

Tarpinio programų kūrimo proceso modelio reikalavimai

Stasys Peldžius

Vilniaus universiteto Matematikos ir informatikos fakulteto doktorantas
 Vilnius University, Faculty of Mathematics and Informatics, PhD student
 Naugarduko g. 24, LT-03225 Vilnius
 Tel. +370 5 219 30 64
 El. paštas: stasys.peldzius@mif.vu.lt

Saulius Ragaišis

Vilniaus universiteto Matematikos ir informatikos fakulteto docentas, daktaras
 Vilnius University, Faculty of Mathematics and Informatics, Assoc. Professor, PhD
 Naugarduko g. 24, LT-03225 Vilnius
 Tel. +370 5 219 30 75
 El. paštas: saulius.ragaisis@mif.vu.lt

Įmonės, siekdamos sėkmingai įgyvendinti programų kūrimo projektus – pateikti rezultatus laiku, neviršydamos biudžeto, ir kad klientai būtų patenkinti jų kokybe, renkasi programų kūrimo proceso modelius, pagal kuriuos gali įvertinti savo programų kūrimo procesą ir jį gerinti. Šiuo metu pasaulyje populiariausi programų kūrimo proceso modeliai yra ISO/IEC 15504 ir CMMI. Įmonėms prasminga turėti įvertinimus pagal abu šiuos proceso modelius, bet kiekvienas įmonės vertinimas brangus tiek finansiniu, tiek laiko požiūriu, todėl būtų naudinga gauti įvertinimus pagal skirtingus modelius, atliekant tik vieną vertinimą. Norėdamos apibrėžti ir (ar) pagerinti savo programų kūrimo procesą, įmonės renkasi įvairius programų sistemų inžinerijos metodus, pavyzdžiui, XP, Scrum, DSDM, RUP. Įmonei svarbu žinoti, ką pasirinktas metodas jai gali duoti: kokį proceso gebėjimą ir (ar) brandą jis gali užtikrinti. Šias aktualias problemas išspręstų tarpinis programų kūrimo proceso modelis, kuris leistų įmonems atlikti įvertinimą pagal vieną proceso modelį ir gauti rezultatus pagal kelis modelius, taip pat žinoti, kokį proceso gebėjimą ir (ar) brandą gali užtikrinti pasirinktas programų sistemų inžinerijos metodas. Straipsnyje pateikiamas pirmasis tokio modelio kūrimo žingsnis – apibrėžiami tokio modelio reikalavimai.

Dauguma programinę įrangą kuriančių įmonių susiduria su problemomis: projektai vėluoja, viršijamas biudžetas, klientai nepatenkinti produktų kokybe, o sėkmingi projektai sudaro tik nedidelę dalį. Šis reiškinys yra toks paplitęs, kad net pavadintas programinės įrangos krize (Dijkstra, 1972). Dažniausiai projektai būna sunkiai prognozuojami, vykdomi chaotiškai, nėra nusistovėjusio programų kūrimo proceso – kiekvienas projektas vykdomas vis kitaip. Projektų sėkmė tiesiogiai priklauso nuo įmonės darbuotojų patirties ir herojiškų pastangų. Tokios įmonės dažniausiai pajėgios sėkmingai įgyvendinti tik labai panašius, tipinius projektus, be to, jos labai priklausomos nuo savo darbuotojų – išėjęs žmogui iš darbo, dažniausiai dingsta visa sukaupta patirtis ir žinios apie projektą. Visa tai

reikalauja papildomų tiek pačios įmonės, tiek užsakovo išteklių. Įmonės tampa nebe pajėgios konkuruoti rinkoje.

Daugelis problemų kyla dėl prasto programų kūrimo proceso, pagal kurį įmonė įgyvendina projektus. Net geriausi įrankiai ir metodai neduoda pageidaujamo rezultato, kai jie naudojami chaotiškai, o kai kurios būtinos projekto veiklos visai neatliekamos.

Programų kūrimo proceso modelių atsiradimą paskatino užsakovų poreikiai turėti objektyvius kriterijus tinkamam projekto vykdytojui pasirinkti. Taip pat jau nuo pat modelių kūrimo pradžios ne mažiau svarbus tikslas buvo jų tinkamumas procesui gerinti. Proceso modelis priklausomai nuo architektūros leidžia įvertinti įmonės visuminio programų proceso brandą – nustatyti,

kiek įmonės procesas yra apibrėžtas, valdomas, matuojamas bei nuolatos gerinamas – arba atskirų procesų (pvz., konfigūracijos valdymo) gebėjimą. Kuo proceso branda / gebėjimas didesnis, tuo įmonės projektuose mažiau aptinkama defektų, smarkiai mažėja darbo sąnaudos, tiksliau prognozuojami projekto terminai ir biudžetas.

Šiame straipsnyje nagrinėjamos proceso modelių taikymo problemos, pristatomi susiję moksliniai straipsniai, kurie bando jas spręsti. Iškeliama idėja sukurti tarpinį proceso modelį, kuris turėtų padėti įmonėms spręsti su proceso modeliais susijusias problemas. Pateikiami tarpinio proceso modelio reikalavimai.

Esminės sąvokos

Programų kūrimo proceso modeliai vystėsi skirtingai, pavyzdžiui, CMMI šeima prasidėjo nuo pakopinio CMM, o ISO/IEC 15504 pirm-takas SPICE buvo tolydusis, todėl susiformavo skirtinga terminija. Siekiant vienareikšmiškumo, toliau pateikiami esminių sąvokų apibrėžimai.

Programų kūrimo procesu vadinama visuma veiklų, kurios vykdomos kuriant programinį produktą ar paslaugą. Šis procesas dar vadinamas visuminiu procesu. CMMI programų kūrimo procesas vadinamas tiesiog procesu, o ISO/IEC 15504 atitinkamos sąvokos apskritai nėra.

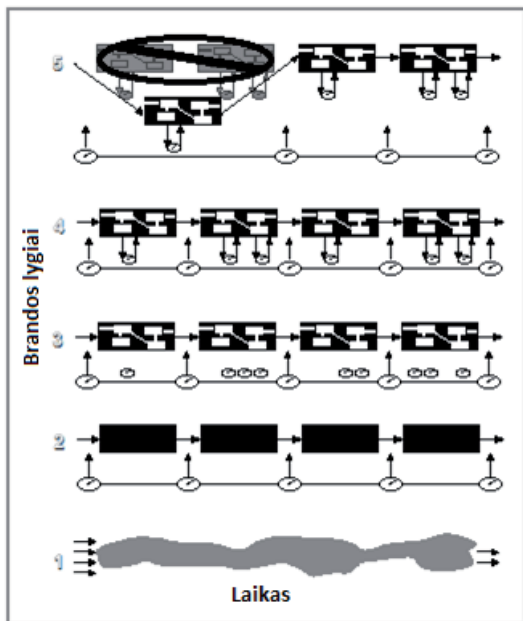
Vardinis procesu vadinamas rinkinys veiklų, susijusių pagal tikslus programinio produkto ar paslaugos gyvavimo cikle. Vardiniam procesams keliami tikslai ir aprašomi jų rezultatai. Įvairiuose proceso modeliuose vardinis procesas yra skirtingai vadinamas. Pavyzdžiui, ISO/IEC 15504 jis vadinamas tiesiog procesu, o CMMI modelyje vardinio proceso atitikmuo yra proceso sritis. CMMI proceso sritis apibrėžiama kaip tos srities geriausių praktikų rinkinys, kurį įgyvendinus pasiekiami tos proceso srities išskelti tikslai, kurie padaro reikšmingą įtaką jos gerinimui (CMMI-DEV, 2010).

Proceso modelis yra programų kūrimo proceso apibrėžimas, kuriame nusakyta proceso apimtis, aprašyti vardiniai procesai, jų tikslai ir rezultatai. Proceso modelių yra daug ir įvairių. Įmonė turi nuspręsti, kurį modelį pasirinkti savo procesui vertinti ir gerinti.

Gebėjimas yra proceso charakteristika, nusakanti rezultatų, kuriuos galima gauti taikant tą procesą, pasiskirstymą (diapazoną), t. y. galimybę (tikimybę), kad procesas pasieks jam keliamus tikslus. Tradiciškai gebėjimo sąvoka taikoma vardiniam procesui, nors ji tinkama ir visuminiam programų kūrimo procesui. Gebėjimas yra proceso charakteristika, išreiškiama gaunamais proceso darbo rezultatais. Aukštesnis proceso gebėjimas rodo, kad yra didesnė tikimybė, kad procesas gaus numatytus rezultatus, mažiau nukrypdamas nuo užsibrėžtų kainos, terminų, kokybės tikslų.

Gebėjimo lygis yra įvertis diskrečioje skalėje, nusakantis tam tikrą proceso gebėjimo pasiekimą. Kadangi beveik yra neįmanoma išmatuoti galimybės gauti siekiamus rezultatus, remiamasi praktikoje patvirtinta prielaida, kad kokybiškų rezultatų gavimo tikimybė tiesiogiai priklauso nuo paties proceso charakteristikų. Todėl modeliuose proceso gebėjimo lygis apibrėžiamas konstruktyviai, nurodant proceso indikatorius – veiklas, kurias atliekant sukuriami proceso aplinka, užtikrinanti jo rezultatų stabilumą.

Branda yra proceso charakteristika, nusakanti, kiek procesas yra apibrėžtas, valdomas, matuojamas, kontroliuojamas ir nuolatos gerinamas, o brandos lygis yra aiškiai apibrėžta pakopa visuminio proceso brandos evoliucijoje. Vizualiai brandos lygių įtaka organizacijos procesui pavaizduota 1 paveiksle: pirmo lygio procesas yra chaotiškas, kur tarsi debesyje neįmanoma nieko konkretaus išžiūrėti, ir valdyti galima daugiausia remiantis intuicija ir savo patirtimi, o ne objektyviomis proceso charakteristikomis; kuo aukštesnis brandos lygis, tuo labiau procesas yra aiškus, struktūruotas ir valdomas. Branda gali būti suprantama kaip konkrečios organizacijos palyginimas su įsivaizduojama „idealia“ organizacija, o brandos lygis – kaip atstumas iki „idealos“ organizacijos. Organizacija, pasiekusi aukštesnį proceso brandos lygį, tarsi pakyla į tam tikrą stabilią būseną arba kokybinį lygmenį, kuriame ji kokybinėmis charakteristikomis iš esmės skiriasi nuo žemesnio lygio organizacijų. Akivaizdu, kad aukštesnis brandos lygis implikuoja aukštesnį proceso gebėjimą (tikimybę pasiekti keliamus tikslus).



1 pav. Skirtingų brandos lygių procesai

Programų kūrimo proceso vertinimo modelis yra dviejų dimensijų modelis, sudarytas iš proceso ir gebėjimo. Proceso dimensijai yra naudojamas programų kūrimo proceso modelis, o gebėjimo dimensijai – proceso atributai, sugrupuoti į gebėjimo lygius. Pavyzdžiui, ISO/IEC 15504-5: 2006 vertinimo modelyje proceso dimensija remiasi ISO/IEC 12207 proceso modeliu, o gebėjimo dimensijai naudojami ISO/IEC 15504-2 standarte apibrėžti procesų atributai ir gebėjimo lygiai.

Atvaizdavimo problema

Populiariausi pasaulyje programų kūrimo proceso modeliai yra tarptautiniu standartu patvirtintas ISO/IEC 15504 ir CMMI, kuris yra tapęs standartu *de facto* (Pino, Garcia, Piattini, 2008). Įmonės, norėdamos būti oficialiai pripažintos platesnėje aplinkoje, renkasi vieną iš šių dviejų modelių. Būtina pažymėti, kad pasirinkimą dažniausiai lemia ne paties modelio savybės, bet išorinės priežastys, pavyzdžiui, jei įmonė siekia dirbti su JAV užsakovais, ji neabejotinai renkasi CMMI. Tokia pati padėtis yra ir Lietuvoje.

Įmonės dažniausiai pasirenka CMM/CMMI (Balandis, Laurinskaitė, 2005; Mikaliūnas, Reingardtas, 2005), o valstybės remiamuose projektuose skatinama naudoti ISO/IEC 15504 (Bendinskas, Mikaliūnas, Mitašiūnas, Ragaišis 2005; Mitašiūnas, Ragaišis 2006b).

Įmonės labai pageidautų galimybės derinti abu šiuos modelius (Ferreira, Machado, 2009; Khoshgoftar, Osman, 2009), bet vis dar nėra oficialių proceso vertinimo pagal vieną modelį rezultatų atvaizdavimo į kitą modelį. Pirmi bandymai suderinti šiuos du proceso modelius prasidėjo daugiau kaip prieš dešimt metų. Iš pradžių buvo analizuojama, ar jie apskritai yra suderinami (Rout, 1998). Vėliau buvo pradėti nagrinėti jų sąryšiai (Paulk, 1999). Esminius tyrimus atliko (Rout, 2001): buvo pateiktas CMMI v.1.1 brandos lygių atvaizdis į ISO/IEC 15504-2: 1998 gebėjimo profilius. Šių modelių ryšys patikslintas (Mitašiūnas, Ragaišis, 2006a): buvo sugraduotas pirmas gebėjimo lygis bei patikslintas ketvirto ir penkto gebėjimo lygių atvaizdavimas. Modelių gebėjimo dimensijų atvaizdį pateikė Rout ir Tuffley (2007). Su naujesnėmis proceso modelių versijomis – CMMI v.1.2 ir ISO/IEC 15504 – buvo atlikti nauji atvaizdavimai (Ragaisis, Peldzius, Simenas, 2010; Peldzius, Ragaisis, 2011). Prieš pusę metų buvo išleista nauja CMMI 1.3 versija, kol kas nėra atnaujintų atvaizdžių į ISO/IEC 15504, bet įvertinus nedidelius pakeitimus atvaizdžio pokyčiai turėtų būti minimalūs. Kaip jau buvo minėta, abu modeliai puikiai tinka įmonės proceso pranašumams ir trūkumams nustatyti, bet jų nepakanka įmonės procesui apibrėžti, nes pagal savo esmę jie nurodo, kas turi būti daroma, bet nenusako (ir negali nusakyti), kaip tai turi būti daroma. Todėl įmonės, norėdamos apibrėžti ir (ar) pagerinti savo programų kūrimo procesą, renkasi įvairius programų sistemų inžinerijos metodus, pavyzdžiui, RUP (Kruchten, 2004), XP (Beck, Andres, 2004), „Scrum“ (Schwaber, Beedle, 2002), DSDM (DSDM 2008). Metodų yra tikrai daug ir įvairių, todėl įmonei renkančis svarbu žinoti, ką pasirinktas metodas jai gali duoti: kokį proceso gebėjimą ir (ar) brandą jis gali užtikrinti. Proceso keitimas yra sudėtingas ir brangus, todėl išankstinis metodo įvertinimas

prieš diegiant leistų įmonei sutaupyti daug laiko ir lėšų.

ISO/IEC 15504 ir CMMI turi vertinimo metodikas, bet jos skirtos „gyvam“ įmonės procesui, o ne programų sistemų inžinerijos metodams vertinti. Šiuo metu yra mokslo darbų, nagrinėjančių, kokį brandos lygį pagal CMMI (Anderson, 2005; Fritzsche, Keil, 2007; Mahdy; Yellayi, 2009) ir ISO/IEC 15504 (Lami, Falcini, 2009) gali užtikrinti populiariausi judrieji metodai. Tačiau šie darbai iš esmės yra atvejo tyrimai, nes nagrinėjamas tik konkretaus judriojo metodo arba tik pasirinktų jo veiklų vertinimas, nepateikiamos jokios metodikos, kurias būtų galima pritaikyti kitam metodui arba tam pačiam metodui vertinti pagal kitą modelį. O galimybė įvertinti metodus pagal vis labiau populiarėjantį ISO/IEC 15504 daugumai įmonių yra ne mažiau svarbi.

Vieningo modelio idėja

Tuo metu, kai buvo pradėta atvaizduoti modelius, imtasi kurti ir bendrą modelį, t. y. siekta sukurti naują proceso modelį, kuris sujungtų visus tuo metu buvusius modelius. Deja, vienintelis plačiau žinomas toks modelis yra Wang ir King (2000) pasiūlytasis SEPRM (angl. *the Software Engineering Process Reference Model*). Šis modelis pavadintas vieningu, nes pagal jį įvertinus įmonės procesą rezultatus būtų galima transformuoti ir į kitus proceso modelius. Anot autorių, SEPRM turėjo tapti pagrindiniu vertinimo modeliu ir pakeisti visus kitus naudojamus proceso modelius. Visgi šis modelis neišpopuliarėjo, apskritai nesulaukė jokių vertinimų, todėl modelio taikymas ir iškeltų tikslų siekimas, kad būtų pakeisti kiti esami proceso modeliai, tapo abejotinas. Tačiau SEPRM autorių iškelta idėja, kad reikia sujungti skirtingus proceso modelius ir sukurti vertinimo rezultatų transformavimo mechanizmą, yra naudinga. Toliau pateikiami esminiai SEPRM modelio trūkumai, kurie neleido šiam modeliui išpopuliarėti. Šių trūkumų bus vengiama kuriant naują tarpinį modelį.

Anot SEPRM autorių, šis modelis ne tik apima CMM, ISO 9001, BOOTSTRAP, ISO/IEC 15504, bet dar ir siūlo papildomų detaliųjų praktikų, kurių, manoma, jiems trūksta. Ši išva-

da sukelia prieštaravimą, nes modeliai atspindi programinę įrangą kuriančių įmonių geriausią sukauptą praktiką, todėl kiekvienas modelis savo procesą apibrėžia išsamiai pagal to laiko situaciją. Neaišku, kaip SEPRM buvo gautos tos praktikos, kurių nėra nė viename iš kitų proceso modelių. Dauguma šių praktikų, kurių nėra kituose modeliuose, konkrečiai įvardija, kaip reikia daryti, nors turėtų būti aprašoma, kas turi būti daroma nenurodant konkrečių mechanizmų. Pavyzdžiui, CMM ir ISO/IEC 15504 modeliuose nėra nė vienos praktikos iš SEPRM procesų: Kūrimo pagalbiniai įrankiai, Valdymo pagalbiniai įrankiai. Bet CMMI šių procesų praktikos yra specifinių praktikų detaliosiose praktikose, kurios nėra privalomos, o yra tik informacinio pobūdžio, t. y. palengvina įgyvendinti praktikas. Todėl galima daryti išvadą, kad SEPRM nėra išsamesnis už kitus modelius, o tiesiog į normatyvinį lygį iškelia objektus, kituose modeliuose apibrėžtus kaip informacinius. Tokiu būdu SEPRM apriboja įvairių metodų taikymą, nes reikalauja naudoti konkrečias technologijas ar įrankius. Toks procesų praktikų detalizavimas siaurina šio modelio taikomumą, nes orientuojasi į to laiko technologijas, o jos gali greitai pasikeisti. Proceso modeliai neturėtų reikalauti naudoti konkrečių technologijų.

Dar viena SEPRM modelio problema yra gebėjimo dimensija. Nors pagal šį modelį galima įvertinti organizacijos gebėjimą, o paskui jį transformuoti į kitus modelius, tačiau SEPRM gebėjimo lygiai suteikiami visiškai kitaip nei CMM ar ISO/IEC 15504. Pastaruosiuose modeliuose, kad procesas būtų pirmo gebėjimo lygio, jis turi visiškai tenkinti pirmo gebėjimo lygio reikalavimus: procesas turi pasiekti specifinius tikslus. Tačiau SEPRM reikalaujama, kad proceso tikslas pirmame lygyje būtų pasiektas tik 20–39 proc. ISO/IEC 15504 reikalauja >50 proc. pasiekiamumo, o pagal CMMI visos praktikos turi būti įgyvendintos ir esamų trūkumų visuma nedaryti didelio neigiamo poveikio tikslui pasiekti – procentinės išraiškos nėra, bet pagal aprašymą jį galima įvertinti kaip >50 proc. Likusiuose keturiuose gebėjimo lygiuose yra ne tik bendrieji procesų reikalavimai, bet ir reikalaujama pasiekti vis daugiau vertinamų procesų

tikslų. Anot SEPRM autorių, proceso tikslai turi būti visiškai pasiekti tik penktame lygyje, o štai ISO/IEC 15504 to reikalauja antrame lygyje. ISO/IEC 15504 ir CMMI gebėjimo dimensijų esmė beveik nesiskiria. Lieka neaišku, kodėl SEPRM autoriai pasiūlo naują požiūrį į gebėjimo dimensiją, todėl kyla abejonė dėl SEPRM tinkamumo gebėjimui vertinti.

Problemų kyla ir dėl SEPRM modelio formalaus aprašymo. SEPRM pateikia formalų gebėjimo vertinimo algoritmą. Autoriai teigia, kad taip būtų išvengiama klaidų vertinant, tačiau toks matematinis vertinimas netinkamas, nes vertinimą įmonėse gali būti taikomos skirtingos praktikos procesų tikslams pasiekti. Pavyzdžiui, vertinant pagal CMMI žiūrima ne tik į formalų praktikos įgyvendinimą, bet ir į esamus trūkumus, kurie gali neleisti pasiekti procesui pirmo lygio, net jei aritmetiškai praktikos yra įgyvendintos 90 proc. Taip pat CMMI leidžiamas ir alternatyvus specifinio tikslo įgyvendinimas, kurį būtų galima pasiekti naudojant judriuosius ar kitokius metodus. Tokiu atveju specifinės praktikos galėtų būti visiškai kitokios. Todėl įmonės vertina žmonės, kurie sprendžia, ar tam tikrų praktikų dalinis įgyvendinimas turi esminių trūkdujų proceso tikslui pasiekti, ar ne.

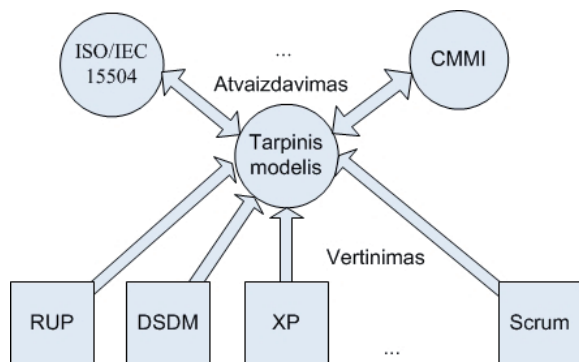
Kita SEPRM problema kyla iš brandos vertinimo atmetimo. SEPRM autoriai teigia, kad brandos lygiuose yra nepagrįsti procesų grupavimai ir toks vertinimas apskritai nenaudingas. Tačiau praktikoje tiek pakopinis, tiek tolydusis organizacijos vertinimas turi savo pranašumų ir trūkumų ir nė vieno negalima atmesti. Kaip jau buvo minėta, brandos lygis labai svarbus atliekant įmonių atranką, nes leidžia objektyviai palyginti įmonių galimybes tinkamai atlikti projektą. Taip pat brandos lygiai pateikia organizacijai gairių, kaip tobulinti savo programų kūrimo procesą. Brandos lygių procesai surinkti neatsitiktinai, bet remiasi įmonių patirtimi. Pakopinio vertinimo tikslingumą įrodo ir jo atsiradimas ISO/IEC 15504, kuris visą laiką buvo tolydus. SEPRM akivaizdi klaida vertinant CMMI buvo ta, kad jis laikytas tolydžiuoju modeliu, nors paimta versija, kurioje jis buvo tik pakopinis.

Galiausiai problemų kelia ir šio modelio senumas. SEPRM transformuoja rezultatus į jau pasenusias proceso modelio versijas. Jame nėra aprašytos metodikos, kaip galima pakeisti modelius naujomis versijomis, kaip papildyti visai nauju proceso modeliu. Todėl dabar šį modelį galima naudoti tik kaip teorinį pavyzdį bandant sukurti bendrą modelį, kuris jungtų skirtingus proceso modelius ir atliktų vertinimo transformacijas.

Peržvelgus esminius modelio trūkumus, darytina išvada, kad jis yra netinkamas vienu proceso modelių vertinimo rezultatams atvaizduoti į kitus, daugių daugiausia galima tik pažūrėti, kaip yra dengiamos modelių praktikos. Gebėjimo lygių analizei jis taip pat nenaudingas. Vadinas, šis modelis netinka, kai norima įvertinti organizaciją vieną kartą, o gauti rezultatus pagal kelis proceso modelius.

Tarpinis modelis

Tarpinis programų kūrimo proceso modelis galėtų išspręsti aprašytas problemas: padėti transformuoti vertinimo rezultatus tarp proceso modelių, išspręsti atvaizdavimo problemą, rasti brandos ir (ar) gebėjimo lygius, kuriuos galėtų užtikrinti populiariausi judrieji ir kiti metodai. Taip pat būtų kuriami abipusiai atvaizdžiai tarp ISO/IEC 15504 ir CMMI bei programų sistemų inžinerijos metodų vertinimo pagal tarpinį modelį metodika. Tokio tarpinio modelio ryšiai su kitais proceso modeliais ir metodais schemiškai vaizduojami 2 paveiksle.



2 p a v. Tarpinio modelio ryšiai

Toks tarpinis modelis leistų transformuoti įmonės programų kūrimo proceso vertinimo pagal ISO/IEC 15504 rezultatus į vertinimą pagal CMMI ir atvirkščiai bei, neatlikus diegimo įmonėje, įvertinti, kokią gebėjimą ir (ar) brandą pagal ISO/IEC 15504 ir CMMI gali užtikrinti pasirinkto judriojo ar kito programų kūrimo metodo taikymas. Taip pat būtų kuriama metodika, kaip reikėtų praplėsti tarpinį modelį, kad būtų galima sėkmingai atlikti vertinimus pagal kitą programų kūrimo proceso modelį ar atnaujintą esamo proceso modelio versiją. Vertinant pagal esamus proceso modelius, galima gauti vertinimo sertifikatus, kuriuose nurodoma, kad vertintos organizacijos procesai pasiekia tam tikrus gebėjimo lygius (brandą). Atlikus vertinimą pagal tarpinį modelį įmonė negautų jokio sertifikato, tačiau rezultatus galėtų transformuoti į vertinimus pagal CMMI ar ISO/IEC 15504.

Tarpinio modelio reikalavimai

Šiame skyriuje yra pateikiami šeši reikalavimai, kuriuos turi tenkinti konstruojamas tarpinis programų kūrimo proceso modelis.

Programų kūrimo proceso ontologija

Norint pristatyti ar aprašyti bendras modelių savybes, kyla neaiškumų, kokius terminus vartoti. Pavyzdžiui, tai, kas ISO/IEC 15504 vadinama procesu, CMMI vadinama proceso sritimi, kas ISO/IEC 15504 vadinama proceso atributais, CMMI – bendraisiais tikslais. Todėl reikia nuolat patikslinti, kurio modelio terminais remiasi aprašymas. Dėl skirtingų terminų ir apibrėžimų ne visada galima vienareikšmiškai nustatyti vieno modelio savybės atitikmenį kitame modelyje. Tokia padėtis netenkina, todėl reikalaujama sukurti ontologiją, pagal kurią būtų galima susieti skirtingų modelių terminiją, kuri dabar yra visiškai nesuderinta.

Naudojant šią ontologiją būtų aišku, kokie yra objektų atitikmenys kituose proceso modeliuose. Aprašymai būtų aiškiai ir vienareikšmiškai atspindimi skirtinguose modeliuose pagal apibrėžtus terminų ryšius. Kad prie tarpinio modelio būtų galima pridėti naujus proceso modelius, reikia turėti metodiką, kaip naujo proceso

modelio objektus suderinti su šia ontologija. Tai kartu padės nustatyti, ar modelį apskritai galima pridėti prie esamo tarpinio modelio. Jeigu jam neįmanoma pritaikyti ontologijos, darytina išvada, kad jis yra nesuderinamas su kitais proceso modeliais arba galbūt mėginama pridėti net ne proceso modelį. Šio reikalavimo rezultatas būtų programų kūrimo procesų modelių ontologija. Taip pat sukurta ontologijos pritaikymo naujam pridedamam proceso modeliui metodika.

Formalizuotas modelio aprašymas

Naudojantis apibrėžta ontologija reikia formalizuotai aprašyti proceso modelius, kad visų apibrėžimai būtų tokios pačios formos. Tada išryškėtų jų panašumai ir skirtumai, būtų aišku, kas juos jungia. Reikia turėti vienareikšmiškai aprašytą praktikų įgyvendinimo, tikslų pasiekimo, proceso gebėjimo įvertinimą. Taip pat sukurti formalizuotą gebėjimo nustatymo aprašą, pagal kurį būtų pateikiami visų modelių gebėjimo nustatymo aprašai. Sukūrus skirtingų proceso modelių formalizuotus aprašus, būtų galima sudaryti tarpinio proceso modelio aprašą. Toks formalizuotas modelio aprašas palengvintų pridėti naujus proceso modelius prie proceso modelių, pagal kuriuos galėtų būti atvaizduojami turimi tarpinio modelio vertinimo rezultatai.

Modelio išsamumas

Turint formalizuotą modelio aprašymą, reikia nustatyti jo apimtį: kokias įmonės veiklas jis turi apimti ir kaip jos turi būti plečiamos ar siaurinamos. Kadangi skirtingi modeliai apima skirtingas organizacijos veiklos sritis, todėl kiekvienas modelis turi specifinių sričių, kurių nėra kituose. Kad atvaizdavimas tarp modelių būtų visapusiškas, tarpinis modelis turi apimti įtrauktų procesų tikslus (CMMI ir ISO/IEC 15504). Taip pat turi būti aprašyta, kaip modelis turi būti plečiamas, jei atvaizdis papildomas nauju modeliu, kuriame yra naujų praktikų, kurių iki tol nebuvo, nes modelis turi užtikrinti visą atvaizdavimą į pasirinktus modelius. Pridedant naujas praktikas, gali tekti koreguoti ir esamas, nes šios gali iš dalies klotis su naujomis, todėl esamas praktikas reikėtų skaidyti ar kitaip jungti. Turi būti apibrėžta modelio apimtis, procesų tikslai ir rezultatai. Šio reikalavimo rezultatas – esamų

proceso modelių pagrindu sukurtas visas tarpinis proceso modelis, taip pat metodika, kaip praktikos būtų keičiamos prisidėjus naujam proceso modeliui ar keičiantis esamų modelių versijoms.

Proceso vertinimo modelis

Sukūrus proceso modelį, turi būti pateiktas ir vertinimo modelis, pagal kurį organizacija galėtų sudaryti savo gebėjimo profilį. Tam reikia sukurti tarpinio modelio gebėjimo dimensiją, kurią sudaro gebėjimo lygiai, proceso atributai ir vertinimo aprašas. Proceso vertinimo modelis turi būti suderinamas su prieš tai apibrėžta ontologija bei visu tarpiniu proceso modeliu, o vertinimo modelis – su kitais vertinimo modeliais (CMMI, ISO/IEC 15504). Taip pat turi būti kuriamas vertinimo procesas, kuris reikalingas programų kūrimo inžinerijos metodų užtikrinamam gebėjimui įvertinti.

Vertinimo rezultatų transformuojamumas

Kuriant vertinimo modelį, būtina jį derinti su kitais proceso modeliais. Reikalaujama išvengti keleriopo vertinimo norint gauti procesų gebėjimo profilius pagal skirtingus modelius. Turi būti galimybė proceso vertinimo rezultatus, gautus pagal tam tikrą vertinimo modelį, viena-reikišmiškai transformuoti į vertinimo pagal kitą modelį rezultatus per tarpinį vertinimo modelį. Tarkime, įmonė turi savo proceso vertinimo rezultatus pagal CMMI ir nori sužinoti, koks

vertinimo rezultatas būtų pagal ISO/IEC 15504. Tokiu atveju ji transformuotų CMMI vertinimo rezultatus į tarpinį modelį, o tada – į vertinimo pagal ISO/IEC 15504 rezultatus. Tokiu būdu įmonė įvertins savo procesus tik pagal vieną modelį, o kitų modelių įvertinimus gaus per tarpinį vertinimo modelį. Bus pasiektas pagrindinis tikslas: įmonei nereikės kelis kartus atlikti realaus proceso vertinimo.

Empiriškumas ir aprašomumas

Modelis turi remtis geriausiomis programinės įrangos kūrimo praktikomis. Įmonių procesą geriausiai atspindi iš jų pačių kilusios ir sėkmingai išbandytos teorinės praktikos, o ne formaliai sukurtas teorinis procesas, kuris nebuvo išbandytas organizacijų veiklose. Jis turi aprašyti „ką daryti“, o ne „kaip daryti“, iš ši klausimą paliekama atsakyti metodams.

Išvados ir rezultatai

Įmonės, naudodamos tarpinį modelį, turėtų išspręsti daugelį su keleriopu vertinimu susijusių problemų. Tarpinis modelis padėtų įvertinti organizacijos procesą pagal kelis modelius arba pasirinkti, kurį metodą naudoti. Šiame straipsnyje pateikti būtini tokio tarpinio programų kūrimo proceso modelio reikalavimai, kuriuos įgyvendinus būtų galima pasiekti šiuos išskeltus tikslus.

LITERATŪRA

ANDERSON, David J. (2005). Stretching Agile to fit CMMI Level 3. In *Experience Report Proceedings of the Agile Development Conference*, Denver, July 2005, p. 193–201.

BALANDIS, Osvaldas; LAURINSKAITĖ, Laura (2005). Software Process Improvement in Lithuania – UAB Sintagma Case Study. *Information Technology and Control*, vol. 34, no. 2A, p. 195–201.

BECK, Kent; ANDRES, Cynthia (2004). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional. 224 p. ISBN 978-0321278654.

BENDINSKAS, Vitolis; MIKALIŪNAS, Gediminas; MITAŠIŪNAS, Antanas; RAGAIŠIS, Saulius (2005). Towards Mature Software Process. *Information Technology and Control*, vol. 4, no. 2A, p. 209–214.

CMMI® for Development, Version 1.3, *Improving processes for developing better products and servi-*

ces, CMU/SEI-2010-TR-033, Software Engineering Institute, 2010.

DIJKSTRA, Edsger W. (1972). The Humble Programmer. *Communications of the ACM*, vol. 15, no. 10, p. 859–866.

DSDM Atern (2008). *The Handbook*. DSDM Consortium. 201 p. ISBN 0-9544832-2-7.

FERREIRA, Andre; MACHADO, Ricardo (2009). Software Process Improvement in Multimodel Environments. In *Fourth International Conference on Software Engineering Advances*, p. 512–517.

FRITZSCHE, Martin; KEIL, Patrick (2007). Agile Methods and CMMI: Compatibility or Conflict? *e-Infomatica Software Engineering Journal*, vol. 1, issue 1.

ISO/IEC 15504-5: 2006 Information technology – Process Assessment – Part 5: An exemplar Process Assessment Model.

KHOSHGOFTAR, Mohammad; OSMAN, Omar (2009). Comparison of maturity models. In *2nd IEEE International Conference on Computer Science and Information Technology*, p. 297–301.

KRUCHTEN, Philippe (2004). *The Rational Unified Process: An Introduction*. Addison-Wesley Professional. 310 p. ISBN 0321197704.

LAMI, Giuseppe; FALCINI, Fabio (2009). Is ISO/IEC 15504 Applicable to Agile Methods? *Agile Processes in Software Engineering and Extreme Programming Lecture Notes in Business Information Processing*, vol. 31, part 3, part 4, p. 130–135.

MAHDY, Ahmed; YELLAYI, Sree (2009). Practical Report: CMMI Measurements and Analysis practices based on Agile (Scrum) Method. *CMMI 9th Technology Conference and User group*.

MIKALIŪNAS, Gediminas; REINGARDTAS, Martynas (2005). Software Process Improvement in Lithuania – AB Alna Case Study. *Information Technology and Control*, vol. 34, no. 2A, p. 215–218.

MITAŠIŪNAS, Antanas; RAGAIŠIS, Saulius (2006a). Government-Industry-Academia Partnership in Software Process Improvement. *Baltic IT&T review*, no. 1, p. 45–50.

MITAŠIŪNAS, Antanas; RAGAIŠIS, Saulius (2006b). Relationship between CMMI maturity levels and ISO/IEC 15504 processes capability profiles. In *Databases and information systems: 7th international Baltic conference*, p. 119–129.

PELDZIUS, Stasys; RAGAIŠIS, Saulius (2011). Comparison of Maturity Levels in CMMI-DEV and ISO/IEC 15504. In *Proceedings of 5th WSEAS International Conference on COMPUTER ENGINEE-*

RING and APPLICATIONS Puerto Morelos, Mexico, January 29–31, p. 117–122. ISBN: 9789604742707.

PINO, Francisco; GARCIA, Felix; PIATTINI, Mario (2008). Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review. *Software Quality Journal*, vol. 16(2), p. 237–261.

PAULK, Mark (1999). C. Analyzing the Conceptual Relationship Between ISO/IEC 15504 (Software process Assessment) and the Capability Maturity Model for Software. In *International Conference on Software Quality*, Cambridge.

RAGAIŠIS, Saulius; PELDZIUS, Stasys; SIME-NAS, Justinas (2010). Mapping CMMI-DEV Maturity Levels to ISO/IEC 15504 Capability Profiles. In *Proceedings of the 9th WSEAS International Conference on Telecommunications and informatics*, Catania, Italy, May 29–31, p. 13–18. ISBN 9789549260021.

ROUT, Terence (1998). SPICE and the CMM: is the CMM compatible with ISO/IEC 15504. In *AquIS'98*, Venice, Italy, March 1998.

ROUT, Terence; TUFFLEY, Angela; CAHILL, Brent (2001). *CMMI Evaluation: Capability Maturity Model Integration Mapping to ISO/IEC 15504 2:1998*, Software Quality Institute, Griffith University, Brisbane.

ROUT, Terence; TUFFLEY, Angela (2007). Harmonizing ISO/IEC 15504 AND CMMI. *Software Process Improvement and Practice*, vol. 12, p. 361–371.

SCHWABER, Ken; BEEDLE, Mike (2002). *Agile Software Development with Scrum*. Prentice Hall. 158 p. ISBN 0-13-067634-9.

WANG, Yingxu; KING, Graham (2000). *Software engineering processes: Principles and Applications*. CRC Press LLC. 708 p. ISBN 0849323665.

REQUIREMENTS OF TRANSITIONAL SOFTWARE PROCESS MODEL

Stasys Peldžius, Saulius Ragaišis

Summary

Most of the software development projects face the following problems: projects are delayed; they overrun the budget; customers are dissatisfied with the quality of the software delivered. This phenomenon is so widespread that it is even called a software crisis. It has been understood that software process maturity is closely related to project success and the quality of a software product, so various software process models have been introduced. Organizations want to get all the advantages of the different process models that stimulate their harmonization and the investigation of process improvement in multimodel environments. Organizations seek to get their capability/maturity assessment ac-

ording to several process models by conducting only a single assessment. This could be achieved by the transitional software process model. Organizations employ different software development methodologies, but they are unable to assess how they affect their process without implementation. The transitional model helps to assess the capability/maturity according to ISO/IEC 15504 and CMMI which could be ensured by agile or other software development methodologies. Similar ideas have been introduced in the unified SEPRM model. A detailed review of the deficiencies that led to its unpopularity is made. The paper presents the requirements for the transitional software process model.