

Kauno technologijos universitetas

Informatikos fakultetas

Kokybės valdymo modeliai (bendra apžvalga)

Tadas Laurinaitis, IFM-3/2

Studentas

Doc. dr. Armantas Ostreika

Dėstytojas

2024, Kaunas

Turinys

Ižanga	3
Rayleigh modelio karkasas (The Rayleigh Model Framework)	4
CMMI (Capability Maturity Model Integration)	6
Six sigma	8
Išvados	10
Literatūros sąrašas	11

Ižanga

Anot knygos “Metrics and Models in Software Quality Engineering” autorių Stephen H. Kan ir Addison Wesley, yra svarbu įvertinti programinės įrangos kokybę, numatyti galimų defektų skaičių ir įvertinti vidutinį laiką iki kito gedimo, po to kai kūrimo darbai bus baigti, tačiau dar svarbiau yra stebėti ir valdyti programinės įrangos kokybę, kai ji kuriama. Šios užduotys tenka programinės įrangos kokybės valdymo modeliams ir procesų metrikoms, ir yra pagrindiniai jų tikslai. Nors kokybės valdymo modeliai mažiau tikslūs ir matematiškai nei nuspėjamieji modeliai, į jų tikslus taip pat patenka ir ankstyvių pablogėjimo ar pagerėjimo ženklų pastebėjimas, pagrinde tam, kad galima būtų planuoti ir vykdyti atitinkamus tolimesnius žingsnius kuo anksčiau. Kad šie modeliai būtų naudingi programinės įrangos kūrimo organizacijai, jie turi padengti pradines šio proceso fazes. Nors modeliai, grįsti duomenų, surinktų kūrimo proceso pabaigoje, padeda programinės įrangos patikimimui užtikrinti, tačiau jie taip pat palieka labai mažai laiko reikiamiems veiksams defekto radimo atveju, kadangi procesas jau būna paskutinėje fazėje. Lyginant su patikimumo modeliais, kurių literatūroje galima rasti visą galybę, kokybės valdymo modelių naudojamų kūrimo proceso metu yra ne daug [8].

Šiame darbe tiriami mažiau girdėti, tačiau aktualūs ir vis dar naudojami kokybės valdymo modeliai ir metodikos [1]:

1. Rayleigh modelio karkasas (The Rayleigh Model Framework)
2. CMMI (Capability Maturity Model Integration)
3. Six Sigma

Rayleigh modelio karkasas (The Rayleigh Model Framework)

Rayleigh modelio karkasas - kokybės valdymo modelis, kuris pasižymi tam tikra matematine struktūra, naudojama prognozuoti defektų skaičių ir jų šalinimo veiksmų efektyvumą per vystymo ciklą. Vienas svarbiausių principų programinės įrangos kūrime yra „padaryti tai teisingai pirmą kartą“, o programinės įrangos kūrime, Rayleigh modelis šį principą interpretuoja taip:

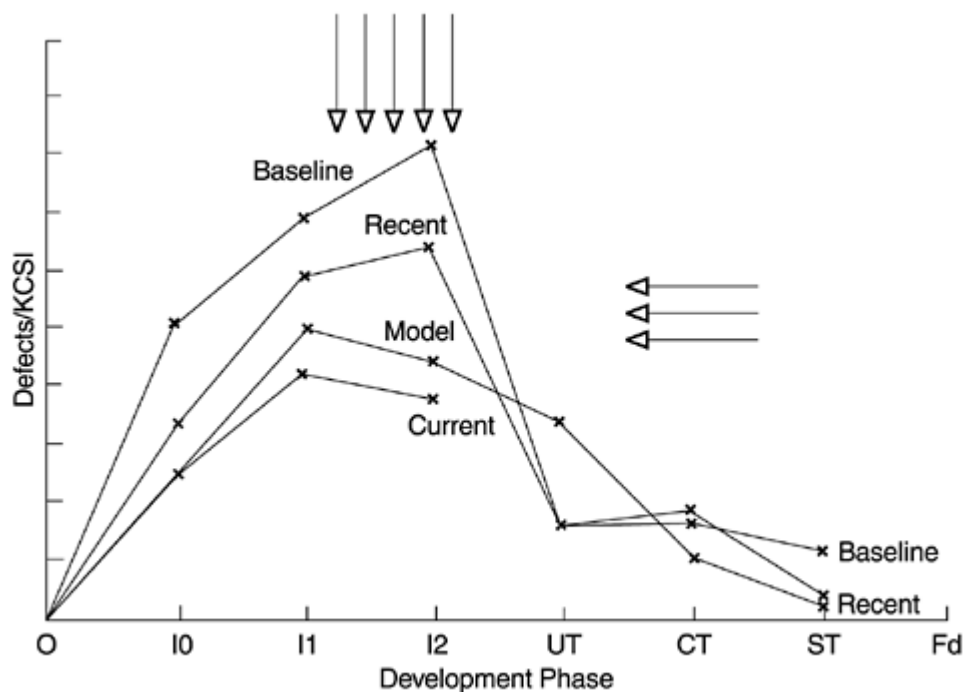
- Geriausias scenarijus yra užkirsti kelią klaidoms patekti į kūrimo procesą (**Pav. 1**)
- Atsiradus klaidoms, pagerinti priekinę priekinę kūrimo proceso dalį, kad kuo didesnis klaidų skaičius būtų ištaisytas kuo anksčiau [8] (**Pav. 1**).
- Jeigu projektas neapsiriboja dizaino ir kodo rašymo fazėmis, vienetų testai ir bet kokie papildomi testai veikia kaip papildoma apsauga, kad kuo didesnė defektų dalis būtų rasta ir panaikinta prieš kodui [8] papuolant į produkcinės aplinkas ir kad kuo mažesnė jų dalis būtų integruota į sisteminės bibliotekas. Vienetų testų ir kitos testavimo fazės prieš kodui integruojantis į bendrą sistemos struktūrą yra paskutinis šansas padaryti viską teisingai iš pirmo karto [8].

		Defect Rate	
		Higher	Lower
Inspection Effort	Higher	Not Bad/ Good	Best Case
	Lower	Worst Case	Unsure

Pav 1. Defektų ieškojimo pastangų ir defektų kiekio sąryšis [8]

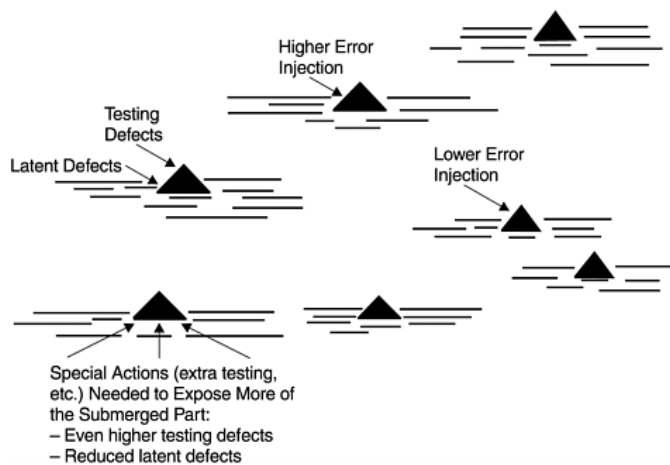
Bendrai, Rayleigh modelis yra puikus modelis kokybės valdymui. Jis artikuliuoja defektų prevencijos ir ankstyvų defektų pašalinimo taškus ir užtikrina, kad kuo anksčiau sumažinus klaidų patekimo į sistemą kiekį, sumažės ir defektų kiekis, randamas vėlesnėse testavimo fazėse, bei vėliau pasitaikysiantis programinei įrangai patekus į realaus naudojimo fazę [8].

Pagal Rayleigh modelį, defektų šalinimo greitis yra aukštesnis pradžioje ir pamažu mažėja laikui bėgant (**Pav. 2**). Šis modelis padeda organizacijoms suprasti, kaip keičiasi defektų skaičius ir kaip greitai jie gali būti pašalinti per vystymo procesą. Šis modelis taip pat suteikia pagrindą prognozuoti kokybės valdymo veiksmų efektyvumą ir leidžia organizacijoms projekto vykdymo metu planuoti tinkamus šalinimo veiksmus, siekiant užtikrinti aukštą programinės įrangos kokybę ir klientų pasitenkinimą [8].



Pav 2. Defektų šalinimo pagerėjimas kūrimo metu [8]

Naudojantis ledkalnio analogija nupasakojant santykį tarp testavimo ir defektų, aptinkamų realioje aplinkoje kiekio, ledkalnio viršūnė simbolizuoja defektų nustatymo kiekį testavimo metu, o ledkalnio apačia – defektų kiekį realioje aplinkoje. Ledkalnio dydis yra lygus bendram defektų kiekiui, o prasidėjus formaliam testavimui realioje aplinkoje, ledkalnis jau būna susiformavęs ir jo dydis jau būna nustatytas. Kuo didesnė ledkalnio viršūnė, tuo didesnis ir pats ledkalnis, o norint sumažinti panirusią ledkalnio dalį, turi būti skiriamos papildomos pastangos, kad iškelti šią dalį į paviršių. Ledkalnio analogija atvaizduojama **Pav. 3**.



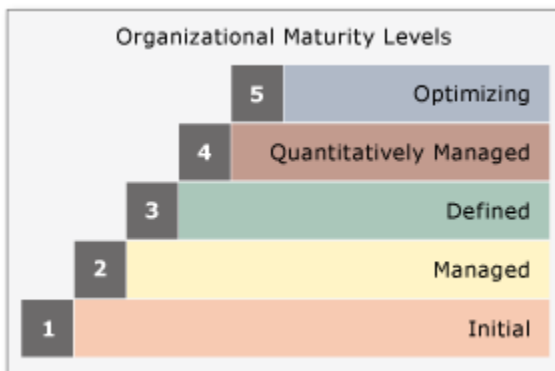
Pav 3. Ledkalnio analogijos pavyzdys Rayleigh kokybės valdymo modelyje [8].

CMMI (Capability Maturity Model Integration)

CMMI - struktūrinis kokybės valdymo modelis, sukurtas siekiant padėti organizacijoms įvertinti ir tobulinti jų gebėjimą efektyviai vystyti ir prižiūrėti programinę įrangą bei kitas sistemas. CMMI modelis buvo sukurtas 1987 Carnegie-Mellon Universiteto tyrimų centre, kurį įkūrė ir finansavo JAV gynybos departamentas. Šis modelis suteikia galimybę organizacijoms stebėti savo vystymosi lygį ir nustatyti, kuriose srityse reikia tobulėti ir daryti pakeitimus, kad pasiekti aukštesnį lygį [5].

Šis modelis buvo sukurtas kaip pagrindas organizacijos programinės įrangos kūrimo proceso pagerinimo iniciatyvai. Svarbu suprasti, kad šis modelis neapibrėžia tikslaus proceso, bet yra tam tikras žemėlapis, kuris atskleidžia trūkumus esamuose procesuose, ir padeda juos sutvarkyti. Fundamentali CMMI dalis yra procesų aplinka, kuri apibrėžia tikslus ir veiksmus, kurie veda prie tų tikslų [2][3].

CMMI-DEV yra sudarytas iš dviejų modelių, kurie dalijasi tuo pačiu pagrindiniu elementų rinkiniu. Pirmas ir labiausiai žinomas yra Staged Representation (liet. Stadijų reprezentacija) – apibrėžiantis 22 procesų aplinkas, suskirstytas į 5 organizacijos brandumo lygmenis. Brandumo lygmenys: *Initial* (liet. Pradinis), *Managed* (liet. Valdomas), *Defined* (liet. Apibrėžtas), *Quantitatively Managed* (liet. Kiekybiškai valdomas), ir *Optimizing* (liet. Optimizuojamas). Kiekvienas lygmuo apibrėžia vis aukštesnius reikalavimus ir gerinimo tikslus organizacijos procesams. Organizacijos vertinimo metu nustatomas brandumo lygmuo indikuoja organizacijos gebėjimą valdyti riziką ir vykdyti pažadus [2][4].



Pav 4. CMMI modelio organizacijos brandumo lygmenys [2].

Akronimas	Proceso aplinka
CAR	Causal Analysis & Resolution
CM	Configuration Management
DAR	Decision Analysis & Resolution
IPM	Integrated Project Management
MA	Measurement & Analysis
OID	Organizational Innovation & Deployment
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training

Akronimas	Proceso aplinka
PI	Product Integration
PMC	Project Monitoring & Control
PP	Project Planning
PPQA	Process & Product Quality Assurance
QPM	Quantitative Project Management
RD	Requirements Definition
REQM	Requirements Management
RSKM	Risk Management
SAM	Supplier Agreement Management
TS	Technical Solution
VER	Verification
VAL	Validation

Lentelė 1. 22 proceso aplinkos, sudarančios CMMI modelį [2].

Organizacijos, taikydamos CMMI, gali įgyvendinti procesų tobulinimo strategijas, siekdamos efektyviau planuoti, kurti, testuoti ir prižiūrėti savo programinę įrangą. CMMI suteikia struktūrizuotą kelią organizacijoms siekti aukštos kokybės standartų ir efektyvumo, o taip pat padeda užtikrinti, kad procesai būtų nuolat tobulinami, atitinkantys besikeičiančius rinkos poreikius ir standartus. Tai stiprus įrankis organizacijoms, siekiančioms tapti arba išlikti konkurencingomis, bei siekiančioms nuolatinio tobulėjimo savo dalykinėse srityse [2].

Naudoti šį modelį kaip procesų tobulinimo programos pagrindą gali būti pavojinga tai atvejais, kai implementatoriai pamiršta, kad CMMI nėra nei procesas, nei darbo eigos modelis, o modelis, padedantis teikti tikslus tiek procesui, tiek darbo eigai. Šių tikslų įgyvendinimas didina organizacijos brandumą ir tikimybę, kad užsibrėžti organizaciniai tikslai vyks taip kaip suplanuota. Svarbu prisiminti, kad bet kokios organizacijos procesų tobulinimo veiklos tikslas turėtų būti pamatuojamas pagerėjimas, o ne skaičius, pasiektas sukuriant procesus ir infrastruktūrą vien tam, kad galima būtų praeiti įvertinimą ir pakilti į aukštesnį brandos lygmenį [2][3].

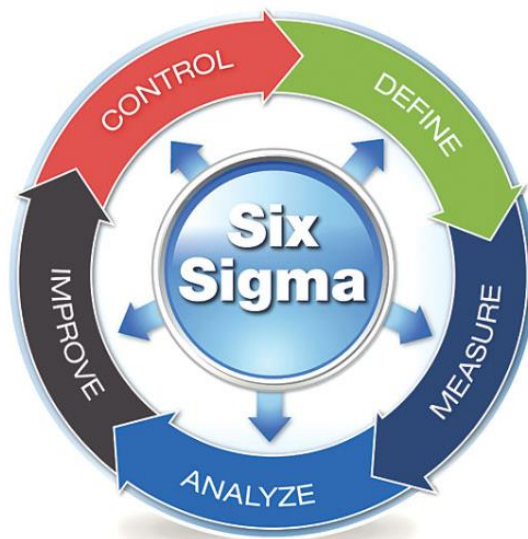
Six sigma

Six Sigma (6σ) yra procesų tobulinimo metodika, kuri siekia sumažinti klaidų ir defektų skaičių organizacijos veikloje. Ši metodika yra plačiai naudojama tiek pramonėje, tiek paslaugų sektoriuje, įskaitant programinės įrangos kūrimą.

Pagrindinis šios metodikos tikslas yra optimizuoti organizacijos veiklą, padidinant produktyvumą ir kokybę bei mažinant laiko ir resursų švaistymą. Six Sigma modelis remiasi statistiniais metodais ir procedūromis, kuriomis siekiama identifikuoti, analizuoti ir pašalinti veiksnius, kurie prisideda prie defektų atsiradimo. Pagrindinis šio modelio principas yra koncentracija į procesų tobulinimą ir optimizavimą, siekiant sumažinti procesų variaciją ir panaikinti defektus bei nukrypimus nuo norimo rezultato.

Pagrindiniai Six Sigma etapai [6][7][9]:

- Apibrėžimas (Define): Nustatomi ir apibrėžiami procesai, kurie yra svarbūs organizacijai, bei jų tikslai ir reikalavimai.
- Matavimas (Measure): Atliekamas procesų matavimas, siekiant nustatyti dabartinę situaciją, identifikuoti problemas ir sukurti matavimo rodiklius.
- Analizė (Analyze): Atliekama išsamioji analizė, siekiant identifikuoti pagrindines problemų priežastis ir nustatyti galimas gerinimo sritis.
- Tobulinimas (Improve): Įgyvendinami procesų tobulinimo veiksmai, siekiant pašalinti identifikuotas problemas ir optimizuoti procesus.
- Kontrolė (Control): Sukuriamos kontrolės priemonės ir procedūros, kad būtų užtikrinta, jog pagerinimai bus išlaikyti ir procesai veiks efektyviai ir stabiliai.



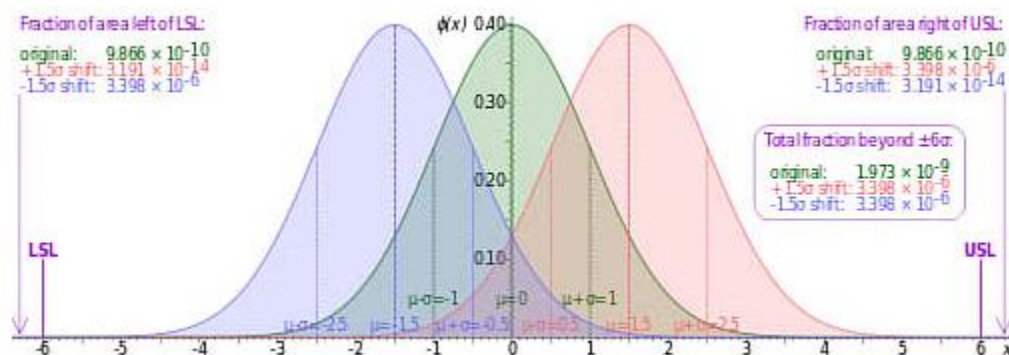
Pav 5. CMMI modelio organizacijos brandumo lygmenys [9].

Six Sigma modelis turi griežtus reikalavimus ir standartus, kurie turi būti laikomi siekiant sėkmingai įgyvendinti tobulinimo procesą. Taip pat šis modelis naudoja specialius ženklinimo būdus ir metodikas, tokius kaip DMAIC (Define, Measure, Analyze, Improve, Control) arba DMADV (Define, Measure, Analyze, Design, Verify), kad padėtų organizacijoms struktūrizuoti ir valdyti tobulinimo veiklą. Modelio sėkmės priežastys yra didelis dėmesys kokybei, mokslškai pagrįstos sprendimų priėmimas, statistinė analizė ir nuoseklus procesų

tobulinimas. Šie veiksniai leidžia organizacijoms efektyviai pasiekti savo tikslus ir pagerinti veiklos rezultatus [7].

Sigma six etimologija remiasi Graikų abėcėlės simboliu „sigma“ arba „ σ “ statistiniu terminu skirtu matuoti proceso nuokrypį proceso tikslo. Šis modelis kilęs iš normaliosios skirstinio kreivės, naudojamos statistikoje, kur vienas Sigma simbolizuoja standartinį nuokrypį nuo vidurkio. Jeigu procesas susideda iš 6 sigmų, trijų virš ir trijų žemiau vidurkio, defektų rodiklis klasifikuojamas kaip labai mažas [6].

Žemiau pateiktas grafikas parodo statistines Sigma six modelio prielaidas priklausomai nuo pasiskirstymo. Kuo didesnis standartinis nuokrypis, tuo plačiau pasiskirsčiusios sutinkamos reikšmės, todėl procesai, kurių vidurkis yra mažiausiai 6σ nuo arčiausio specifikacijos limito nukreipiami į Sigma six modelį [6].



Pav 6. Six sigma reikšmių normalės pasiskirstymo grafikas [6].

Išvados

Šiame darbe buvo išnagrinėti mažiau girdėti, bet svarbūs ir vis dar naudojami programinės įrangos kokybės valdymo modeliai ir metodikos:

- Rayleigh modelio karkasas
- CMMI
- Six Sigma.

Nors šie modeliai nėra plačiai žinomi, ypač atėjus naujai programinės įrangos inžinierių kartais, tačiau jie net ir po daugelio metų išlieka viena ar kita forma naudojami įvairių, programinę įrangą kuriančių ir prižiūrinčių organizacijų visame pasaulyje. Šie modeliai vienokia ar kitokia forma duoda galimybę organizacijoms valdyti atsirandančius defektus ir kitokias klaidas programinės įrangos kūrimo ir priežiūros metu, bei padeda planuoti kaip jų išvengti ir kaip su jais tvarkytis ateityje. Šie metodai taip pat padeda organizacijoms išsikelti tikslus, paskirstyti prioritetus defektų ar kitų klaidų atžvilgiu šitaip padėdami joms siekti ilgalaikės sėkmės ir konkurencingumo rinkoje.

Literatūros sąrašas

- [1] Software Quality Models: A Comprehensive Guide, 2023-07-11, kreiptasi 2024-05-05, nuoroda: <https://medium.com/@publicapplicationcenter/software-quality-models-a-comprehensive-guide-7faa8d2b9009>
- [2] Background to Capability Maturity Model Integration (CMMI), 2023-06-27, kreiptasi 2024-05-05, nuoroda: <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops>
- [3] Understand CMMI process template artifacts, 2024-03-25, kreiptasi 2024-05-05 nuoroda: <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi-process?view=azure-devops>
- [4] CMMI process work item types and workflow in Azure Boards, 2023-03-25, kreiptasi 2024-05-05 nuoroda: <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi-process-workflow?view=azure-devops>
- [5] CMMI insititute pagrindinis puslapis, kreiptasi 2024-05-05 nuoroda: <https://cmmiinstitute.com/>
- [6] What is Six Sigma: Everything You Need to Know About It, 2023-03-20, kreiptasi 2024-05-05 nuoroda: <https://www.simplilearn.com/what-is-six-sigma-a-complete-overview-article#:~:text=Six%20Sigma%20is%20a%20set,3.4%20defects%20per%20million%20opportunities.>
- [7] Six Sigma, kreiptasi 2024-05-05, nuoroda: https://en.m.wikipedia.org/wiki/Six_Sigma
- [8] Metrics and Models in Software Quality Engineering, Second Edition, Stephen H. Kan, 2002-09-20, naudota: 2024-05-05
- [9] Back to Basics: Six Sigma, Edward McMenamin, 2018-01-01, kreiptasi 2024-05-05 nuoroda: <https://www.qualitymag.com/articles/94429-back-to-basics-six-sigma>