinės įrangos architektūrinius elementus (pavyzdžiui, verslo logiką nuo atvaizdavimo). Jei reikia labai daug tokių pakeitimų, galima juos įgyvendinti palaipsniui vystant programinę įrangą.
 - b) Įgyvendinant šį testavimą, reikia nuspręsti, ar visi reikalingi testai bus parašyti pradžioje ar bus rašomi vystant produktą. Taip pat galimas kompromisinis variantas: pradžioje parašyti testus tik

kritiškiausios dalims ir toliau rašyti išsamesnius testus vystant programinę įrangą.

4. Sukurti reikalingą technologinę automatinio testavimo infrastruktūrą:
 - a) Reikės įvertinti reikalingus įrankius ir pasirinkti iš kelių galimų trečios šalies įrankių arba kurti savus įrankius, tačiau mažai įmonei tą daryti pareikalautų per daug resursų, todėl nuspręsta kiek įmanoma naudoti trečios šalies įrankius net jei jie pilnai neatitinka įmonės poreikių.
 - b) Reikalavimai įrankiams:
 1. Gali automatizuoti kiek įmanoma daugiau testavimo ir reikalauja kuo mažiau rankinio darbo pasikartojančioms užduotims.
 2. Tarpusavyje suderinami, pavyzdžiui, nenutrūkstamos integracijos serveris turėtų mokėti dirbti su automatinio integravimo (angl. *build*) įrankiais, automatinio testavimo įrankiais, išėtinio kodo kontrolės sistema.
 3. Būtų galima naudoti kituose projektuose, ne tik gerinamame projekte.
 - c) Reikia paruošti techninę infrastruktūrą, pavyzdžiui, reikalingus serverius.
 - d) Reikia parašyti ir įvykdyti kelis bandomuosius programinių vienetų testus, siekiant išbandyti, kaip veikia įdiegta sistema.
5. Matavimų įgyvendinimas. Visus planuojamus matavimus reiktų kiek įmanoma automatizuoti ir tam galima panaudoti trečių šalių įrankius, pavyzdžiui galima rasti įrankius, kurie gali pamatuoti, kiek programų sistemos kodo padengia programinių vienetų testai.
6. Pagerinimo įtvirtinimas. Reikia pristatyti darbuotojams naujus įrankius ir procedūras, juos apmokyti naudotis naujais įrankiais ir procedūromis. Reikia padaryti įtvirtinimo susitikimą su visais darbuotojais ir suderinti, kaip bus dirbama. Po to reikia užtikrinti, kad būtų laikomasi nustatytų programinių vienetų testavimo procedūrų. Kadangi visi darbuotojai dirba vienoje aplinkoje ir nėra atskiri vienas nuo kito, tai padaryti nėra sudėtinga. Tiesiog dirbant matoma, ar vykdomos nustatytos veiklos, ir galima atlikti korekcijas žodiniu susitarimu.

7 priedas. Integracinio testavimo automatizavimas

Pagal judrųjį testavimą, integracinis testavimas pradedamas dar sudarant reikalavimus programų sistemai. Reikalavimai suformuluojami testų pavidalu ir programinės įrangos gamybos metu šie testai padeda kūrėjams užtikrinti, kad gaminama programų sistema atitinka kliento reikalavimus. Šie testai gali būti sudaromi klientui ir programų sistemos kūrėjui suprantama kalba, pavyzdžiui, sudaromos lentelės su įvesties ir išvesties duomenimis arba naudojama formali kalba (labai panaši į natūralią kalbą) aprašyti programų sistemos elgseną. Tam tikrais atvejais (klientas yra su IT išsilavinimu, kai gaminamas produktas yra skirtas rinkai) testai gali būti rašomi programavimo kalba. Integraciniai testai vykdomi veikiančiai programų sistemai, tuo tarpu programinių vienetų testų vykdymui nebūtina paleisti programų sistemos.

Automatinio integracinio testavimo įgyvendinimo planas iš esmės labai panašus į programinių vienetų testavimo automatizavimo planą:

1. Sudaryti testavimo kontrolinį sąrašą, užtikrinantį testavimo veiklų vykdymą.
 - a) Į kontrolinį sąrašą reikia įtraukti šias užduotis:
 1. Nuspręsti, kokios programų sistemos dalys bus testuojamos.
 2. Nuspręsti, kaip bus vykdomas testavimas.
 3. Nuspręsti, kada bus vykdomas integracinis testavimas.
 4. Reikės numatyti, kokie testai bus įtraukiami į regresinį testavimą.
 5. Nuspręsti, kas kurs integracinius testus.
 6. Nuspręsti, kaip bus valdomos testavimo užduotys.
 7. Nuspręsti, kaip bus valdomi testavimo proceso produktai.
 - b) Kontrolinis sąrašas susitikimo metu turi būti suderintas su vadovybe ir darbuotojais, kurie bus atsakingi už jos vykdymą.
2. Numatyti reikalingus matavimus. Integracinis testavimas judriajame testavime yra skirtas nustatyti, kiek produktas atitinka klientų poreikius ir reikalavimus. Integracinio testavimo matavimai būtent tai ir turėtų parodyti.
3. Jei reikia, padaryti architektūrinius programų sistemos pakeitimus, kurie užtikrintų, jog bus galima vykdyti automatinius integracinius testus. Gali būti, kad po pakeitimų, kurių reikalavo programinių vienetų testavimas, integraciniams testavimui skirtų pakeitimų net nereikės.
4. Sukurti integracinio testavimo infrastruktūrą. Tam reikės panaudoti infrastruktūrą, sukurtą programinių vienetų testavimui, ją papildyti įrankiais, skirtais integraciniam testavimui:
 - a) Automatiškai paimančiais naujus pakeitimus
 - b) Sukuriančiais veikiančią testinę programų sistemą su naujausiais pakeitimais.
 - c) Galinčiais šiai testinei sistemai vykdyti integracinius testus.
 - d) Parašyti ir įvykdyti kelis bandomuosius integracinius testus, siekiant išbandyti kaip veikia įdiegta sistema.
5. Įgyvendinti matavimus kiek įmanoma juos automatizuojant.
6. Pagerinimo įtvirtinimas – daromas taip pat kaip ir programinių vienetų testavimo.

8 priedas. Programinės įrangos gamybos padengimas po pagerinimo

Praktika	Dabar yra	Bus po pagerinimo
ENG.6.BP.1 Sukurti verifikavimo procedūrą	Nėra	Prieš programuojant programinius vienetus bus sukuriami programinių vienetų testai
ENG.6.BP.2 Sukurti programinius vienetus	Programiniai vienetai	Pakeitimų nebus
ENG.6.BP.3 Užtikrinti vientisumą	Nėra	Numatytas procesas nereikalauja projektavimo kaip atskiros fazės, tačiau užtikrina, kad programiniai vienetai tiesiogiai atitiks numatytus reikalavimus, o architektūros pakeitimas (jei keičiasi reikalavimai ar numatyta architektūra yra netinkama) supaprastinamas automatinių testų, kurie užtikrina, kad programų sistema net po esminių architektūrinių pakeitimų veiks
ENG.6.BP.4 Verifikuoti programinius vienetus	Nėra	Programinių vienetų testai vykdomi gamybos metu ir visada įvykdomi automatiškai surinkimo (<i>angl. build</i>) metu
GP.2.1.1 Identifikuoti procesų vykdymo tikslus	Nėra	Bus įvardinta testavimo procedūroje
GP.2.1.2 Planuoti ir stebėti procesą	Vykdoma projektų valdymo procese	Pakeitimų nebus
GP.2.1.3 Koreguoti proceso vykdymą	Vykdoma projektų valdymo procese	Pakeitimų nebus
GP.2.1.4 Nurodyti atsakomybes	Nėra	Bus įvardinta testavimo procedūroje
GP.2.1.5 Suteikti reikalingus resursus	Nėra	Testavimo procedūroje turės būti įvardinta, kaip resursai bus suteikiami
GP.2.1.6 Valdyti sąryšius tarp suinteresuotų šalių	Vykdoma projektų valdymo procese	Pakeitimų nebus
GP.2.2.1 Nurodyti reikalavimus darbo produktams	Nėra	Testavimo procedūroje bus nurodyta, kaip suformuluojami reikalavimai darbo produktams
GP.2.2.2 Nurodyti reikalavimus produktų	Nėra	Testai nebus dokumentuojami bet jie bus

Praktika	Dabar yra	Bus po pagerinimo
dokumentavimui ir valdymui		valdomi konfigūracijos valdymo procese
GP.2.2.3 Identifikuoti, dokumentuoti ir valdyti darbo produktus	Nėra	Testai valdomi konfigūracijos valdymo procese
GP.2.2.4 Peržiūrėti ir koreguoti darbo produktus	Nėra	Testavimo procedūroje bus nurodyti bendri reikalavimai programinių vienetų testams, tačiau atskirai atitikimas reikalavimams nebus tikrinamas

9 priedas. Programinės įrangos integravimo padengimas po pagerinimo

Praktika	Dabar yra	Bus po pagerinimo
ENG.7.BP.1 Sukurti programinės įrangos integravimo strategiją	Nėra	Testavimo procedūroje nurodyta, kad reikia numatyti, kaip bus integruojamas produktas naudojantis integravimo įrankiais
ENG.7.BP.2 Sukurti testus integruotai programinei įrangai	Nėra	Prieš programuojant bus sukuriami integraciniai testai, atitinkantys reikalavimus
ENG.7.BP.3 Integruoti programinę įrangą	Integruojama rankiniu būdu	Tai bus daroma automatiškai
ENG.7.BP.4 Ištestuoti integruotą programinę įrangą	Testavimas nereguliarus	Testai bus vykdomi po kiekvieno integravimo
ENG.7.BP.5 Užtikrinti vientisumą	Nėra	Numatytas procesas nereikalauja projektavimo kaip atskiros fazės, tačiau užtikrina, kad programinė įranga tiesiogiai atitiks numatytus reikalavimus, o architektūros pakeitimas (jei keičiasi reikalavimai ar numatyta architektūra yra netinkama) supaprastinamas automatinių testų, kurie užtikrina, kad programų sistema net po esminių architektūrinių pakeitimų veiks
ENG.7.BP.6 Vykdyti regresinį testavimą	Nėra	Integraciniai testai bus vykdomi po kiekvieno integravimo
GP.2.1.1 Identifikuoti procesų vykdymo tikslus	Nėra	Bus nurodyta testavimo procedūroje
GP.2.1.2 Planuoti ir stebėti procesą	Vykdoma projektų valdymo procese	Pakeitimų nebus
GP.2.1.3 Koreguoti proceso vykdymą	Vykdoma projektų valdymo procese	Pakeitimų nebus
GP.2.1.4 Nurodyti atsakomybes	Nėra	Bus nurodyta testavimo procedūroje
GP.2.1.5 Suteikti reikalingus resursus	Nėra	Testavimo procedūroje turės būti įvardinta, kaip resursai bus suteikiami
GP.2.1.6 Valdyti sąryšius tarp suinteresuotų šalių	Vykdoma projektų valdymo procese	Pakeitimų nebus

Praktika	Dabar yra	Bus po pagerinimo
GP.2.2.1 Nurodyti reikalavimus darbo produktams	Nėra	Testavimo procedūroje bus nurodyta, kaip suformuluojami reikalavimai darbo produktams
GP.2.2.2 Nurodyti reikalavimus produktų dokumentavimui ir valdymui	Nėra	Testai nebus dokumentuojami bet jie bus valdomi konfigūracijos valdymo procese
GP.2.2.3 Identifikuoti, dokumentuoti ir valdyti darbo produktus	Nėra	Testai valdomi konfigūracijos valdymo procese
GP.2.2.4 Peržiūrėti ir koreguoti darbo produktus	Nėra	Testavimo procedūroje bus nurodyti bendri reikalavimai integraciniams testams, tačiau atskirai atitikimas reikalavimams nebus tikrinamas