

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

# **CMMI un AGILE metodoloģijas ieviešana uzņēmumā**

MAGISTRA DARBS

Autors: Artūrs Grīnhofs

Stud.apl. ag08221

Darba vadītājs: prof. Dr. dat. Māris Treimanis

RĪGA 2010

## **Anotācija**

Darba mērķis ir veikt SIA „HC” un SIA „HC Latvia” uzņēmumu programmatūras izstrādes un sistēmu uzturēšanas procesu analīzi, atspoguļojot un novērtējot visus CMMI (Capability Maturity Model Integration) pārvaldības brieduma līmeņus un spējās metodes. Teorētiskajā daļā ir detalizēti aprakstīti CMMI un spējās pārvaldības principi. Tiek novērtēta savstarpēja procesu modeļu mijiedarbība, veicot arī pielietoto rīku apskatu, aprakstīti saistītie moduļi un funkcionalitāte. Analizējot uzņēmuma procesus, tiek izstrādāta metode, kas savstarpēji mijiedarbojas ar CMMI un spējās programmatūras izstrādes principiem un praksi, nodrošina procesu uzlabojumu veikšanu uzņēmumā, samazinot izstrādāto produktu piegādes termiņus, izmaksas un paaugstinot produktu piegādes kvalitāti uzņēmumā.

## **Abstract**

Labour aim is to make analysis of companies “HC” and “HC Latvia” enterprise software development and systems maintenance processes, reflecting and evaluating all CMMI (Maturity Model Integration Capability) management maturity levels and agile methods. In theoretical part there is described in detail CMMI and agile principles. Cross-process models relational interaction is assessed reviewing applied tools and describing associated modules and functionality. Analyzing business processes method is developed which interacting with CMMI and agile software development principles and practices provide process improvements in the company, reducing developed products delivery time, costs and raising product quality of supply in company.

## **Atslēgvārdi**

CMMI, Agile, spējā, programmatūra, process, metode, metodoloģija, ieviešana, uzturēšana, konfigurācija, pārvaldība, kvalitāte, testēšana.

## **Autoreferāts**

Šī maģistra darba izstrādes gaitā autors ir izpētījis kompāniju „HC” un „HC Latvia” projektu un inženierijas pārvaldības procesus un rīkus, veicot to rūpīgu analīzi svarīgākajās jomās pa procesu apgabaliem, pielietojis izstrādāto savstarpēji mijiedarbojošos metodi jaunu projektu izstrādē, kā arī pilnveidojis projekta izstrādes, ieviešanas un uzturēšanas procesus uzņēmumā, kā arī izstrādājis dokumentācijas ietvarus procesu kvalitātes nodrošināšanai.

# SATURS

ATTĒLU SARAKSTS .....	8
APZĪMĒJUMU UN JĒDZIENU SARAKSTS.....	9
1. IEVADS.....	10
1.1. Darbības raksturojums un pārskats .....	10
2. PROCESU SĀKOTNĒJAIS NOVĒRTĒJUMS .....	12
2.1. Projektu vadības un izstrādes procesu nozīme .....	12
2.1.1. Programminženierijas metodes .....	13
2.1.2. Projektu vadības metodes .....	13
2.2. Sākotnējo uzņēmuma rīku un atbilstošo procesu analīze.....	14
2.3. Metodoloģiju izvēles principi un statistika.....	18
3. CMMI METODOĻĢIJA UN SAISTĪTIE PROCESI .....	23
3.1. CMMI – 1.līmenis.....	24
3.2. CMMI – 2.līmenis.....	24
3.2.1. Projektu plānošana .....	25
3.2.2. Projektu uzraudzība un kontrole .....	26
3.2.3. Konfigurācijas pārvaldība .....	26
3.2.4. Produktu un procesu kvalitātes pārvaldība .....	27
3.2.5. Piegādātāju vienošanās pārvaldība .....	27
3.2.6. Mērīšana un analīze .....	27
3.2.7. Prasību pārvaldība .....	28
3.3. CMMI – 3.līmenis.....	28
3.3.1. Organizācijas procesu fokuss .....	29
3.3.2. Organizācijas procesu definīcija .....	29
3.3.3. Organizācijas apmācība .....	30
3.3.4. Integrēta projektu pārvaldība .....	30
3.3.5. Risku pārvaldība .....	31
3.3.6. Lēmumu analīze un atrisināšana .....	31
3.3.7. Prasību pārvaldība .....	32
3.3.8. Tehniskie risinājumi .....	32
3.3.9. Produktu integrācija .....	33
3.3.10. Verifikācija .....	33
3.3.11. Validācija .....	34
3.4. CMMI – 4.līmenis.....	34
3.4.1. Kvantitatīva projektu pārvaldība .....	35

3.4.2.	Organizācijas procesu virzība un attīstība	35
3.5.	CMMI – 5.līmenis	35
3.5.1.	Cēloņu analīze un atrisināšana	36
3.5.2.	Organizācijas inovācijas un to ieviešana	37
4.	SPĒJĀ PROGRAMMATŪRAS IZSTRĀDE	38
4.1.	Spējie principi un vērtības	38
4.2.	Spējās metodes	39
4.2.1.	Scrum metode	39
4.2.2.	Extreme Programming (XP) metode	43
4.3.	Metožu pārbaude un izsekojamība	44
5.	PIELIETOTĀ METODE UN SITUĀCIJAS NOVĒRTĒJUMS	49
5.1.	Izmantoto rīku novērtējums	49
5.2.	Metožu pielietojums un mijiedarbība	53
5.2.1.	Projektu novērtējums un metodes izvēle	54
5.2.2.	Pielietotās metodes apskats	54
5.2.3.	Uzdevumu uzskaitē un pārvaldība	56
5.2.4.	Uzdevumu un pakotņu testēšana	57
5.2.5.	Sadarbības nodrošināšana	57
5.2.6.	Darba uzdevumu novērtējuma sniegšana un precizitāte	58
5.2.7.	Metodes mijiedarbība	58
5.2.8.	Pielietotā inženierijas metode	62
5.3.	Procesu novērtējums un uzlabojumi	63
	NOBEIGUMS UN SECINĀJUMI	67
	IZMANTOTĀ LITERATŪRA	68
	PIELIKUMI	69
	Pielikums Nr.1 – Prasību specifikācijas dokumenta apstiprināšana (SRS)	70
	Pielikums Nr.2 – Prasību specifikācijas dokuments (SRS)	72
	Pielikums Nr.3 – Prasību izsekojamības matricas dokuments	73
	(Traceability matrix)	73
	Pielikums Nr.4 – Programmatūras projekta pārvaldības plāns	74
	Pielikums Nr.5 – Pārraudzības un kontroles nodrošināšana	84
	Pielikums Nr.6 – Procesu novērtējums SIA „HC Spania”	87
	Pielikums Nr.7 – Darbu izpildes novērtējuma anketa projektu ietvaros	90
	Pielikums Nr.8 – Darbinieku zināšanu un prasmju dokuments	91
	DOKUMENTĀRĀ LAPA	91

## ATTĒLU SARAKSTS

Attēls Nr.2.3.1. „CMMI brieduma līmeņu popularitātes novērtējums” [3]	19.lpp
Attēls Nr.2.3.2. „CMMI izmantošana dažādās pasaules valstīs” [3]	20.lpp
Attēls Nr.5.2.2.1. „Sprinta Nr.1 atlikušā darba grafiks”	55.lpp
Attēls Nr.5.3.1. „Resursu noslodzes ģenerēts pārskats”	64.lpp
Attēls Nr.5.3.2. „Atvērto pret aizvērto problēmu pieteikumu skaita analīze pa dienām”	65.lpp

## APZĪMĒJUMU UN JĒDZIENU SARAKSTS

Agile	Spējā programmatūras izstrāde, kas nodrošina sistēmas ātru un pakāpenisku ieviešanu pa posmiem, izstrādes procesā definējot jaunākās lietotāja prasības, kur katrā posmā tiek pabeigta noteikta programmatūras funkcionalitāte.
Audits	Sistemātisks, neatkarīgs un dokumentēts visu procesu un darbības objektīvs novērtējums, lai noteiktu pakāpi, līdz kurai ir izpildīti audita kritēriji
CMMI	Spēju brieduma modeļu integrācija, no angļu valodas „Capability Maturity Model Integration”
FRS	Funkcionālo prasību dokuments (Functional Requirements document)
IT	Informācijas tehnoloģijas
Izpildītājs	Projekta izpildes un ieviešanas komanda Uzņēmumā
Kvalitāte	Pakāpe, kādā piemītošu īpašību kopums izpilda izvirzītās prasības
MRD	Mārketinga prasību dokuments (Marketing Requirements document)
MS	Kompānija “Microsoft”
Pasūtītājs	Veiktā projekta pasūtījuma kontaktpersona, klients
Procedūra	Noteikts izpildes veids, kā veikt darbību vai procesu
Process	Savstarpēji saistītu vai mijiedarbībā esošu darbību un aktivitāšu kopums, kas veido rezultātu kopumu
Projekts	Pasākumu kopums, ko raksturo tā sarežģītība, veids, izstrādes mērķis, finanses, laiks, resursu pieejamība, organizācijas veids, izmantotie standarti un metodes
Rīks	Programmatūra vai tās funkcionalitāte, kas nodrošina projekta pārvaldības vai inženierijas prakšu realizāciju, kā, piemēram, informācijas uzglabāšanu un tās tālāko apstrādi (darba pakotņu uzglabāšana, koda rediģēšana, testēšana) un ir pielietots dažādu saistīto procesu nodrošināšanai
Scrum	Projektu pārvaldības metode
SRS	Sistēmas prasību dokuments (System Requirements document)
Stakeholder	Visas ieinteresētās puses darbību pārskatīšanai un apstiprināšanai
Traceability matrix	Prasību izsekojamības matrica
Uzņēmums	“HC Latvia” SIA
XP	Inženierijas pārvaldības metode

## 1. IEVADS

SIA "HC Latvia" ir meitas uzņēmums ASV esošam uzņēmumam, kas Latvijā dibinājis atsevišķu uzņēmumu lētā darbaspēka un atvieglotās nodokļu politikas dēļ salīdzinājumā ar citām valstīm. Latvijas uzņēmums tiek uzskatīts par ārzonas kompāniju. Uzņēmums nodarbojas ar maksājuma karšu terminālu un programmatūras izstrādi tiem visās Pasaules valstīs, kā, piemēram, Spānijā, Francijā, Vācijā, Krievijā, ASV, Zviedrijā, Brazīlijā u.c.

Sakarā ar to, ka uzņēmuma veiksmīgu produktu piegādei nepieciešama papildus kvalitātes procesu īstenošana un pielietošana ikdienas darbā, nepieciešams veidot vienotu pieeju produktu izstrādē pēc apstiprinātas metodoloģijas un prakses piemēriem, izmantojot rīkus un vienkāršotus procesus par kuriem ir vienojušās visas ieinteresētās puses un reģioni.

Uzņēmums "HC" ir izvirzījis galvenos nosacījumus metodoloģijas izvēlē – CMMI kā populārākā produktu ieviešanas un procesu uzlabošanas metode ASV un citās pasaules valstīs, kā arī spējā programmatūras izstrāde, biežo klientu pieprasījumu un dažādo izmaiņu dēļ, izstrādes, ieviešanas un uzturēšanas ciklu laikā.

Līdz gada beigām uzņēmumā nepieciešams izvērtēt un ieviest atbilstošus CMMI brieduma līmeņu procesu apgabalu definētās labās prakses piemērus, ar iespējamiem procesu, produktu un servisu attīstības ieteikumiem un tendences uzlabojumiem, kā arī specifiskiem kritērijiem un mērķiem, analizējot vienlaicīgi visus brieduma līmeņu procesu apgabalus un neveicot CMMI sertifikāciju. Papildus nepieciešams ieviest un pēc vajadzības arī pielietot spējās programmatūras izstrādes metodes uzņēmumā, kā arī izstrādāt dokumentu paraugformas procesu uzlabojumu nodrošināšanai.

### 1.1. Darbības raksturojums un pārskats

SIA „HC” ir viens no lielākajiem elektronisko norēķinu sistēmu programmatūras un tehnikas (karšu maksājumu termināli) izstrādātājiem un saistīto risinājumu sniedzējiem pasaulē. Esošās produktu sērijas “Optimum” piedāvātie termināli tiek lietoti katrā uzņēmumā, kam nepieciešams norēķināties ar karti. SIA „HC” ir konkurētspējīgākais uzņēmums savā nozarē, kas nodrošina pilnīgu, augstas drošības maksājumu risinājumu izstrādi terminālos, nodrošinot ātru un drošu datu pārraidi, jo piedāvā dažādu produktu un servisu klāstu atbilstoši klienta vēlmēm, iespējām un vajadzībām. Uzņēmums sniedz savu produkciju un pakalpojumus, izmantojot vispasaules tīklu, pārdošanas, servisa un attīstības birojus daudzās valstīs, kā, piemēram, ASV un dažādās Eiropas valstīs. Uzņēmums

nodarbojas ne vien ar programmatūras produkta izstrādi un projektēšanu, bet arī ar aparatūras projektēšanu, ieviešanu, daudzu sistēmu vienotu integrēšanu vienā produktā, ko tālāk piedāvā ieinteresētajiem klientiem [2].

Tāpat SIA „HC” projektē dažādas perifērijas ierīces, darījumu sakaru ierīces, kā arī darījumu pārvaldības sistēmas, lietojumprogrammas un sniedz informācijas piegādes pakalpojumus. Jaunākie maksājumu un darījumu risinājumi ietver liela ātruma, augstas drošības aparatūru, kas kopā ar programmatūru ir neatsverams ieguvums citu uzņēmumu veiksmīgai attīstībai un spējai nodrošināt atbalstu visiem maksājumu karšu veidiem. Ievērojot Uzņēmuma darbības specifiku, nepieciešams veikt pārvaldības un ieviešanas procesu analīzi, lai nodrošinātu veicamo darbu efektivitāti, pārskatāmību un kontroli, sasniegto augstu klientu apmierinātības līmeni, piegādājot savlaicīgi izstrādātu, ievērojot kvalitātes un sistēmas prasības un izmaksas, gala produktu.

## **2. PROCESU SĀKOTNĒJAIS NOVĒRTĒJUMS**

### **2.1. Projektu vadības un izstrādes procesu nozīme**

Veicot programmatūras izstrādi uzņēmumā nepieciešams pielietot vienotu procesu un rīku pieeju, izmantošanu. Sākotnējā procesu novērtējumā procesu un rīku izmantošana ir haotiska, līdz ar to apkopot visus saistītos procesu apgabalus ir sarežģīti, jo neeksistē nekādi sistēmas procesu un darbības apraksti, līdz ar to arī lēmumu pieņemšanas process bieži vien nav pamatots un ir haotisks. Taču procesu lietošana un to attīstība uzņēmumā veido un attīsta jaunas zināšanas, kā arī uzkrāj jau esošās visa uzņēmuma zināšanas. Procesu pielietošana sekmē projektu veiksmīgu virzību un arī pabeigšanu. Procesu izveide un attīstība sekmē dažādu notikumu virzītu attīstību un variācijas iespējamības likvidēšanu vai mazināšanu, ja tāda rodas. Tajā pašā laikā, paredzot notikumus iespējams jau savlaicīgi rīkoties atbilstoši iepriekš definētam plānam vai procesam, kas veidotu prognozējamu programmatūras izstrādes vidi uzņēmumā. Tāpat procesu ievērošana un aprakstīšana nepieciešama, lai attīstītu zināšanu pilnveidošanas procesu, kas ir visu uzņēmumu veiksmes kritērijs.

Veiksmīga procesa ieviešanas gadījumā tiek izstrādāti kontrolsaraksti un mērījumi, kas apraksta lielāko daļu (aptuveni 80%) no veicamā darba apjoma, līdz ar to mazākā daļa līdz gala rezultāta sasniegšanā ir jāpiedomā klāt. Jaunos uzņēmumos procesu ieviešana ir īpaši nozīmīga kā projektu savlaicīgas pabeigšanas un klientu apmierinātības rādītājs.

Situācijā, kad uzņēmums sāk gatavoties projektu izstrādei vai uzturēšanas darbiem, nepieciešams rūpīgi izsvērt pielietojamās projektu pārvaldības pieejas un inženierijas metodes pēc kādām darbinieki turpmāk strādās. Inženierijas metodes definē kādā veidā tiks izstrādāta sistēma, precīzi nosakot sistēmas tehniskos rīkus, pieejas un dažādus paņēmienus projektu nodevumu sagatavošanai. Galvenie ietekmējošie faktori ir projekta sarežģītība, prasību precizitāte un noteiktība, resursu pieejamība, kā arī veiksmīga sadarbība ar klientu. Līdz ar to katrā projektā un pat nozarē, metodes pielietojums kā veidot un organizēt katru projektu ir projekta vadītāja pienākums un atbildība.

Lai ar klientu būtu veiksmīga sadarbība jau no projekta izstrādes dzīves cikla sākuma, nepieciešams vienoties par atbilstošāko metožu pielietojumu projektā.

### ***2.1.1. Programminženierijas metodes***

Inženierijas metode nosaka izstrādes posmu secību, kārtību un prioritāti, ievērojot noteiktus darbības principus. Atbilstoši noteiktajai un izvēlētajai metodei tiek izstrādāts konkrēts projekts.

Sistēmu izstrādē pēc inženierijas metodēm nošķir [6]:

- Kovboja metodi – brīva pieeja procesu un rīku interpretācijā, nekontrolēta un nepārvaldīta, vienlaicīgi arī haotiska.
- Pareģojošo metodi – prasību specificēšana ir noritējusi veiksmīgi un tās apkopotas pietiekamā detalizācijas līmenī, kas turpmākajā projekta gaitā netiks būtiski vai arī netiks vispār koriģētas.
- Iteratīvo metodi – programmatūras izstrāde tiek īstenota ciklos, kur pēc noteikta skaita sprintiem, kas katrs nav garāks par 4 nedēļām, tiek piegādāts produkts vai kāda strādājoša produkta funkcionalitāte.

Sistēmas izstrādes galvenie posmi [8]:

- Sistēmas prasību specificēšana un analīze
- Sistēmas projektējuma aprakstīšana un analīze
- Programmatūras prasību specificēšana un analīze
- Programmatūras arhitektūras projektējums
- Programmatūras produkta detalizēts projektējums
- Programmēšana un testēšana
- Kvalifikācijas testēšana
- Integrācija
- Produkta instalācija
- Uzturēšana

### ***2.1.2. Projektu vadības metodes***

Savukārt projekta vadības metodes definē inicializācijas, plānošanas, īstenošanas, kontroles un pārraudzības, kā arī slēgšanas procesus.

Projektu vadības metodes iedala [7]:

- Tradicionālā pieeja – šajā metodē projekta pārvaldības plānu izstrādā projekta sākumā faze, kas balstīts uz nemainīgām vai nebūtiskām sistēmas prasību izmaiņām.
- Spējā (no angļu val. – Agile) pieeja – šajā metodē projekta plāns tiek veidots iterācijās pa cikliem, kur tiek definēts projekta ietvars un mērķis, bet prasības tiek precizētas projekta laika gaitā.

## **2.2. Sākotnējo uzņēmuma rīku un atbilstošo procesu analīze**

### ***Projektu plānošana un vadība***

Sākuma izvērtēšanas procesā visa projektu plānošana tiek veikta izmantojot Microsoft (turpmāk tekstā – MS) Project rīku. Šis rīks ir noderīgs mazu un salīdzinoši vienkāršu projektu pārvaldībā, lai nodrošinātu veicamo darbu apjomus un plānotos termiņus. Rīks nenodrošina atlikušā darba uzskaites procesa nepieciešamību no komandas dalībnieku puses. Lielāku un sarežģītāku projektu gadījumā, kādi parasti (četri no pieciem) tiek ieviesti Uzņēmumā, nepieciešams pārskatāmāks rīks, kas ir vieglāk izmantojams, pielietojams, nodrošinot savstarpēju saikni starp vairākiem standarta formātiem, kā arī nodrošinātu projekta finansu uzskaites un kontroles mehānismus. Papildus tam nepieciešams rīks, kas veido savstarpēju integrāciju ar citiem projektiem, tādā veidā mērot un analizējot resursu noslodzi un kopējo darba spēju.

Projektu biežo izmaiņu pieprasījumu gadījumā MS Project rīks un izmantotā metode (Tradicionālā pieeja – Ūdenskrituma modelis) nav efektīva. Projektu sarežģītība un termiņu pārlikšana paildzina visa gala produkta piegādi klientam. Visi iepriekšminētie nosacījumi liecina par projekta vadības pieejas izmaiņu nepieciešamību, kas sekmētu klienta apmierinātību, savlaicīgu spēju reaģēt uz prasību izmaiņām un uzlabotu savstarpēju komunikāciju starpreģionu un klientu līmenī. Uzsākot procesu izpēti, paralēli uzņēmumā tiek startēta un testēta Clarity programmatūra, kas nodrošina projektu integrētu pārvaldību un finansu plānošanas un pārraudzības galvenās funkcijas.

Plānošanas process tiek organizēts, sākotnēji komandas vadītājam sastādot projekta plānu. Komandas vadītājs paralēli pilda arī daļu no sistēmas arhitekta lomas. Plānošanas procesā netiek uzklauts un ņemts vērā komandas darba uzdevumu novērtējums. Komanda netiek iesaistīta darba pakotņu novērtēšanas procedūrā. Darba uzdevumi tiek sadalīti pēc nejaušības principa, neievērojot zināšanu līmeņus katram komandas dalībniekam.

### ***Projektu uzraudzība un kontrole***

Projektu pārraudzība un atjaunināšana notiek ne biežāk kā reizi divās nedēļās, parasti reizi mēnesī pie kārtējo atskaišu sagatavošanas prezentācijas veidā augstākā līmeņa vadībai. Nav iespējams izsekot regulārajām projekta izmaiņām, jo netiek kontrolēta padarītā darba uzskaitē. Projektu vadībā iesaistītajam personālam ir vājas zināšanas par projektu uzraudzību un kontroli. Projektu veiksmes kritēriji netiek statistiski uzkrāti un analizēti. Testējot Clarity programmatūru iespējams izmantot esošo funkcionalitāti, lai nodrošinātu pamata līmeņa projektu pārraudzību un kontroli. Iespējams veikt dažādus programmatūras uzlabojumus, lai pilnvērtīgi nodrošinātu dažādu projektu integrētu pārvaldību, nodrošinot arī statistikas un mērījumu uzkrāšanu un analīzi, kas vienlaicīgi saistās ar citiem procesu apgabaliem.

Sarežģītu, prioritāru un biežu izmaiņu vai jaunas funkcionalitātes rašanās gadījumos, projektos neatbilstoša kontrole pār paveiktajiem darbiem ir projektu vadītāju zināšanu trūkuma rezultāts. Papildus nepieciešams definēt pārraudzības un kontroles mehānismus, lai īstenotu atbilstošu projektu pārvaldību, ņemot vērā projekta veidu, steidzamību, sarežģītību, resursu apmācības nepieciešamību un citus saistītos faktoros.

### ***Konfigurācijas pārvaldība***

Konfigurācijas pārvaldības procesa nodrošināšanā tiek izmantots Subversion (SVN) rīks, kas uzkrāj visu failu sākotnējās un iepriekšējās versijas. Papildus tam daļēji (ne visos reģionos) tiek izmantots Sharepoint rīks, kas nodrošina dokumentācijas uzkrāšanu un iepriekšējo versiju saglabāšanu. Sharepoint rīks nepieciešams visas ar projektu saistītās dokumentācijas izvietojumā vienotā sistēmā, kas pieejama visiem reģioniem. Dokumentu uzkrāšana iespējams var dubultoties ar testētās Clarity programmatūras projektos pievienoto dokumentāciju.

Konfigurācijas pārvaldības procesā nav skaidri definētas procesu plūsmas galveno pieturpunktu atbildīgie cilvēki, piemēram, problēmu izstrādes vai atrisināšanas gadījumā nav noteikta atbildīgā persona par izstrādātā koda pārbaudes veikšanu un pareizību, un atbilstošas dokumentācijas sagatavošanu un izvietojumu vienotajā dokumentu uzkrāšanas sistēmā. Kodu pārbaudei un versiju kontrolei papildus tiek izmantots Code Collaborator rīks. Tas rezultējas ar haotisku dokumentu un dokumentu versiju kontroli un pārvaldību, jo dažādas dokumentu versijas un koda gabali ir izvietoti dažādās vietās, neievērojot loģisku datu uzglabāšanas un struktūras ievērošanas principu.

### ***Kvalitātes pārvaldība***

Kā galvenais un būtiskākais kvalitātes nodrošināšanas nosacījums ir dokumentācijas esamība katrai produkta funkcionalitātei, kā arī dokumentācijai jābūt vienmēr aktualizētai un izvietotai Sharepoint rīkā, taču tas lielo problēmu pieteikumu un steidzamo ziņojumu dēļ daudzkārt uzņēmumā tiek ignorēts vai aizmirsts, kas rezultējas ar neatbilstošas dokumentācijas esamību projektu ietvaros. Ļoti bieži dokumentācija nav vispār izstrādāta līdz ar to problēmu ziņojumu apstrādē, piemēram, nomainoties iesaistītajam personālam (piemēram, personāla slimības gadījumā), nepieciešams veikt sākotnējo analīzi un problēmas reproducēšanu no sākuma. Papildus tam, katrai programmatūras funkcionalitātei jā satur arī testa piemēri un funkciju apraksti, kas ļautu testēt konkrēto situāciju. Tāpat netiek veikta integrācijas testēšana, kas nodrošina stabilas un drošas programmatūras produktu un servisu piegādi klientiem. Daļēji tiek ievērota komentāru iekļaušana kodā, kas aprakstītu katru funkcionalitāti un savstarpējo funkciju mijiedarbību, taču tas neatrisina lielākās problēmas, kas rodas nedisciplēta personāla vai nesakārtotu procesu rezultātā.

Kvalitātes pārvaldības procesā netiek veidots atbilstošs kvalitātes pārvaldības plāns. Papildus tam jā sakārto procesu virzība, piemēram, izstrādājot programmatūras kodu ir jā veic tā pārskats likvidējot nevajadzīgo un savlaicīgi ievērojot iespējamās problēmas. Papildus tam nepieciešams novērtēt un ievērot CMMI labās prakses piemērus kvalitātes nodrošināšanas procesā. Kvalitātes procesa ietvaros nepieciešams papildus dokumentu un iespējams arī sistēmas izstrāde, kas ļautu izseko visa personāla atvaļinājumu grafikam, zināšanu apgabalu novērtējumu katram cilvēkam individuāli.

### ***Piegādātāju izvēle un pārvaldība***

Piegādātāju izvēle notiek pamatojoties uz kādreiz noslēgtajiem līgumiem un vienošanām. Netiek novērtēta piedāvātās cenas atbilstība reālajai situācijai. Netiek analizēta piegādātāju pieredze, piegādes termiņi un piedāvātā pakalpojuma vai servisa kvalitāte, kas rezultējas ar darbinieku un nobeigumā ar iespējamu klientu neapmierinātību, jo netiek piegādāts atbilstošs serviss vai produkts. Nepieciešams definēt specifiskus mērķus šī procesa apgabala sakārtošanā, kā, piemēram, piegādātāju izvēles principi (saimnieciski izdevīgākais vai zemākās cenas piedāvājums), dokumentācijas paraugi (Veicamo darbu vienošanās, līgums), kas ir apstiprināti ar augstāka līmeņa vadību.

### ***Mērījumu veikšana un analīze***

Netiek veikti un uzkrāti jebkādi statistiski mērījumi uz kā pamata būtu iespējams veikt loģiskus rezultātu apkopojumus, izdarīt secinājumus. Ir redzams, ka nepieciešama

automātiska statistikas uzkrāšana, izmantojot vienotu pieeju starp visiem reģioniem, lai nodrošinātu ātru pieredzes uzkrāšanu un secinājumu izdarīšanu balstoties ne vien uz savām, bet arī citu pieļautajām un identificētām kļūdām vai problēmām. Lai to nodrošinātu, nepieciešams veidot ciešu sadarbību un komunikāciju starp visiem reģioniem, kā arī komandas locekļiem. Dažādās situācijās varētu būt nepieciešami dažādi mērījumi, kurus iespējams uzkrāt un pēc tam arī analizēt testējamā Clarity programmatūrā. Mērījumu veikšanu nepieciešams veikt ne vien jaunu projektu izstrādē, bet arī uzturēšanas projektos, kas turpinās ieviešot jaunu terminālu produktu modeļu sērijas klāstu.

### ***Prasību pārvaldība***

Prasību pārvaldības procesā identificēti vairāki trūkumi projektu un inženierijas līmeņos. Nenoliedzami prasību precīza un savlaicīga uzkrāšana sākot jau ar projekta sākuma fāzi ir nozīmīgs ieguvums projekta tālākai attīstībai un veiksmīgai realizācijai. Galvenās identificētās problēmas prasību pārvaldības procesā ir komunikācijas trūkums starp mārketinga aktivitātēs iesaistītajiem cilvēkiem, saiknē ar klientiem un vispārējā prasību aprakstošā dokumentācijas trūkumā. Veidojot atbilstošu komunikācijas attīstības plānu projektu vadības līmenī būtu iespējams pēc iespējas agrāk jebkurā projekta attīstības periodā noteikt sākotnēji definētās vai jaunās prasības attiecībā uz izstrādājamo programmatūru vai piegādāto pakalpojumu un servisu. Prasību izsekošana dažādos ieviešanas līmeņos ir apgrūtināta, līdz ar to nav iespējams precīzi novērtēt gala produkta funkcionalitāti. Gadījumā, ja funkcionalitāte tiek ieviesta pa vairākiem posmiem, nav iespējams precīzi pateikt kāda daļa no vēlamās funkcionalitātes tiek iekļauta katrā posmā, kādi ir ierobežojumi un sagaidāmie rezultāti, jo uzdevumu sākotnējais novērtējums ir pārāk optimistisks, neizvērtējot iespējamos riskus un iepriekšējo zināšanu pieredzes izmantošanu darba pakotņu novērtēšanā līdzīgu projektu ieviešanā.

### ***Risku pārvaldība***

Riski netiek rūpīgi analizēti un uzskaitīti konkrēta projekta ietveros. Nepieciešams veidot risku pārvaldības plānu, kas agrīnā projekta ieviešanas stadijā identificē pēc iespējas vairāk iespējamus ierobežojumus un problēmas, kas var rasties turpmākā projekta izpildes gaitā. Kā būtiska risku pārvaldības sastāvdaļa projektu ietvaros ir resursu pieejamība. Ne vien cilvēku, bet arī tehniski materiālo resursu pieejamība ir jāpārvalda. Cilvēku pieejamību jāplāno vadoties pēc projekta uzdevumu nozīmības, tālāk veidojot atbilstošu atvaļinājuma plānu darbiniekiem. Tehnisko materiālu resursu pieejamība jāplāno uz priekšu, atbilstoši projektā paredzamajiem darba uzdevumiem, piemēram, specifiska platforma –

termināli, uz kuriem paredzēts lietot izstrādājamo programmatūru. Katra platforma satur specifiskus kritērijus, funkcionalitātes nosacījumus katram reģionam atsevišķi, padarot sarežģītāku projektu ieviešanas gaitu un vienotas pieejas izstrādi visā uzņēmumā.

### ***Apmācības***

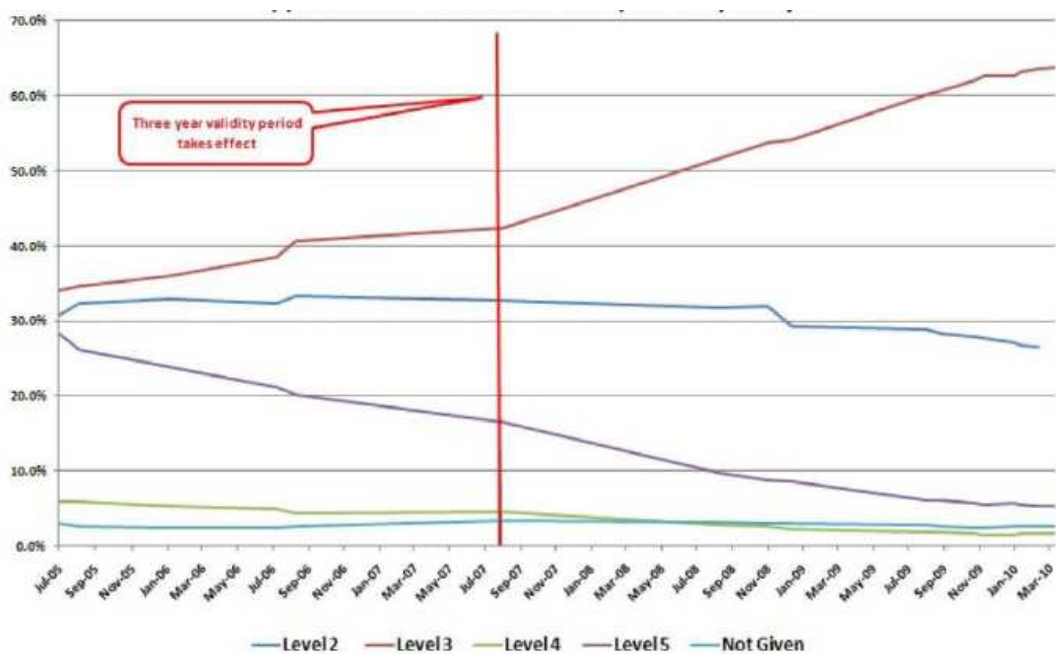
Apmācības netiek nodrošinātas, t.i. apmācības organizē pats personāls tikai pēc pašu interesēm un iniciatīvas. Netiek uzkrāts vēlamo apmācību veids, apjoms un nozīmīgums katram cilvēkam konkrēta projekta ietvaros. Nepieciešams izstrādāt vienotu pieeju apmācību procesa nodrošināšanai, novērtējot esošos zināšanu apgabalus katram cilvēkam individuāli, kā arī nepieciešamās apmācības, kas papildus ietekmē visus pārējo procesu apgabalus. Visaugstāk uzņēmumā jānovērtē tieši cilvēkresursi, līdz ar to apmācības programmas ir atbilstoši arī jāplāno, jo tas ietekmē uzņēmuma turpmāko darbību un attīstības efektivitāti, programmatūras kvalitātes nodrošināšanas un citas saistītajās jomas.

## **2.3. Metodoloģiju izvēles principi un statistika**

Veicot uzņēmuma sākotnējo procesu analīzi, ir definēti galvenie procesu apgabali, novērtēti projektu attīstības virzieni, kā arī izstrādes un uzturēšanas jomu nepieciešamie uzlabojumi. Lai veiktu procesu uzlabojumus, īpaši, ja tas ir saistīts ar citiem procesu apgabaliem, nepieciešama starp reģionāla vienošanās. Tā piemēram, uzlabojot problēmu pieteikšanas, apstrādes un pārvaldības veidu, nepieciešama vienošanās ar Kvalitātes nodrošināšanas departamentu (turpmāk tekstā – QA), kas atrodas Amerikā. Līdz ar to nepieciešams vienoties par testēšanas un reližu gatavošanas procesu, jo liela daļa komponentu izstrāde un labojumu veikšana notiek Latvijas uzņēmumā. No tā secināms, kad papildus testēšana, kā, piemēram, dažādi automātiskie funkcionalitātes testi, kā arī integritātes testi, Latvijā netiek veikti cilvēkresursu trūkuma dēļ. Nepieciešamā integrācijas testēšana un pilnīga funkcionālā testēšana un reližu nodošana klientam tiek veikta Amerikas galvenajā mītnē Arizonā. Papildus nepieciešams izstrādāt automatizētu testēšanas sistēmu, kas nodrošinātu ne vien statistisku testu izpildi, veidojot testu izpildi pēc izvēlētajām funkcijām, bet arī dinamiskā testēšana, kas nodrošinātu arī sistēmas stabilitātes pārbaudi dažādās produktu savstarpēji saistītās funkcionalitātes jomās.

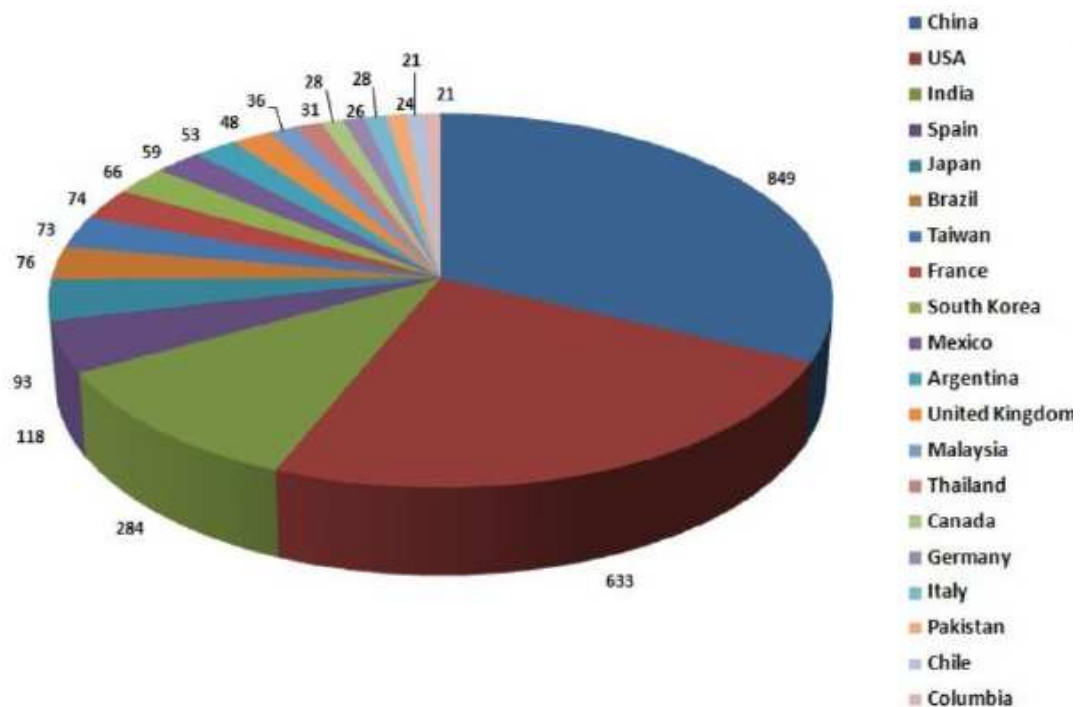
Lai nodrošinātu sistematizētus un korektus procesu uzlabojumus, nepieciešams no katra reģiona izvirzīt, deleģēt vienu vai vairākas personu, kas pārzinātu visus reģionā esošā uzņēmuma pārvaldības un inženierijas procesus un tiktu iesaistīta procesu uzlabošanā.

Lai veiktu sistematizētus procesu uzlabojumus tiek izvēlēta metodoloģija, pēc kuras Amerikā procesu uzlabojumus veikušas daudzas kompānijas. Kā populārākā procesuālo uzlabojumu metode ir CMMI. Zemāk esošajā attēlā iespējams aplūkot CMMI popularitāti pa valstīm. Līdz ar to veicot metodoloģiju izvēli, galvenais novērtējums tiek likts uz mātes uzņēmuma galvenajām prioritātēm un nepieciešamībām, analizējot datus galvenokārt pēc Amerikas statistikas sniegtajiem pārskatiem.



Attēls Nr.2.3.1. „CMMI brieduma līmeņu popularitātes novērtējums” [3]

No augstāk atspoguļotās informācijas secināms, ka trīs gadu laikā CMMI standarta izmantošanas skaits ir būtiski pieaudzis. Attīstības tendences un novērtējums veikts piecu gadu garumā. Lielākoties kompānijas izvēlas ieviest trešā līmeņa procesu uzlabojumu labās prakses piemērus, kas nodrošina arī pamata darbības un veiksmīgu projektu attīstību balstoties uz definētiem, labi aprakstītiem procesiem, kas rezultējas ar augstāku klientu apmierinātības līmeni projektu ieviešanas procesos.



Attēls Nr.2.3.2. „CMMI izmantošana dažādās pasaules valstīs” [3]

Austāk atspoguļotajā attēlā redzams CMMI popularitātes novērtējums kompāniju skaita ziņā dažādās valstīs. Statistikas mērījums veikts Amerikas Savienotajās valstīs 2009.gada decembrī. Kopš 2009.gada novembra CMMI pielietojuma pieaugums mērāms: Ķīna – 14 kompānijas, ASV – 9 kompānijas, Indija – 4 kompānijas.

Veicot procesu analīzi Latvijas uzņēmumā un definējot procesu apgabalus un pielietotos rīkus dažādu problēmu risināšanā, tiek secināts, ka testējamā programmatūra Clarity ir atbilstoša un pielietojama, jo pārsedz lielu daļu no nepieciešamajām jomām, galvenokārt integrētas projektu pārvaldības jomās, saistībā ar MS Project rīku, kas iepriekš ir izmantots, kā arī kā atbilstošs finansu pārvaldības rīks, kas nodrošina atbilstošu pārraudzību un kontroli projektu ietvaros. Šādu funkcionalitāti nenodrošina tādi populāri rīki kā Jira, taču nākotnē iespējams nepieciešams izvērtēt citus projektu pārvaldības rīkus, kas būtu integrēti ar finansu pārvaldības procesu apgabalu. Finanšu pārvaldībai jābūt integrētai ar projektu pārvaldības rīku, jo tādā veidā iespējams izsekot izlietoto resursu daudzumam kāda projekta ietvaros. Tā, piemēram, jauna projekta ietvaros, tiek definēti uzdevumi un noteikts to izpildes termiņš. Tādā veidā iespējams, pēc sākotnējā novērtējuma, noteikt aptuvenu projekta izpildes laiku, kā arī nepieciešamās projekta izmaksas. Ja projekta izmaksas acīmredzami pārsniedz plānoto, nepieciešams projektu apstiprināšanas komisijā iesniegt nepieciešamos labojumus. Veicot projekta sākotnējo novērtējumu bija redzams, ka izmaiņu veikšana projektos ir ļoti

apgrūtināta, jo, piemēram, sistēmas komponentes tiek izstrādātas sākot ar operētājsistēmas pielāgošanu konkrētajai platformai līmeni, izstrādājot atbilstošus dziņus, kas savstarpēji mijiedarbojas ar servisiem un produkta komponentēm. Tālāk tiek izstrādātas atbilstošas klases un funkciju grupas, kuras tālāk izmantos konkrētas aplikācijas. Papildus tam, dažādi klienti izmanto arī savas veidotas aplikācijas vai papildina jau esošās izstrādātās finansu aplikācijas, kur līdz ar to papildus nepieciešams izstrādāt papildus funkcionalitāti kāda produkta veidošanas procesa ietvaros. Līdz ar to, ja, piemēram, tiek veiktas kādas izmaiņas kaut vai pieejas maiņā kā kāda ierīce komunicē ar kādu citu funkciju, piemēram, komunikāciju nodrošināšana vai maksājumu kartes nolasīšanas un kodēšanas, kriptēšanas kārtība un veids, nepieciešams veikt izmaiņas saistītajās komponentēs labākajā gadījumā, taču sliktākajā – veikt izmaiņas daudzos arhitektūras līmeņos. Lai veiktu izmaiņas kādos programmatūras līmeņos un vispār visā funkcionalitātē, pēc apzinātajām problēmu jomām, tiek noteikts:

- izstrādāt atbilstošus vienību testus, kas nodrošinātu atbilstošu funkcionalitātes ieviešanu, balstoties uz definētajām prasībām
- veidot atbilstošu komunikāciju ar klientu, pasūtītāju, lai definētu precīzi visas prasības
- tāpat nepieciešams veidot atbilstošu komunikāciju ar klientu, lai izprastu pieteikto problēmu ziņojumu pieteikumus. Problēmu pieteikumu sākuma fāzē nepieciešams veikt pieteiktās problēmas atkārtošanu, lai spētu identificēt problēmas atrašanās apgabalu komplicētajā sistēmā. Nākošajos produktos, lai problēmu pieteikumu apgabalus spētu identificēt ātrāk ir nepieciešams iestrādāt problēmu žurnālēšanas pasākumus, funkcionalitāti
- ātri reaģēt uz pieteiktajiem un aktuālākajiem problēmu ziņojumiem. Nepieciešams izveidot vienotu problēmu pieteikumu reģistru, nodrošinot to kvalitatīvu aizpildi un detalizētu problēmu pieteikumu aprakstu
- jau iepriekš spēt paredzēt aktuālākās problēmas, veikt atbilstošu analīzi un novērtējumu, veikt preventīvos pasākumus
- izstrādāt un aprakstīt atbilstošu funkcionalitāti dokumentācijas veidā, papildus tam uzlabojot problēmu pieteikumu žurnālēšanas principus

Izvērtējot katra projekta tipu, sarežģītību un inovāciju ieviešanas nepieciešamību tiek noteikta izvēlētais projektu vadības pieeja. Vienkāršu projektu gadījumā, kā arī projektu vadītāja pieredzes trūkuma gadījumā tiek izmantota tradicionālā pieeja, kur sākotnējās prasības praktiski nemainās. Tas secināms arī gadījumā, ja šāda līdzīga projekta izstrāde jau

ir veikta, piemēram, līdzīgas funkcionalitātes ieviešana kādā produktu līnijā pēc klienta vēlmēm un prasībām, kas pieejama jau citos produktos. Gadījumā, ja projektā nepieciešams ieviest jaunas tehnoloģijas, secināms, ka sākotnējās prasības vēl tiks papildinātas un sākuma fāzē ir daudzas neskaidrības, projekti tiks ieviesti, izmantojot iteratīvu spējo projektu vadības metodi, kas nosaka arī komandas vadības pieeju, organizacionālās pieejas, taču paralēli definējot un ievērojot arī procesu apgabalus un standartus, kas nepieciešami, lai nodrošinātu koordinētu projektu tālāku virzību un attīstību.

### 3. CMMI METODOĻĢIJA UN SAISTĪTIE PROCESI

CMMI no angļu valodas tulkojumā (Capability Maturity Model Integration) skaidrojams kā „neizmantoto spēju brieduma modeļa integrācija”. Tā ir saistība starp vairākiem ieviešanas procesa modeļiem. Tāpat arī šī modeļu integrācija apraksta labās prakses piemērus konkrētā procesa līmenī, kā arī parāda katra līmeņa savstarpējās atšķirības, kas iepriekšējā līmenī nav minētas vai sasniegtas. Tas ir standarts, kas ietver sevī definētus procesu uzlabojumus, kas organizācijā pēc tam nodrošina efektīvu procesu vadību un pielietošanu. CMMI ir ietvars vai modeļu viens, lai uzlabotu biznesa procesus kopumā, un ir galvenokārt pielietots un pieprasīts ASV programmatūras izstrādes kompānijās. Latvijā CMMI ir apguvusi arī kompānija „Exigen”, kas ir saņēmusi arī atbilstošu sertifikātu, ievērojot noteiktu sertifikāta izsniegšanas periodu un pārbaudes noteikumus.

CMMI galvenokārt ir veidots kā modelis, lai īstenotu procesu uzlabošanas sistēmas. Procesu uzlabošana tiek analizēta apkārt notiekošo un tiek ieviesta tādā pašā veidā kā veidojot kādu algoritmu vai būvējot kādu māju – soli pa solim. Definējot noteiktus procesu apgabalus un rīkus, kas tiek vai tiks izmantoti, kā arī paralēli analizējot visus saistītos procesus, lai sasniegtu izvirzīto biznesa mērķi, kur iespējams sasniegt vēlā ņemamus rezultātus efektivitātes un kvalitātes uzlabošanā produktu ieviešanas un uzturēšanas ciklos.

CMMI iedala trīs veidos [4]:

- CMMI priekš izstrādes (no angļu val. - CMMI for Development)  
(izstrādājot produktus un sarežģītus servissus)
- CMMI priekš iegūšanas (no angļu val. - CMMI for Acquisition)  
(apgūstot produktus un servissus no citiem)
- CMMI priekš servisiem (no angļu val. - CMMI for Services)  
(piedāvājot un piegādājot servissus).

Papildus tam uzņēmumā iespējams ieviest People CMM, kas definē kādās savstarpēji saistītās procesu jomās nepieciešami uzlabojumi, mērījumi, pārvaldība, kā arī kontrole un uzraudzība, atbilstoši cilvēkresursu attīstībai un to noteiktajām prioritātēm uzņēmumā, kas šajā darbā netiek papildus apskatīts. Iepazīstoties ar „CMMI for Development” galvenajām nostādnēm tiek secināts, ka šī aprakstošā metodoloģija un labo prakšu krājums, ņemot vērā noteiktā standarta ieviešanas nepieciešamību, uzskatāms par atbilstošāko programmatūras izstrādes un uzturēšanas projektu īstenošanas procesa nodrošināšanā.

Nosacījums CMMI ieviešanā kādā organizācijā ir jau eksistējoši iekšēji un ārēji standarti, procesi vai procesu kopums kā procedūras, līdz ar to nosakot kā šiem procesiem

būtu nepieciešams mijiedarboties. CMMI mērķis ir uzlabot programmatūras izstrādes procesus un kopumā visas uzņēmuma darbības kvalitāti, veikspēju un efektivitāti, balstoties uz iepriekš izteikto nosacījumu. CMMI izstrādāts, lai palīdzētu organizācijām uzlabot savu spēju nepārtraukti un paredzamā veidā piegādāt produktus un servisu saviem klientiem, tieši to, ko viņi vēlas, kad un par kādu samaksu viņi vēlas. Tiek uzlabota komunikācija ar klientu, tiešā veidā pieaug arī klienta apmierinātība, kā arī laika gaitā programmatūras izstrādes procesi tiek uzlaboti, kā piemēram, mērījumi un statistika par darba uzdevumu novērtējuma kvalitāti un atbilstību reāli nostrādāto stundu daudzumam kādam noteiktam uzdevumam. CMMI ir procesu uzlabošanas modelis no kura uzņēmums veido konkrētajai situācijai atbilstošas prioritātes un veic unikālus procesu uzlabošanas risinājumus.

### **3.1. CMMI – 1.līmenis**

Pirmajā brieduma līmenī procesi parasti ir haotiski un neskaidri. Organizācija parasti nespēj nodrošināt stabilu apkārtējo un procesuālo vidi programmatūras izstrādē iesaistītajiem darbiniekiem. Veiksme šādās organizācijās balstās nevis uz pierādītiem un definētiem procesiem, bet gan uz atsevišķu cilvēku pieredzi un vēlmi kaut ko uzlabot un attīstīt. Pirmajā līmenī uzņēmums visbiežāk spēj izstrādāt produktus un piegādāt servisu, kas strādā, taču parasti pārsniedzot atvēlēto budžetu un projektu termiņus.

Pirmais organizācijas līmenis parasti tiek raksturots ar tendenci visu pārvērtēt, aizmirstot par procesiem, piemēram, kritiskā ceļa programmatūras izstrādes uzdevumos. Līdz ar to arī nav iespējams veikt atkārtotu un identisku darbību, kas ir veikta pirms tam, gadījumā, ja rezultāts ir bijis veiksmīgs un klients ir apmierināts ar iepriekš definētām prioritātēm (piemēram, izmaksas, laiks un produkta kvalitāte).

### **3.2. CMMI – 2.līmenis**

Otrajā brieduma līmenī organizācija ir sasniegusi specifiskus un vispārējos mērķus no CMMI otrā līmeņa procesu apgabaliem. Citiem vārdiem sakot, projektos tagad ir pārvaldītas, fiksētas prasības un procesi tiek plānoti, izpildīti, mērīti un atbilstoši arī kontrolēti. Sākot ar šo līmeni tiek ievērota definēto procesu disciplīna, t.i. iesaistītās puses vienojas par darbu izpildes kārtību – procesu kopumā un ievēro to turpmākajā darbā. Šāda disciplīnas ievērošana nodrošina to, kad stresa situācijā tiek ievērots un pielietots apstiprinātais process. Kad šāda prakse tiek ievērota, projekti tiek izpildīti un pārvaldīti atbilstoši noteiktajiem dokumentētajiem un fiksētajiem plāniem. CMMI otrajā līmenī prasības, procesi, gala produktu izstrāde un servisi tiek pārvaldīti, produktu piegādes un to termiņi ir caurskatāmi un

tos arī ir iespējams pārvaldīt, jo satur definētus plāna pieturpunktus (no angļu val. - *milestones*). Visas izmaiņas, plāni un saistītā dokumentācija līdz ar to arī tiek apstiprināta starp visām ieinteresētajām pusēm pēc vajadzības un pieprasījuma. Tādā veidā tiek nodrošināta atbilstoša komunikācija starp visām iesaistītajām pusēm un gala produkts satur visas nepieciešamās komponentes, līdz ar to sagaidot klienta pozitīvu atsauksmi produkta izstrādes un beigu piegādes ciklā.

Galvenie CMMI otrā līmeņa galvenie mērķi (*turpmāk tekstā - GG*) un izvirzītie prakses piemēri un mērķi (*turpmāk tekstā - GP*):

GG L2.: Veidot pārvaldītu procesu [4]:

GP L2.1.: Nodrošināt izveidoto organizācijas politiku

GP L2.2.: Plānot procesus

GP L2.3.: Piešķirt resursus, nodrošināt to pieejamību

GP L2.4.: Deleģēt atbildības jomas

GP L2.5.: Nodrošināt cilvēkresursu apmācību

GP L2.6.: Pārvaldīt konfigurāciju

GP L2.7.: Identificēt un iesaistīt visas ieinteresētās puses

GP L2.8.: Pārraudzīt un kontrolēt procesus

GP L2.9.: Objektīvi novērtēt informāciju, stingri ievērot procesus

GP L2.10.: Pārskatīt statusu ar augstāka līmeņa vadību

### **3.2.1. Projektu plānošana**

Projektu plānošanas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi (*turpmāk tekstā – SG*), kā arī konkrētas prakses (*turpmāk tekstā – SP*) [4].

SG 1.: Veidot novērtējumus

SP 1.1.: Novērtēt projekta mērķi

SP 1.2.: Veidot novērtējumu darbu pakotnēm, uzdevumiem un to atribūtiem

SP 1.3.: Definēt projekta dzīves ciklu

SP 1.4.: Noteikt darbietilpības un izmaksu novērtējumu

SG 2.: Izstrādāt projekta plānu

SP 2.1.: Veidot budžetu un sastādīt grafiku

SP 2.2.: Identificēt projekta riskus

SP 2.3.: Izstrādāt datu pārvaldības plānu

SP 2.4.: Izstrādāt projekta resursu pārvaldības plānu

- SP 2.5.: Izstrādāt zināšanu un prasmju plānu
- SP 2.6.: Plānot ieinteresēto pušu līdzdalību projekta gaitā
- SP 2.7.: Nodrošināt plāna atbilstību reālajai situācijai

SG 3.: Nodrošināt saistību ar plānu

- SP 3.1.: Pārskatīt plānus, kas ietekmē projektu
- SP 3.2.: Līdzsvarot un saskaņot darbu un resursu līmeņus
- SP 3.3.: Nodrošināt plānu savstarpēju saistību

### ***3.2.2. Projektu uzraudzība un kontrole***

Projektu uzraudzības un kontroles procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Pārraudzīt projektu attiecībā pret projekta plānu

- SP 1.1.: Pārraudzīt projekta plānošanas parametrus
- SP 1.2.: Pārraudzīt saistības
- SP 1.3.: Pārraudzīt projekta riskus
- SP 1.4.: Pārraudzīt datu pārvaldību
- SP 1.5.: Pārraudzīt ieinteresēto pušu līdzdalību un iesaisti
- SP 1.6.: Veikt progresu pārskatus
- SP 1.7.: Veikt pieturpunktu pārskatus

SG 2.: Pārvaldīt ietekmējošo darbību slēgšanu

- SP 2.1.: Analizēt problēmas
- SP 2.2.: Izstrādāt un pielietot problēmu atrisinājumu
- SP 2.3.: Pārvaldīt atrisinājumu pielietojumu

### ***3.2.3. Konfigurācijas pārvaldība***

Konfigurācijas pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Veidot projekta bāzlīniju

- SP 1.1.: Identificēt konfigurācijas priekšmetus, vienumus
- SP 1.2.: Veidot, uzturēt un attīstīt konfigurācijas pārvaldības sistēmu
- SP 1.3.: Veidot relīzes bāzlīniju

SG 2.: Izsekot un kontrolēt izmaiņas

- SP 2.1.: Izsekot izmaiņu pieprasījumus

SP 2.2.: Kontrolēt konfigurācijas priekšmetus, vienumus

SG 3.: Nodrošināt integritāti

SP 3.1. Veidot konfigurācijas pārvaldības ierakstus

SP 3.2. Veikt konfigurācijas auditus

#### ***3.2.4. Produktu un procesu kvalitātes pārvaldība***

Produktu un procesu kvalitātes pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Objektīvi novērtēt procesus un darbu pakotnes, produktus

SP 1.1.: Objektīvi novērtēt procesus

SP 1.2.: Objektīvi novērtēt darbu pakotnes, produktus un servisu

SG 2.: Nodrošināt uzdevumu izpratnes spēju

SP 2.1.: Pārbaudīt neievēroto problēmu atrisināšanu

SP 2.2.: Veidot ierakstus

#### ***3.2.5. Piegādātāju vienošanās pārvaldība***

Piegādātāju vienošanās pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Nodrošināt piegādātāju vienošanās

SP 1.1.: Noteikt piegādātāju iegūšanas veidu

SP 1.2.: Izvēlēties piegādātājus

SP 1.3.: Nodrošināt vienošanās ar piegādātājiem

SG 2.: Izpildīt piegādātāju vienošanās nosacījumus

SP 2.1.: Noslēgt piegādātāju vienošanos

SP 2.2.: Pārraudzīt izvēlēto piegādātāju procesus

SP 2.3.: Novērtēt izvēlēto piegādātāju produktus vai pakalpojumus

SP 2.4.: Pieņemt un apstiprināt iegūtos produktus vai pakalpojumus

SP 2.5.: Pārejas produkti vai pakalpojumi

#### ***3.2.6. Mērīšana un analīze***

Mērīšanas un analīzes procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

- SG 1.: Noteikt mērīšanas un analīzes aktivitātes
- SP 1.1.: Veidot mērījumu uzdevumus
  - SP 1.2.: Noteikt mērījumus
  - SP 1.3.: Noteikt datu uzkrāšanas un glabāšanas procedūras
  - SP 1.4.: Noteikt analīzes veikšanas procedūras
- SG 2.: Nodrošināt mērījumu rezultātus
- SP 2.1.: Uzkrāt mērījumu datus
  - SP 2.2.: Analizēt mērījumu datus
  - SP 2.3.: Saglabāt datus un rezultātus
  - SP 2.4.: Apspriest rezultātus

### ***3.2.7. Prasību pārvaldība***

Mērīšanas un analīzes procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses.

- SG 1.: Pārvaldīt prasības
- SP 1.1.: Iegūt sapratni par prasībām
  - SP 1.2.: Iegūt uzticēšanos jautājumos par prasību specificēšanu
  - SP 1.3.: Pārvaldīt prasību izmaiņas
  - SP 1.4.: Nodrošināt abpusēju prasību caurskatāmību, izsekojamību
  - SP 1.5.: Identificēt nesaskaņas starp projekta uzdevumiem un prasībām

## **3.3. CMMI – 3.līmenis**

CMMI trešajā brieduma līmenī organizācija ir sasniegusi specifiskus un vispārējus mērķus, kas tajā pašā laikā attiecas arī uz otro un trešo brieduma līmeni. Trešajā līmenī procesi ir labi aprakstīti un arī saprotami. Tie ir aprakstīti iekšējās procedūrās un standartos, izmantojot dažādas metodes un rīkus. Galvenā atšķirība starp otro un trešo līmeni ir tajā, ka trešajā līmenī procedūras, standarti un procesu apraksti ir vērsti uz visu organizāciju kopumā, jo otrais līmenis pieļauj atšķirības, piemēram, izmantoto rīku izvēlē dažādos reģionos. Tātad sasniedzot trešo līmeni organizācija sāk izmantot vienotus mehānismus un procesus mērķu sasniegšanā, kas arī saistītas ar otrā līmeņa definētajām procedūrām un procesiem. Rezultātā procesi ir vienoti, apstiprināti, ieviesti un pielietoti visā uzņēmumā, pieļaujot nelielas atšķirības noteikto vadlīniju ieviešanā.

Tāpat vēl viena būtiska atšķirība starp otro un trešo brieduma līmeni procesos ir to aprakstā. Trešajā līmenī procesu apraksti ir daudz detalizētāki un uzdevumu sīkāk aprakstošāki, iedziļinoties vairāk detaļās. Tāpat procesi ir labāk pārvaldīti, jo tiek sekots principam – tiek ievērota un jau pārskatīta savstarpēja saikne starp procesiem un aktivitātēm, kas var ietekmēt galvenos procesus, kā arī tiek veikti, izdomāti un apstiprināti procesu un uzdevumu mērījumi – statistika.

### ***3.3.1. Organizācijas procesu fokuss***

Organizācijas procesu fokusa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Noteikt procesu uzlabošanas iespējas

SP 1.1.: Izveidot organizatorisko procesu vajadzības

SP 1.2.: Novērtēt organizācijas procesus

SP 1.3.: Identificēt organizācijas procesu uzlabojumus

SG 2.: Plānot un ieviest procesu uzlabošanas pasākumus

SP 2.1.: Izveidot procesu rīcības plānu

SP 2.2.: Ieviest procesu rīcības plānu

SG 3.: Ieviest organizatorisko procesu vērtību un iekļaut tajā gūto pieredzi

SP 3.1.: Ieviest organizatorisko procesu vērtību

SP 3.2.: Ieviest standarta procesus

SP 3.3.: Pārraudzīt ieviešanu

SP 3.4.: Iekļaut procesu saistīto un gūto pieredzi organizatorisko procesu vienumā

### ***3.3.2. Organizācijas procesu definīcija***

Organizācijas procesu definīcijas galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izveidot organizatorisko procesu vērtības

SP 1.1.: Izveidot standarta procesus

SP 1.2.: Izveidot dzīves cikla modeļa aprakstus

SP 1.3.: Izveidot kritērijus un vadlīnijas

SP 1.4.: Izveidot organizācijas mērījumu krātuvi

SP 1.5.: Izveidot organizācijas procesa vērtības bibliotēku

### **3.3.3. Organizācijas apmācība**

Organizācijas apmācības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izveidot apmācības iespējas organizācijā

SP 1.1.: Noteikt stratēģisko apmācību vajadzības

SP 1.2.: Noteikt kura apmācību vajadzība ir organizācijas atbildībā

SP 1.3.: Izveidot organizācijas taktisko apmācību plānu

SP 1.4.: Veidot apmācības iespējas

SG 2.: Sniegt nepieciešamo apmācību

SP 2.1.: Piegādāt apmācības

SP 2.2.: Izveidot apmācību ierakstus

SP 2.3.: Novērtēt izglītības efektivitāti

### **3.3.4. Integrēta projektu pārvaldība**

Integrētas projektu pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izmantot projekta noteiktu procedūru

SP 1.1.: Izveidot projekta noteiktu procedūru

SP 1.2. Izmantot organizatorisko procesu vērtības plānošanas projekta aktivitātēm

SP 1.3. Izveidot projekta darba vidi

SP 1.4. Integrēt plānus

SP 1.5. Pārvaldīt projektu, izmantojot integrētus plānus

SP 1.6. Papildināt organizācijas procesa vērtības

SG 2.: Koordinēt un sadarboties ar visām ieinteresētajām pusēm

SP 2.1. Pārvaldīt ieinteresēto pušu iesaisti

SP 2.2. Pārvaldīt visas atkarības

SP 2.2. Risināt koordinācijas problēmas

### **3.3.5. Risku pārvaldība**

Risku pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Sagatavoties risku pārvaldībai

SP 1.1.: Noteikt riska cēloni, avotus un kategorijas

SP 1.2.: Definēt riska parametrus

SP 1.3.: Veidot riska pārvaldības stratēģiju

SG 2.: Identificēt un analizēt riskus

SP 2.1.: Risku noteikšana

SP 2.2.: Veikt risku novērtēšanu, grupēšanu un prioritizēšanu

SG 3.: Mazināt riskus

SP 3.1.: Izstrādāt riska mazināšanas plānu

SP 3.2.: Īstenot riska mazināšanas plānu

### **3.3.6. Lēmumu analīze un atrisināšana**

Lēmumu analīzes un atrisināšanas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Novērtēt alternatīvas

SP 1.1.: Izveidot pamatnostādnes attiecībā uz lēmumu analīzi

SP 1.2.: Izveidot novērtēšanas kritērijus

SP 1.3.: Identificēt alternatīvus risinājumus

SP 1.4.: Izvēlēties novērtēšanas metodes

SP 1.5.: Novērtēt alternatīvas

SP 1.6.: Izvēlēties risinājumus

### **3.3.7. Prasību pārvaldība**

Prasību pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izstrādāt klientu prasības

SP 1.1.: Noskaidrot prasības

SP 1.2.: Izstrādāt klienta prasības

SG 2.: Izstrādāt produkta prasības

SP 2.1.: Izveidot produktu un produktu sastāvdaļu prasības

SP 2.2.: Piešķirt produkta komponentu prasības

SP 2.3.: Identificēt saskarnes prasības

SG 3.: Analizēt un apstiprināt prasības

SP 3.1.: Izveidot operatīvas koncepcijas un scenārijus

SP 3.2.: Izveidot nepieciešamās funkcionalitātes definīciju

SP 3.3.: Analizēt prasības

SP 3.4.: Analizēt prasības, lai sasniegtu līdzsvaru

SP 3.5.: Pārbaudīt prasības

### **3.3.8. Tehniskie risinājumi**

Tehnisko risinājumu procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izvēlēties produkta komponentu risinājumus

SP 1.1.: Izstrādāt alternatīvus risinājumus un atlases kritērijus

SP 1.2.: Izvēlēties produkta komponentu risinājumus

SG 2.: Izstrādāt dizainu

SP 2.1.: Veidot produktu un produktu komponentu dizainu

SP 2.2.: Izveidot tehnisko datu pakotni

SP 2.3.: Izveidot saskarni izmantojot definētus kritērijus

SG 3.: Īstenot produkta dizainu

SP 3.1.: Īstenot dizainu

SP 3.2.: Izveidot produkta atbalsta dokumentāciju

### **3.3.9. Produktu integrācija**

Produktu integrācijas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Sagatavoties produktu integrācijai

SP 1.1.: Noteikt integrācijas secību

SP 1.2.: Izveidot produktu integrācijas vidi

SP 1.3.: Izveidot produktu integrācijas kārtību un kritērijus

SG 2.: Nodrošināt interfeisa savietojamību

SP 2.1.: Pārskatīt interfeisa aprakstu un atbilstību

SP 2.2.: Pārvaldīt interfeisu

SG 3.: Sagatavot produktu komponentes un piegādāt produktu

SP 3.1.: Sagatavot produktu komponentes integrācijas nodrošināšanai

SP 3.2.: Sagatavot produkta sastāvdaļas

SP 3.3.: Novērtēt sagatavotā produkta sastāvdaļas

SP 3.4.: Iepakot un piegādāt produktu vai tā sastāvdaļu

### **3.3.10. Verifikācija**

Verifikācijas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Sagatavoties verifikācijai

SP 1.1.: Izvēlēties strādājošus produktus verifikācijai

SP 1.2.: Izveidot darba vidi verifikācijas veikšanai

SP 1.3.: Izveidot verifikācijas procedūras un kritērijus to ieviešanai un nodrošināšanai

SG 2.: Veikt salīdzinošo verifikāciju

SG 2.1.: Sagatavoties salīdzinošajai pārbaudei

SG 2.2.: Izpildīt salīdzinošo pārbaudi

SG 2.3.: Analizēt salīdzinošās pārbaudes datus

SG 3.: Verificēt izvēlētos strādājošos produktus

SG 3.1.: Veikt verifikāciju

SG 3.2.: Analizēt verifikācijas rezultātus

### **3.3.11. Validācija**

Validācijas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Sagatavoties validācijai

SP 1.1.: Izvēlēties produktus validācijai

SP 1.2.: Izveidot darba vidi validācijas veikšanai

SP 1.3.: Izveidot validācijas procedūras un kritērijus to ieviešanai un nodrošināšanai

SG 2.: Pārbaudīt produktu un produkta komponentes

SP 2.1.: Veikt validāciju

SP 2.2.: Analizēt validācijas rezultātus

## **3.4. CMMI – 4.līmenis**

Ceturtajā CMMI brieduma līmenī uzņēmums ir sasniedzis visus specifiskos procesu apgabalu mērķus, kas attiecas arī uz otro, trešo un ceturto līmeni, kā arī vispārējos mērķus. Ceturtajā līmenī apakšprocesu ir definēti un aprakstīti tādā līmenī, lai pārklātu visus procesu apgabalus un nodrošinātu to sekmīgu izpildi. Šie definētie apakšprocesu tiek kontrolēti izmantojot statistiskus un kvantitatīvus mērījumus un tehnikas, piemēram, izmantojot papildus programmatūras pielikumus problēmu pieteikumu uzskaites un dažādu statistisko mērījumu atspoguļošanai vienotā sistēmā visa uzņēmumā. Kā viens no mērījumiem varētu būt aizvērto un atvērto problēmu pieteikumu skaits izstrādātajā gala produktā (gan aparatūras, gan programmatūras līmenī). Mērījumu sekošana un savlaicīga reakcija uz dažādām izmaiņām ļauj uzlabot procesu un produktu kvalitāti un to pārvaldību. Šajā līmenī jānosaka kvantitatīvus mērķi, kas balstās uz klienta, gala lietotāja un kopumā visa uzņēmuma vēlmēm un definētām prioritātēm, kas arī ir jāapstiprina ar visām iesaistītajām pusēm. Gadījumā, ja ir radušies kādi īpaši cēloņi kas ļauj interpretēt procesu savādāk vai ir izveidojusies kāda procesu variācija, tiek novērstas visas nepilnības un neskaidrības, kā arī process tiek pārskatīts un uzlabots, lai pārpratumi un kļūdas vairāk neatkārtotos.

Kvalitātes un procesu veikuma mērījumi ir iekļauti vienotajā uzņēmuma mērījumu sistēmā vai kādā citā labi pārvaldāmā veidā, lai īstenotu uz faktiem balstītu lēmumu pieņemšanu nākotnē.

Nozīmīgākā atšķirība starp trešo un ceturto brieduma līmeni ir procesu virzības paredzamība. Ceturtajā līmenī procesu virzība ir kontrolēta, izmantojot statistiskas un kvantitatīvas metodes, kur trešajā līmenī procesi ir tikai kvantitatīvi paredzami un nojaušami.

### ***3.4.1. Kvantitatīva projektu pārvaldība***

Kvantitatīvas projektu pārvaldības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Kvantitatīvi pārvaldīt projektu

SP 1.1.: Noteikt projekta mērķi

SP 1.2.: Aprakstīt definētos procesus

SP 1.3.: Izvēlēties apakšprocesus, kas tiks statistiski pārvaldīti

SP 1.4.: Pārvaldīt projekta izpildi

SG 2.: Statistiski pārvaldīt projektu apakšprocesus

SP 2.1.: Izvēlēties mērījumus un analītiskas metodes

SP 2.2.: Pieteikt statistikas metodes, lai varētu saprast variācijas

SP 2.3.: Pārraudzīt apakšprocesu un vērtību izpildi

SP 2.4. Saglabāt statistikas pārvaldības datus

### ***3.4.2. Organizācijas procesu virzība un attīstība***

Organizācijas virzības un attīstības procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Nodrošināt veikspējas bāzlīnijas un modeļus

SP 1.1.: Izvēlēties procesus

SP 1.2.: Veidot procesu veikspējas mērījumus

SP 1.3.: Noteikt kvalitātes un procesa veikspējas mērķus

SP 1.4.: Veidot procesu veikspējas bāzlīniju

SP 1.5.: Veidot procesu veikspējas modeļus

## **3.5. CMMI – 5.līmenis**

CMMI piektajā brieduma līmenī uzņēmums ir sasniedzis visus otrā, trešā un ceturta līmeņa specifiskos, kā arī otrā un trešā līmeņa vispārīgos mērķus, kas izvirzīti un apstiprināti katrā no līmeņiem. Procesi nepārtraukti attīstās un tiek uzlaboti balstoties uz sapratni par vienotām problēmām un to cēloņiem. Piektais līmenis tiek balstīts uz procesu attīstību, izmantojot inovatīvas metodes vai tehnoloģijas, kā arī atspoguļo precīzi mainīgās biznesa

prioritātes un mērķus, ja tādas rodas. Līdz ar to, spējot sekot definētajiem procesiem, iespējams pārvaldīt arī procesu uzlabošanu, jo tas ir detalizēti aprakstīts. Izstrādātie procesu uzlabojumi tiek mērīti atbilstoši iepriekšējām un esošajām izmaiņām. Gan definētie, gan uzņēmuma noteikto standarta procesu kopums ir jāmēra veikto uzlabojumu un aktivitāšu ietvaros. Piekto līmeni raksturo uzņēmuma spēja ātri reaģēt uz izmaiņām, kā arī tiek meklētas iespējas attīstīt un dalīties uzņēmuma pieredzē un zināšanās. Uzlabojumu veikšana ir daļa no katra iesaistītā cilvēka darba pienākumiem, kas rezultējas ar nepārtrauktu procesu uzlabojumu cikliem visu laiku. Galvenā atšķirība starp ceturto un piekto līmeni ir tajā, kad ceturtajā līmenī procesi ir saistīti ar dažādām opcijām un procesu variācijām, kur tiek nodrošināta uzkrātā automatizētas pieejas statistika, kas rezultējas ar procesu paredzamību un caurskatāmību. Lai gan procesi ceturtajā līmenī var būt paredzami, tas var nebūt pietiekami, lai sasniegtu nosprausto mērķi. Piektajā brieduma līmenī procesi tiek pastāvīgi attīstīti balstoties uz iepriekšēju pieredzi un veiktajiem mērījumiem, lai uzlabotu procesu veikspēju un nodrošinātu to veiksmīgu attīstību.

### ***3.5.1. Cēloņu analīze un atrisināšana***

Cēloņu analīzes un atrisināšanas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Noteikt defektu rašanās iemeslus

SP 1.1.: Izvēlēties defekta datus analīzes veikšanai

SP 1.2.: Analizēt cēloņus

SG 2.: Uzsākt defektu cēloņu atrisināšanas procedūru

SP 2.1.: Īstenot veiktos priekšlikumus

SP 2.2.: Novērtēt izmaiņu ietekmi

SP 2.3.: Ierakstīt, uzglabāt datus

### **3.5.2. Organizācijas inovācijas un to ieviešana**

Organizācijas inovācijas un to ieviešanas procesa galvenā mērķa un prakses sasniegšanai tiek izvirzīti specifiski mērķi, kā arī konkrētas prakses [4].

SG 1.: Izvēlēties uzlabojumus

SP 1.1.: Apkopot un analizēt nepieciešamos priekšlikumus uzlabojumu veikšanai

SP 1.2.: Identificēt un analizēt inovācijas

SP 1.3.: Vadīt uzlabojumus

SP 1.4.: Izvēlēties uzlabojumus ieviešanai

SG 2.: Ieviest uzlabojumus

SP 2.1.: Plānot ieviešanas jomas

SP 2.2.: Pārvaldīt ieviešanu

SP 2.3.: Novērtēt mērot uzlabojumu ietekmi

## 4. SPĒJĀ PROGRAMMATŪRAS IZSTRĀDE

Spējā programmatūras izstrāde (no angļu valodas – Agile software development) ir programmatūras izstrādes ietvars, kas nosaka tās izstrādes veidu un darbības pieeju uzdevumu risināšanā. Spējā metode definē projekta attīstību un ieviešanu iteratīvā izstrādes pieejā, kur visa projekta dzīves cikla laikā tiek veidotas mazākas darbu pakotnes kā nodevumi, kas rezultātā ir strādājoši un funkcionējoši sākuma produkti. Katrā ciklā tiek ievēroti tradicionālās pieejas programmatūras izstrādes būtiskākie soļi:

- Sistēmas prasību specificēšana un analīze
- Sistēmas projektējuma aprakstīšana un analīze
- Programmatūras prasību specificēšana un analīze
- Programmatūras produkta detalizēta projektēšana
- Programmēšana un testēšana
- Integrēšana

Katrs cikls tiek veikts noteiktās laika robežās, no 2 - 4 nedēļām. Uzdevumi, kas nav pabeigti kāda cikla ietvaros tiek pārlikti uz nākošo ciklu, izvērtējot tā prioritātes un turpmākos uzdevumus. Spējās programmatūras izstrādes pielietojums veido ciešāku sadarbību ar klientu, labāku komunikāciju komandas dalībnieku starpā, kas ir būtiski sarežģītu un inovatīvu projektu īstenošanā. Izmantojot spējās metodes, darbu nodevumus pēc iespējas ātrākā laika posmā jau var pielietot, jo katrs nodevums ir kāda funkcionējoša izstrādes daļa projekta ietvaros. Īpaši svarīgi tas ir biznesa jomā, kur pamata funkcionalitātes esamība, ātrāka tās pielietošana un novērtēšana, nodrošina uzņēmuma vērtības straujāku izaugsmi un attīstību.

### 4.1. Spējie principi un vērtības

Spējās metodes noteikšanā tiek izmantota pieejas atbilstība spējām vērtībām, principiem un praksēm. Spējās vērtības kā galvenās spējās īpašības [10]:

- ✓ Cilvēki un to mijiedarbība ir svarīgāka par noteiktiem procesiem un rīkiem
- ✓ Strādājoša programmatūra ir svarīgāka par detalizēti aprakstošu dokumentāciju
- ✓ Sadarbība ar pasūtītāju vai klientu ir svarīgāka par līguma saskaņošanas procesu
- ✓ Atbilstoša un ātra reakcija uz nepieciešamajām izmaiņām ir svarīgāka par sekošanu plānam un tā izpildes grafikiem

Spējie principi tiek balstīti uz iepriekš definētajām vērtībām, kurām (vērtībām) ir augstāka prioritāte [11]:

- ✓ Apmierināts klients, kuram regulāri tiek piegādāts strādājošs produkts
- ✓ Bieža komunikācija ar visām iesaistītajām pusēm, kā arī komandu aci pret aci
- ✓ Galvenais izstrādes mērķis – strādājošs produkts vai piegādāts pakalpojums
- ✓ Darbu izstrādē iesaistās izstrādātāji un klienta pārstāvji
- ✓ Ātri pieņemt un pēc iespējas ātrāk atrisināt noteiktās prasības vai izmaiņu pieprasījumus
- ✓ Produkta nodevumu piegāde pēc veiktajiem izstrādes cikliem (piemēram, vairākas reizes mēnesī)
- ✓ Vienkāršība visos procesos, maksimāli daudz paveicot, izmantojot pēc iespējas mazāku piepūli
- ✓ Vienmēr pārraudzīts un atbilstoši novērtēts labs tehniskais risinājums un projektējums
- ✓ Novērtētais tehniskais risinājums, projektējums un definētās prasības, kas atbilst klienta vēlmēm, tiek piedāvāts no vislabāk pašorganizējošākās komandas
- ✓ Motivēts komandas dalībnieks kā būtisks komandas veidošanas priekšnoteikums
- ✓ Komanda novērtē savus darba rezultātus un pilnveido tās saistītos procesus
- ✓ Nemainīgi paredzams produktu vai servisu izstrādes temps

Noteiktas spējās prakses tiek definētas, ņemot vērā aprakstītos spējos principus, augstākā procesu detalizācijas līmenī.

## 4.2. Spējās metodes

Lai nodrošinātu spējo metožu ieviešanu uzņēmumā, nepieciešams detalizētāks to apskats un novērtējums. Apskatītas populārākās spējās metodes pēc to pielietojuma.

### 4.2.1. *Scrum metode*

Scrum ir projektu vadības metode, kas nosaka komandas darba organizācijas veidu, pieeju. Scrum metodes galvenās prakses [6] ir:

#### - *Iterāciju veidošana*

Darbs tiek organizēts un plānots mazās iterācijās, kas nav garākas par 4 nedēļām. Katras iterācijas beigās tiek piegādāts strādājošs produkts, kas ir atbilstoši notestēts un pārbaudīts, vēlāk arī apstiprināts katrs nodevums.

- *Papildinājumu veidošana*  
Veicot papildinājumu veidošanu, tiek nodrošināta tā ātra un vienkāršota integrēšana jau esošajā programmatūras kodā. Papildinājumi tiek veidoti ievērojot vienkāršuma un mazākās piepūles metodi. Sprinta nobeigumā netiek novērtēta neeksistējoša dokumentācija kā vien strādājošs produkts vai piegādāts serviss.
- *Prasību parādīšanās*  
Izstrādājot komplicētus produktus un servissus, iepriekš nav iespējams definēt visas prasības, kā arī pietiekamu to detalizācijas līmeni. Tiek ievērots attīstības un izmaiņu uzklauššanas princips, kas veicina veiksmīgu un pēc iespējas ātrāku tālākā projekta virzību.
- *Komandas pašorganizēšanās*  
Komandas izmanto jebkādu pašas definētus rīkus un procesus, lai sasniegtu izvirzīto mērķi, ko uzskata par vispiemērotākajiem un atbilstošākajiem, kā, piemēram, arhitektūras līmeņu veidošanas rīku vai efektīvākas funkciju darbības dokumentēšanas rīku papildus komentāru esamībai programmatūras kodā.
- *Savstarpēja sadarbība*  
Komandas pašorganizēšanās ir veiksmīga, ja savstarpēja komunikācija starp komandas locekļiem, klientu un visām iesaistītajām pusēm ir brīva un nepiespiesta. Pāru programmēšana un kopīga darba vide, kur visi komandas locekļi atrodas vienā telpā, sekmē savstarpējas komunikācijas uzlabošanu, nodrošinot labās prakses principus.

#### **4.2.1.1. Scrum lomas**

##### ***Produkta īpašnieks***

- Iepazīstina ar produktu
- Identificē un apkopo produkta vai servisa prasības
- Prioritizē produkta darba uzdevumu sarakstu atbilstoši biznesa vajadzībām
- Definē sprinta saturu un mērķus, iekļaujot katrā sprintā būtiskākos uzdevumus no darba uzdevumu saraksta, kas iepriekš ir prioritizēti
- Piedalās sprinta pārskata sanāksmēs un nodrošina atbilstošu atskaišu sagatavošanu

### ***Scrum skolotājs***

- Uzlabo un palīdz veikt komandas darbu, atbilstoši Scrum principiem un labās prakses piemēriem
- Nodrošina komandas netraucētu darbību
- Pieņem lēmumus, kas saistīti ar komandas tālāko darbību
- Sadarbojas ar augstāka līmeņa vadību, kā arī komunicē ar komandu
- Vada ikdienas Scrum 15 minūšu sapulces
- Vada sprinta pārskata sapulci

### ***Komanda***

- Labi motivēta, kas veic visas nepieciešamās darbības, lai sasniegtu sprinta mērķi
- Komandas sastāvā netiek veiktas izmaiņas viena sprinta ietvaros
- Definē detalizētus uzdevumus, lai veiktu nosprausto sprinta mērķi

#### ***4.2.1.2. Scrum principi***

Produkta īpašnieks apkopo visas iespējamās sākotnējās prasības pret izstrādājamo produktu un definē to izpildes secību, veicot sākotnējo uzdevumu izpildes novērtējumu. Atbilstoši produkta īpašnieka definētajām prioritātēm, kas balstītas uz svarīgākajām biznesa vajadzībām, komandas dalībnieki nosaka kādi uzdevumi jāveic, lai īstenotu konkrētu aktivitāti, novērtējot katru uzdevumu laika izteiksmē, kā arī definē sprinta saturu un mērķi. Ikdienas sanāsmēs komandas dalībnieki atjauno uzdevumu sarakstu, to novērtējumu un identificē problēmas un radušos šķēršļus, ja tādi radušies kā arī informētu par veicamajiem darbiem šajā dienā. Sprinta beigās komandas dalībnieki kopā ar produkta īpašnieku veic sprinta novērtējumu, analizējot iespējamus uzlabojumus un novērtē tos.

### ***Sapulces***

Lai nodrošinātu veiksmīgu komandas locekļu savstarpēju komunikāciju, nepieciešams organizēt sprinta labās prakses piemēriem atbilstošas sanāksmes.

### ***Sprinta ikdienas sanāksmes***

- Sanāksme tiek organizēta 15 minūšu garumā, kājās stāvot
- Sākums un laiks katru dienu ir viens un tas pats
- Katras sanāksmes ietveros tiek atbildēts uz šādiem jautājumiem:
  - Kas ir paveikt kopš iepriekšējās dienas?
  - Ko tu šodien plāno paveikt?
  - Vai ir kādas problēmas un šķērši kas ir identificēti, lai sasniegtu mērķi?

### ***Sprinta plānošanas sanāksme***

Katrā sprintā tiek organizēta atsevišķa sprinta plānošanas sanāksme

- Sanāksmē noteikt kādi ir sprintā veicamie darbi
- Sagatavot sprinta detalizētu darba uzdevumu sarakstu
- Identificēt un noskaidrot cik daudz darba visticamāk tiks paveikts šajā sprintā
- Astoņu stundu laika limits (produkta īpašnieks un komanda pirmās četras stundas, dialogs par darba uzdevumu saraksta prioritizēšanu, pārējās četras stundas tikai komanda – precizē un pabeidz sprinta darba uzdevumu sarakstu)

Sprinta beigās tiek organizētas divas sapulces – sprinta beigu sanāksme un pārskata veidošanas sanāksme.

### ***Sprinta beigu sanāksme***

- Tiek pārskatīts paveiktais un nepabeigtais darbs
- Prezentēt pabeigto darbu sprinta ietvaros un darbu pakotnes visām iesaistītajām pusēm, kas nav uzskatāma par gala produkta versiju
- Nepabeigto darbu nedrīkst rādīt
- Četru stundu laika limits

### ***Sprinta pārskata veidošanas sanāksme***

- Visi komandas locekļi atskatās uz iepriekšējo sprintu
- Veic nepārtrauktu procesu uzlabošanu
- Galvenie uzdotie jautājumi sanāksmes laikā:
  - Kas gāja labi iepriekšējā sprintā?
  - Ko var uzlabot nākošajā sprintā?
- Trīs stundu laika limits

#### **4.2.2. Extreme Programming (XP) metode**

XP metode laika gaitā ir kļuvusi par sinonīmu spējai programmatūras izstrādei. XP ir inženierijas pārvaldības metode, kuras galvenais mērķis ir samazināt projektu izmaiņu izmaksas, kas īpaši izteiktas ir produkta uzturēšanas fāzē.

##### **4.2.2.1. XP lomas**

XP metodes nosaka arī lomu pielietojumu labās prakses ievērošanas ietvaros [13]:

##### **Treneris**

- Seko līdzi, lai komanda ievērotu XP principus un ir priecīgs strādāt komandā
- Uzņemas atbildību par visām lietām, kas var noiet greizi

##### **Sekotājs**

- Veic projekta progressa mērījumu veikšanu iterācijas laikā, atzīmējot uzdevumus, kuru izpilde ir kavēta
- Pašmotivēta persona, taču ar labām komunikācijas spējām, individuāla darba veikšanai
- Veic mērījumu analīzi (piemēram, uzskaitot visu informāciju MS Excel rīkā)
- Atjauno un papildina relīzes plānu

##### **Programmētājs**

- Pārziņina vairākas programmēšanas valodas
- Nepieciešamības gadījumā apmāca citus un skaidro citiem darba uzdevumus
- Iesaistās darbu uzdevumu novērtēšanā

##### **Klients**

- Produkta pasūtītājs, kas piedalās programmatūras izstrādes un plānošanas procesos
- Vienojas iekšēji pasūtītāja kompānijā par specifisku prasību nepieciešamību un to tālāku nodošanu programmatūras izstrādātājam

##### **4.2.2.2. XP principi**

Ekstrēmā programmēšana aptver šādus galvenos principus [13]:

##### **Plānošana**

- Klienta prasības ir aprakstītas
- Tiek veidotas mazas relīzes, ievērojot nepārtrauktības principu

- Projekts ir nodalīts iterācijās
- Iterāciju plānošana ir katras iterācijas sākuma process

#### ***Pārvaldība***

- Komanda atrodas lielā, nenaslēgtā telpā
- Katru dienu tiek organizēta 15 minūšu sapulce, kājās stāvot
- Komandas dalībnieki ik pa laikam tiek pārvietoti

#### ***Dizaina izstrāde***

- Vienkāršība kā galvenais priekšnoteikums
- Tiek lietota klientam saprotama valoda, izmantojot metaforas
- Papildus funkcionalitāte netiek iepriekš izstrādāta

#### ***Kodēšana***

- Klients vienmēr ir pieejams
- Koda izstrāde balstās uz apstiprinātiem standartiem
- Vienību testi tiek izstrādāti pirms koda izstrādes
- Pāru kodēšana
- Tikai viens pāris integrē kodu konkrētajā momentā
- Biežs un nepārtraukts integrācijas process
- Atbildība par kodu vienādi augsta starp visiem komandas locekļiem
- Integrācijas nodrošināšanai atbilstoša aparatūra

#### ***Testēšana***

Visam kodam nepieciešami vienību testi

Visam kodam jāizpilda vienību testu prasības

Kļūdu gadījumā tiek izstrādāts tests

Akcepttesti tiek bieži izpildīti un rezultāti publicēti

### **4.3. Metožu pārbaude un izsekojamība**

Veicot procesu uzlabojumus un jaunu procesu ieviešanu uzņēmumā, nepieciešams arī objektīvs novērtējums, vai pamatprincipi metožu pielietošanā tiek ievēroti. Visvienkāršākais veids kā veikt novērtējumu ir pamatjautājumu uzdošana komandas vadītājam, kā arī pašai komandai, kas aptver būtiskākās metodes procesu apgabalus vai prakses piemērus, kā, piemēram, ikdienas sapulču organizēšana projekta komandai 15 minūtes garumā. Kontroljautājumus Scrum metodes ieviešanas pārbaudei izveidojusi kompānija Nokia, kas arī izmanto šo metodi jaunu produktu un pakalpojumu izstrādē.

## *Nokia tests*

Kompānija Nokia, kas pati izmanto Scrum, ir izstrādājusi interesantu testu, kas ar vienkāršiem jautājumiem pārbauda vai konkrētajā projektā tiek pielietotas Scrum metodes. Šāds kontroles jautājumu saraksts izveidots, jo ļoti daudzi projektu ieviešanu mēģina piedēvēt kā Scrum, īstenībā izmantojot vai nu tikai kādu daļu no Scrum metodēm vai arī pielietojot pareizos terminus, bet īstenībā tas neatšķiras no tradicionālā Ūdenskrituma modeļa.

Nokia tests sastāv no divām daļām [1]:

*Pirmā daļa, Jūs veicat iteratīvu izstrādi, ja:*

- ✓ Iterācijas ir ar noteiktu maksimālo izpildes laiku – 4 nedēļas
- ✓ Programmatūras funkcionalitāte ir notestēta un strādājoša katras iterācijas beigās
- ✓ Iterācijai jā sākas ātrāk kā ir pilnībā pabeigtas produkta prasības un funkciju specifikācijas

*Otrā daļa, Jūs ievērojat Scrum principus, ja:*

- ✓ Ir zināms, kurš ir produkta īpašnieks (no angļu val. – Product owner)
- ✓ Produkta uzdevumu saraksts (no angļu val. – Product backlog) ir prioritizēts pēc biznesa loģikas un nozīmības
- ✓ Projekta komanda ģenerē veicamos un nosaka paveiktos darbu uzdevumus, tiek veidots atbilstošs darbu izpildes grafiks (no angļu val. Burndown chart and Velocity)
- ✓ Projektu menedžeri (vai jebkurš cits) netraucē komandas darbu

*Trešā daļa kā Scrum principu ievērošanas papildinājums (novērtējums):*

- ✓ Programmatūra tiek piegādāta atbilstoši svarīgākajām biznesa procesu vajadzībām
- ✓ Procesi tiek nepārtraukti uzlaboti
- ✓ Produkta īpašnieks var veikt uzdevumu prioritizēšanu
- ✓ Produkta īpašnieks uztur ciešu kontaktu ar visām projektā iesaistītajām pusēm
- ✓ Produkta uzdevumu sarakstu komanda atjauno katru dienu
- ✓ Komandas sapulces 15 minūšu garumā tiek organizētas katru dienu visu komandas dalībnieku sastāvā, identificējot radušās problēmas, radušos šķēršļus un veicamos darbus
- ✓ Programmatūras nepārtrauktu piegādi pēc katra cikla apstiprina visas iesaistītās puses

- ✓ Pēc katras iterācijas veikšanas tiek veikta sprinta pārskats, iesakot iespējamus uzlabojumus un novērtējot tos
- ✓ Sprintā sasniedzamo darbu rezultātus definē produkta īpašnieks, kas veic aptuveno darbu novērtējumu un prioritizāciju tā, lai tas iekļautos viena sprinta ietvaros, kur tālāk komanda uzskaita precīzu darba uzdevumu sarakstu un veic to precīzu novērtējumu
- ✓ Tiek organizētas sprintu plānošanas sanāksmes, kur produkta īpašnieks atjauno galveno darbu sarakstu un prioritātes
- ✓ Komanda sēž vienā telpā un komandas lielums nepārsniedz 9 cilvēkus

*Ceturtdā daļa - papildus tam iespējams ieviest vēlamo prasību atbilstības novērtējuma sadaļu ar šādiem punktiem:*

- ✓ Komandai ir visas nepieciešamās zināšanas, lai novērtētu darba pakotnes un sasniegtu sprintā definētās prioritātes un mērķus
- ✓ Komandas locekļiem nav konkrēti definētas lomas tā, lai darbu izpilde un atbildība starp visiem būtu līdzvērtīga
- ✓ Iterācijas, kuras nav iespējams pabeigt tiek nekavējoties pārtrauktas
- ✓ Visi komandas locekļi iesaistās darba uzdevumu novērtēšanā (kāršu princips, nosakot iepriekš definētu vērtību kāda uzdevuma izpildei)
- ✓ Visa komanda ir informēta par lielākajām problēmām un šķēršļiem darbu izpildes gaitā
- ✓ Darba uzdevumu saraksta novērtējuma sadaļu katru dienu atjaunina komandas locekļi
- ✓ Darbu izpildes ātrums tiek novērtēts atbilstoši paveiktajam darbam
- ✓ Ikdienas sanāksmes notiek katru dienu vienā laikā, tajā pašā vietā
- ✓ Katrs komandas dalībnieks zin ko pārējie dara
- ✓ Virsstundu darbs ir reta parādība, taču nepieciešamības gadījumā komandas dalībnieki paši uz to piesakās

### ***XP Evaluation Framework***

Novērtējot XP metodes ieviešanu uzņēmumā tiek ievēroti zemāk uzskaitītie XP prakses un principi [9]:

Derīgas atsauksmes:

- Testu virzīta attīstība

- Izstrādāt testus pirms kodēšanas
- Uzrakstītajiem testiem jāpārsedz visas sistēmas funkcionalitāti
- Izstrādātajam kodam jābūt saglabātam koda kontroles repozitorijā
- Plānošanas spēle, process
  - Definēt relīzes sfēru, novērtējot katra uzdevuma izpildes laiku
  - Definēt uzdevumu prioritāro secību, novērtējot tehnisko risinājumu ietekmi uz noteiktajām prasībām
  - Noteikt relīzes sagatavošanas datumu, veicot darba sadalījuma struktūras izveidi un nosakot darba uzdevumus
- Klienta pārstāvis, kas atrodas uz vietas
  - Klienta pārstāvja iesaistīšana plānošanas, izstrādes un darba pakotņu nodošanas - pieņemšanas procesā
- Pāru programmēšana
  - Pāru programmēšana, izmantojot automātisko testu ietvaru (piemēram, Junit vai VJUnit), kas rezultējas ar mazāku kļūdu pieļaušanu un iespējamību
  - Dizaina izstrādē lēmumu pieņem divi cilvēki
  - Veicot pāru maiņu tiek nodrošināta ātrāka zināšanu apmaiņa tālākā darba veikšanā
  - Izstrādājot kodu, tas tiek pārskatīts visu laiku
  - Viens iedziļinās detaļās kamēr otrs cilvēks novērtē konceptu
  - Pāru programmēšana jāievēro kopā ar programmēšanas, testēšanas un kodēšanas standartu XP praksēm

Nepārtraukts process labāk nekā sistematizēts:

- Nepārtraukta integrācija
  - Nepārtraukta koda integrēšana sistēmā pa maziem posmiem (jo lielāki posmi starp integrācijas starpposmiem, jo integrēšanai patērētais laiks paildzinās)
- Mazas relīzes
  - Klients strādā vienoti ar visu komandu, lai definētu prasības un uzdevumu prioritātes
- Pārveidošana
  - Koda uzlabošana, nemainot funkcionalitāti

Vienota izpratne:

- Vienkāršs dizains
  - Tiek izpildīti visi testi
  - Nesatur koda dublikātu
  - Satur vienkāršāko implementāciju, izmantojot vismazāko klašu un metožu skaitu
- Sistēmas metaforu lietošana
  - Metaforu lietošana klientam saprotamā veidā, piemēram, aprakstot kādu funkcionalitāti lietot klientam redzamo darbību aprakstu
  - Komunikācijai ar klientu jābūt efektīvai, visām iesaistītajām pusēm viegli saprotamai
- Kods kā vienots īpašums
  - Sistēmas funkcionalitātes atbildīgo personu deleģēšana, ievērojot pāra principu
- Kodēšanas standarti
  - Vadlīniju un kodēšanas rekomendāciju ievērošana
  - Koda formatēšana (viegli lasāms un pārskatāms)
  - Koda struktūra (datņu, klašu un dažādu resursu loģiska atrašanās vietas definēšana)
  - Vienots metožu, klašu, mainīgo un parametru nosaukumu veidošanas princips
  - Kļūdu aprakstīšana
  - Komentāri, kas apraksta koda būtību un loģiku, kas nodrošina atbilstošu koda kvalitāti un vieglu tā pārskatīšanu, uztveri un tālāku uzlabošanu, ja nepieciešams

Programmētāja labklājība:

- 40 stundu darbs nedēļā
  - Neievērojot šo principu samazinās izstrādātās programmatūras kvalitāte un darba efektivitāte

Papildus autora izstrādāta projektu ieviešanas un rīku izmantošanas, un darbu uzskaites, aizpildes novērtēšanas anketa (skat. Pielikums Nr.7).

## 5. PIELIETOTĀ METODEDE UN SITUĀCIJAS NOVĒRTĒJUMS

### 5.1. Izmantoto rīku novērtējums

Izmantoto rīku apskats pēc sniegtajiem ieteikumiem un rekomendācijām tiek papildināts ik nedēļas procesu un rīku uzlabojumu sapulces ietvaros. Kopš sākotnējā procesu novērtējuma tiek izmantoti vairāki jauni rīki un dokumentu formas, paraugi, kas iepriekš nav minēti, taču ir nepieciešamība tos lietot. Katrs rīks vai procesu apgabals ir novērtēts ar atbilstošu prioritāti (1-augsta, 2-vidēja, 3-zema, nozīmīga) organizācijas kopējo procesu un to attiecības un nozīmības kopskatā. Katram rīkam tiek piešķirta loma procesu nozīmes kontekstā. Kolonna „Lietošanā” atspoguļo pēc sākotnējo procesu analīzes, visus pielietotos rīkus, kur situācijā, ja uz doto brīdi šāds rīks tiek pielietots, tas tiek atzīmēts ar „x”. Identificējot un analizējot visus procesu apgabalus ir secināts kādiem procesiem, izmantojot atbilstošus rīkus, jābūt savstarpēji saistītiem, līdz ar to procesu apgabali, kas ir būtiski projektu ieviešanā vai uzturēšanā, un kuriem nav definētu un izvēlētu rīku, prioritātes novērtējums ir augstāks, nekā situācijās, kad rīks jau tiek izmantots ikdienas darbā. Rīka izmantošanas gadījumā, kas pārsedz jau identificētu procesu apgabalu, prioritāte ir zemāka un nav nepieciešamas steidzamas darbības šajā jomā.

Nr.	Rīks	Procesu un rīku pārvaldītā joma un apraksts	Lietošanā esošs	Plānots ieviest	Prioritāte
<b>Prasību pārvaldība</b>					
1.	Nav definēts	Nav definēts vienots rīks, kas apkopotu projektu prasības (sasaistītu sākuma prasību sarakstu (no angļu val. – backlog) ar projekta plānu)		x	1
2.	MS Excel, MS Word	MS Excel izklājlapa (MRD, FRS, SRS – Izsekojamības matrica)	x		2
3.	Sharepoint	Dokumentu glabātuve, datņu versiju kontrole	x		1

<b>Projektu plānošana</b>					
4.	MS Project 2002	Projektu plānošana (Komandas plānotājs, Gantt grafiks)	x		2
5.	Clarity	Plānošanas un finansu uzskaites rīks (iespēja eksportēt un importēt projektu plānus MS Project formātā)	x		3
6.	MS Excel	Darba uzdevumu sadalījuma struktūra, darba pakotņu sākotnējie novērtējumi	x		2
<b>Resursu pārvaldība</b>					
7.	Clarity	Kā galvenais rīks resursu lietošanā un noslodzes mērīšanā	x		2
8.	MS Project 2002	Komandas plānotāja skata izmantošana	x		2
9.	MS Excel	Produktu vienota MS Excel izklājlapa resursu izmantošanas atspoguļošanai, pa mēnešiem, augstākā līmeņa vadībai (pārklājas ar Clarity uzkrāto un atspoguļoto informāciju)	x		2
<b>Projektu pārraudzība un kontrole</b>					
10.	Clarity	Resursu noslodzes atspoguļošana, uzdevumu piešķiršana un pārdalīšana Reāli paveikto darbu uzskaitē attiecībā pret piešķirtajiem un	x		2

		nozīmētajiem uzdevumiem, Timesheet actuals vs. allocations Allocation vs. assignment			
11.	MS Project 2002	Komandas plānotājs, vienkāršota pieeja resursu noslodzē un uzdevumu progresijai, uzraudzībai un kontrolei.	x		2
<b>Dizains un analīze</b>					
12.	Enterprise Architect	Dizaina un arhitektūras veidošanas rīks	x		2
13.	Code collaborator	Koda analīzes un pārskata rīks	x		3
<b>Izstrāde</b>					
14.	Jebkurš teksta labošanas rīks	Izejas koda labojumu veikšanas rīks	x		3
15.	ODT Artema plugin	Artema Programmatūras izstrādes rīku komplekts	x		3
16.	HC ODT	Optimum Programmatūras izstrādes rīku komplekts	x		3
17.	MS Visual Studio	Aplikāciju izstrādes rīks	x		2
18.	Subversion (SVN)	Programmatūras versiju kontroles rīks	x		3
19.	HOST	Iekšēji izmantotie HC simulāciju rīki	x		3
20.	Doxygen	Rīks programmatūras dokumentēšanai	x		3

21.	Ikdienas relīžu gatavošanas sistēma	Uzņēmuma izveidota sistēma relīžu procesa automatizētai nodrošināšanai, aplikāciju ģenerēšanai (uz doto brīdi automātiski tiek ģenerēta viena no vairākām aplikācijām) Nepieciešams paplašināt sistēmas relīžu gatavošanas procesu visām aplikācijām	x		1
22.	Vertikālā griezumaska rīks	Nav informācijas Izpildījuma analīze		x	2
23.					
24.	<b>Testēšana</b>				
25.	ATS	Automātiskā Testēšanas Sistēma (funkciju un integrētās funkcionalitātes testēšanas rīks)		x	2
26.	Analīzes rīks	Rīks, kas analizē izejas/starpposmu ierakstu failus	x		
27.	Vienību / Moduļu testēšana	Nav informācijas		x	1
28.	Testu pārvaldības rīks	Nav informācijas		x	1
29.	Koda kontroles rīks	Nav informācijas (Iespējamais rīks: GCOV)		x	1
	<b>Izmaiņu pieprasījumi, pieteikumi</b>				
30.	Clarity	Papildus rīks nav nepieciešams problem pieteikumu uzskaitēi	x		3

<b>Problēmu pieteikumu pārvaldība</b>					
31.	Clarity	Problēmu pieteikumu uzskaites un pārvaldības nodrošināšanai papildus rīks nav nepieciešams. Papildinājumi Clarity programmatūrā nepieciešami balstoties uz darba grupās apstiprināto un saskaņoto informāciju	x		3
<b>Mērījumi un analīze</b>					
32.	Clarity	Ieviešana sākuma fāze, izstrādāti daži papildus skati informācijas apskatei un analīzei. Papildus nepieciešams datu apkopojums pa produktu grupām, lai atbilstoši reaģētu uz nākošo plānošanu, plānu izstrādi, ņemot vērā iepriekšējo statistiski uzkrāto pieredzi	x		1

## **5.2. Metožu pielietojums un mijiedarbība**

Programmatūras izstrādē būtisku lomu spēlē projektu vadības un inženierijas metodes izvēle, ko ietekmē projekta veids, tā sarežģītība, biznesa attiecības un vide, sadarbība un komunikācija, resursu pieejamības ietekmējošie rādītāji u.c. aktivitātes un procesi.

Veiktie procesu uzlabojumi uz doto brīdi ietekmē jaunāko produktu līniju ieviešanas un pārvaldības procesus. Izstrādājot jaunu produktu līniju, tiek veidota ciešāka saikne ar potenciālajiem un esošajiem klientiem, kur jau projektu ieviešanas un definēšanas sākuma fāzē tiek noteikts, ka projekta veiksmīgai realizācijai kā pozitīvs jaunpievedums būtu biežāka komunikācija ar klientu un tā iesaistīšana esošajos procesos. Tādējādi iespējams uzsākt

ciešāku sadarbību un nodrošināt klienta līdzdalību projektu definēšanas un ieviešanas fāzēs. Būtisku lomu klienta līdzdalībai projektu ieviešanā spēlē tas, ka klients vienmēr zin labāk to, ko viņš vēlas, līdz ar to prasību specificēšanas un detalizēta procesa aprakstam ir jābūt pēc iespējas vienkāršotam un jānodrošina visātrāko prasību identificēšanu un apkopošanu. Izvēloties projekta vadības pieejas un metodes, paralēli tiek ievērots CMMI princips – dokumentācijas esamība un sekošana tai, atbilstoša tā atjaunināšana, kā arī komandas līdzdalība visos procesos produkta izstrādes gaitā. Papildus tam tiek izmantota Scrum un XP prakšu kopsavilkums, kas pārklāj nedefinētās vai iepriekš neskaidrās projektu vadības un inženierijas metodes un procedūras.

### ***5.2.1. Projektu novērtējums un metodes izvēle***

Projektu metodes izvēlē un noteikšanā lielu lomu spēlē projekta veids, konteksts, sarežģītība, inovāciju ieviešanas nepieciešama u.c. saistītie faktori. Iteratīva pieejas metode nodrošina neskaidru vai nedefinētu sākotnējo prasību ieviešanu sistēmā. Projekta metožu izvēlē projektu ietvaros klienti pilnībā atbalsta biežāku komunikācijas nodrošināšanu, kā arī iteratīvu programmatūras piegādes veidu, piegādājot strādājošu programmatūru pa cikliem.

### ***5.2.2. Pielietotās metodes apskats***

Apskatot populārākās spējās projektu vadības un inženierijas metodes (Scrum un XP) ir secināms, ka ir jāizmanto labo prakšu piemēri, izmantojot savstarpēju šo metožu mijiedarbību. Identificētie procesu apgabali, kas netiek pārklāti ar spējo metožu prakses piemēriem tiek pārklāti ar CMMI procesu uzlabojumu ieteikumiem, kā arī labās prakses piemēriem. Pielietojot spējās metodes, nepieciešams novērtēt galvenos posmus šīs metodes ieviešanā:

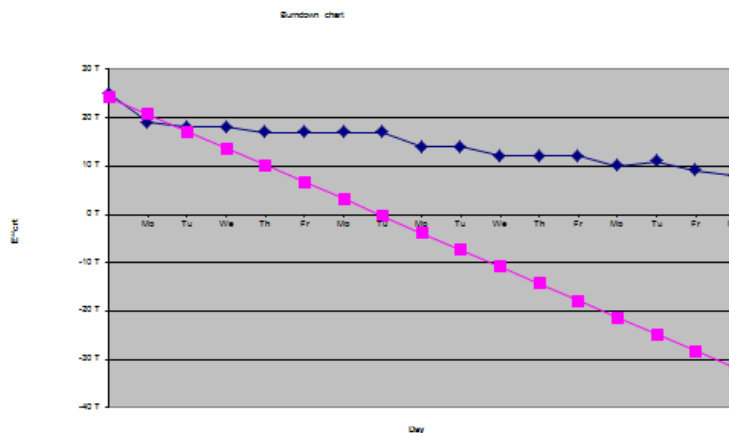
#### ***Sprinta sākums un plānošanas process***

Iterācijas ilgums tiek noteikts 4 nedēļas, kas rezultējas ar relīzes sagatavošanu projektu ietvaros. Četras nedēļas izvēlētas, lai nepārsniegtu maksimālo iterācijas izpildes termiņu, kā arī iekļautos vidēji grūtu uzdevumu risināšanas termiņos viena sprinta ietvaros. Sprinta saturs un mērķis tiek definēts, saņemot reģionālo izstrādes prioritāšu sarakstu no Cross-Functional-Product (turpmāk teksā – CFP) departamenta. Spējo principu ieviešanas fāzē, prioritāšu saraksta apkopošana no reģioniem un pēc tam to tūlītēja grupēšana un prioritizēšana sagādāja daudzas problēmas, jo dažbrīd problēmu pieteikumu vai uzdevumu

prioritārie uzdevumi atradās CFP saraksta lejas daļā. Sarakstam būtu jābūt grupētam pirmajā pozīcijā virzot uzdevumus ar visaugstāko prioritāti. Līdz ar to spējas ieviešanas sākuma stadijā CFP saraksta uzdevumi tika atkārtoti pārgrupēti un prioritizēti, lai darba uzdevumi sprintā atbilstu aktuālākajām biznesa vajadzībām. Sprinta plānošanas procesā tāpat tiek iekļauti no iepriekšējā sprinta nepabeigti darba uzdevumi un pakotnes. Katrā sprinta sākuma etapā tiek noteikts sprinta saturs un mērķis, kur rezultātā tiek piegādāts strādājošs produkts. Projekta komandai jāievēro sprinta sākumā etapā definētais saturs un mērķis visa cikla garumā. Plānošanas procesā komandas dalībnieki definē apakšuzdevumus katrai aktivitātei un veic to darbietilpības novērtējumu. Darba uzdevumi un pakotnes tiek uzskaitītas un aprakstītas Clarity programmatūrā, kas nodrošina darbietilpības novērtējuma pievienošanas iespēju katram darba uzdevumam, žurnālēšanas funkcionalitāti un papildus laukus produktu līnijas aprakstam. Veicot darbietilpības novērtējuma uzkrāšanu un pēc tam analizējot vēsturisko novērtējuma mainību procesa ietvaros, iespējams turpmāko novērtējumu nodrošināt, izmantot statistiski pamatotus secinājumus, nosakot koeficientu salīdzinājumā ar plānoto pret reālo darbietilpības novērtējumu.

### ***Sprinta izpilde un darbu izpilde***

Sprinta izpildes laikā katru dienu tiek organizētas komandas sanāksmes 15 minūšu garumā. Lai ievērotu 15 minūšu robežu, sanāksme notiek visiem kājās stāvot, tādējādi visi komandas dalībnieki vēlas ātri izrunāt pašu svarīgāko: kas ir paveikt kopš iepriekšējās sanāksmes, kas ir jāpadara līdz nākošajai un jāpiemin radušās problēmas vai šķēršļus, ja tādi radušies. Problēmu risināšana šajās sanāksmēs netiek apspriesta, ja tās atrisinājums aizņem pārāk ilgu laiku. Problēmu risināšana tiek veikta atsevišķās sanāksmēs, kas tiek organizētas pēc 15 minūšu sapulces. Komandas dalībnieki atjaunina Sprinta atlikušā darba uzdevumu sarakstu ar jauniem ierakstiem, atjauninot arī katra uzdevuma novērtējumu.



Attēls Nr.5.2.2.1. „Sprinta Nr.1 atlikušā darba grafiks”

Kā redzams, pirmā sprinta plānošana ir bijusi pārāk optimistiski novērtēta un reālo veicamo darbu apjoms, kas bijis ieplānots šajā sprintā pārceļas uz nākošo sprintu, izvērtējot nākošā sprinta prioritātes. Vertikālā ass atspoguļo atlikušā darba daudzumu cilvēkstundās, savukārt horizontālā ass, tā izpildi pa dienām.

Prasību precizēšanai un ievietošanai sprinta uzdevumu sarakstā tiek īstenota, izmantojot telekonferences un dažkārt arī videokonferences zvanus, kā arī e-pasta sarakstes veidā. Pēc prasību identificēšanas tās tiek reģistrētas pievienotas darbu izpildei konkrētajā sprintā. Izstrādājot katru sprinta uzdevumu, izstrādātais kods tiek sagatavots un nosūtīts komandas vadītājam koda apskatei, lai tiktu novērtēts un apstiprināts. Apstiprināšanas gadījumā, koda izstrādātājs pārsūta šo uzdevumu testēšanai.

### ***Sprinta beigšana un pārskata veidošana***

Katra sprinta nobeigumā klientam tiek piegādāts strādājošs produkts, kas pirms tam ir notestēts. Klientam nepieciešams veikt akcepttestēšanu un apstiprināt konkrēto nodevumu. Tāpat sprinta beigās komanda pārskata paveikto sprintā un identificē, un apkopo galvenās problēmas, rosinot iespējamus procesu uzlabojumus turpmākajā darbā. Iterācijas novērtējums, pēc autora domām, ir kritiskākais punkts procesu uzlabošanas pasākumu īstenošanā, kas ir ciešā saistībā ar CMMI brieduma līmeņa definēto mērījumu uzkrāšanas un analīzes procesu apgabalu.

### ***5.2.3. Uzdevumu uzskaitē un pārvaldība***

Uzdevumu uzskaitē un pārvaldībai ir būtiska loma projektu virzības nodrošināšanā, jo balstoties uz šo rīku, iespējams veikt savstarpēju sasaisti starp dažādiem procesu līmeņiem, kā, piemēram, prasību specificēšanu un izskaiti, koda uzglabāšanas un versiju kontroles rīku. Balstoties uz pieteiktajiem un reģistrētajiem uzdevumiem rīkā, tiek nodrošināta komandas veiksmīga sprinta izpilde, tos apstrādājot un piesakot koda izmaiņas, relīzes sagatavošanai. Darbu uzdevumu uzskaites rīkam jāspēj arī nodrošināt šādas funkcijas:

- ✓ Katram reģistrētajam uzdevumam jāietver laika zīmogs, konkrēts apraksts un uzdevuma pieteicējs
- ✓ Katrs uzdevums satur unikālu identifikatoru, lai to būtu iespējams vieglāk izsekot
- ✓ Papildus nepieciešama darbietilpības novērtējuma uzskaitē, izmantojot žurnālēšanas funkcionalitāti, lai spētu novērtēt cik precīzs ir bijis novērtējums kāda uzdevuma izpildei tā atrisināšanas laikā, salīdzinot ar reāli patērēto laiku

- ✓ Izziņošanas funkcionalitāte e-pasta formā par izmaiņām pieteiktajā uzdevumā
- ✓ Papildus nepieciešama pārdomāta un vienkāršota uzdevumu atlasīšanas funkcionalitāte, kas vajadzīga, lai konkrētos darba uzdevumus būtu vieglāk, ērtāk un ātrāk atrast un apskatīt
- ✓ Atskaišu ģenerēšana, izmantojot uzkrātos datu vēstures informāciju

#### ***5.2.4. Uzdevumu un pakotņu testēšana***

Katrs darba uzdevums, kas ir jānotestē satur vienību testus. Bez testu esamības dara uzdevums nav pabeigts un tiek sūtīts atpakaļ izstrādātājam. Šis process nodrošina vienību testu uzkrāšanu vienotā automatisko testu izpildes sistēmā, kur veicot produkta nodošanu tiek izpildīti visi testi, tādā veidā pārliecinoties par funkcionalitātes darbību. Papildus tiek veikta testēšana, izmantojot dažādus skriptus, kas atvieglo dažādu manuālu darbību veikšanu, līdz ar to automatizējot veicamo darbu testētājam. Sprinta beigās relīzes izgatavošanas procesā tiek veikta integrācijas testēšana. Problēmu gadījumā tiek reģistrēti jauni problēmu pieteikumi, kas jāatrisina pēc iespējas ātrākā laika posmā, lai apmierinātu klienta vēlmes un vajadzības, saņemot funkcionāli strādājošu produktu katrā nodevumā.

#### ***5.2.5. Sadarbības nodrošināšana***

Veicot komunikāciju ar klientu, piemēram, prasību specificēšanas laikā, vai komandas sapulču laikā, tiek veidots sanāksmes protokols kurā tiek uzskaitīti visi veicamie darbi, atbildīgie un noteikti to izpildes termiņi, papildus tiek sniegta noderīga informācija un identificētas un uzskaitītas papildus radušās problēmas. Nodrošinot ciešu saikni ar klientu, iespējams pēc iespējas agrākā posmā identificēt kādas izmaiņas vai jaunas prasības attiecībā uz izstrādājamo produktu. Tāpat bieža komunikācija rada brīvu un ierosinājumiem atvērtu vidi, kas veicina produkta ātrāku izstrādi un nodevumu sagatavošanu. Visi vienošanās punkti, kas ir izsvērti un apskatīti sapulcē tiek uzskaitīti sapulces protokolā, kas pēc sanāksmes tiek izsūtīts visiem dalībniekiem un citām ieinteresētajām pusēm, kas kādu iemeslu dēļ nav piedalījušās sanāksmē. Papildus katrs sapulces protokols tiek uzglabāts vienotajā dokumentu uzglabāšanas sistēmā.

### ***5.2.6. Darba uzdevumu novērtējuma sniegšana un precizitāte***

Neprecīza darba uzdevumu novērtēšana komandas dalībnieku starpā ir vien no lielākajiem riskiem ar ko nākas sastapties sprinta plānošanas procesā. Darbietilpības novērtējums parasti tiek veikts pārāk optimistisks, taču pamatojoties uz uzkrātajiem novērtējuma datiem, iespējams izsvērt un analizēt darba uzdevumu novērtējuma precizitātes koeficientu. Spējās metodes ieviešanas sākuma stadijā darbu uzdevumu novērtēšana bija haotiska, kur minētajiem skaitļiem nebija lielas nozīmes, taču pieradināja komandu pie konkrētas kārtības. Nevarēja paļauties uz dotajiem novērtējumiem, jo parasti darba uzdevumi tika novērtēti pārāk agrā stadijā, kad precīzi veicamie darbi nebija pilnībā skaidri. Īsu sprintu veidošanas gadījumā darba uzdevumu novērtējumam jābūt precīzākam, pieļaujot mazāku procentuālo kļūdas iespējamību. Pieļaujamā novērtējuma atšķirība pret reālā laikā paveikto darbu varētu būt 15 līdz 25 procentu robežās. Lai veiktu precīzāku darbu uzdevumu novērtējumu tiek pielietota kāršu metode, kur katram komandas dalībniekam ir jānosaka cik ilgu laiku konkrētais uzdevums varētu aizņemt. Pielietojot šo praksi novērtējumu precizāte nedaudz paaugstinājās. Ja iterācijas garums tiek samazināts, darba uzdevumu lielums nav tik ievērojams, līdz ar to palielinās darba novērtējuma precizitāte.

Darbu piešķiršana sākotnējā spējās metodes ieviešanas stadijā neveicās diez ko labi, jo komanda skeptiski raudzījās uz jaunu un nepārbaudītu metožu ieviešanu komandas darba procesa organizēšanā. Līdz ar to komandas katra dalībnieka darbu uzdevumus noteica komandas vadītājs. Laika gaitā komanda veicamos darbu pienākumus izvēlējās un sadalīja pati un kļuva pašorganizētāka, jo spējās metode visu dalībnieku starpā uzlaboja savstarpēju komunikāciju un uzticību. Vēsturiskās informācijas uzkrāšana un tālāka tā analīze dod priekšstatu par katra komandas dalībnieka spējām un darbu izpildes efektivitāti.

### ***5.2.7. Metodes mijiedarbība***

Spējo metožu ieviešanas laikā galvenais uzsvars tiek likts uz procesu uzlabojumiem. Secinājumus un ierosinājumus iespējams veikt tikai uz reāliem rezultātiem, ko savukārt iespējams iegūt, novērtējot rezultātu, to mērot, piemēram, problēmu pieteikumu skaitu pret plānoto un reāli izstrādāto (sagatavoto relīzē) skaitu ziņā. Scrum principi nodrošina šīs informācijas uzkrāšanu Sprinta pārskata sanāksmju veidā, kas pārklājas ar CMMI mērījumu veikšanas un analīzes procesu apgabalu.

Zemāk aprakstīta dažādu spējo prakšu un mērījumu procesu apgabalu savstarpēja mijiedarbība CMMI procesu uzlabojumu ietvaros.

CMMI definētie procesu apgabali un prakses	Istrādātās metodes principi un atbilstošas prakses, ievērojot nozīmīgākās Scrum metodes
Uzturēt sākotnēji definētos mērījumus, kas izstrādāti balstoties uz iepriekš uzkrātās informācijas daudzumu, pieredzi	Atbilstoši komandas veidošanas procesam, projekta sākuma fāzē
	Mērījumu sistēma kā dažādu vērtību un mērījumu priekšlikumu apkopojums, kas kalpo tālākai projektu veiksmīgai attīstībai un informācijas uzkrāšanai
Definēt specifiskus mērījumus atbilstoši izvirzītajiem kritērijiem un uzdevumiem	Sprinta atlikušā darba grafika izsekošana
	Relīzes atlikušā darba izsekošana, kas parāda cik daudz no programmatūras funkcionalitātes ir ieviests un vēl nepieciešams izstrādāt
Mērījumu pielietošana un identificēšana	Atbilstoši definētajam komandas veidošanas procesam projekta realizācijas fāzē
Noteikt nepieciešamos mērījumus un veidu kā tas tiks darīts	Atbilstoši Sprinta aktivitātēm un relīzes sagatavošanā atlikušā darba daudzumam, paralēli definējot analīzes procesu, skaidri nosakot arī atskaišu gatavošanas un veidošanas procesu
Iegūt specifiskus mērījumu datus	Ikdienas Scrum sapulces, kur katra sprinta un relīžu paveikto darbu apjoms tiek uzkrāts un atbilstoši arī novērtēts
Iegūto mērījumu interpretācija un analīze	Ikdienas Scrum sapulces, kur katra sprinta un relīžu paveikto darbu apjoms tiek uzkrāts un atbilstoši arī novērtēts
Pārvaldīt un uzglabāt mērījumu datus, konkrētas specifikas, kā arī rezultātu analīze	Definēts komandas veidošanas vai projekta identificēšanas un uzsākšanas fāzē

Atskaišu veidošana no iegūtajiem un analizētajiem statistiskiem mērījumiem priekš visām ieinteresētajām pusēm	Ikdienas Scrum sapulces, kur katra sprinta un reližu paveikto darbu apjoms tiek uzkrāts un atbilstoši arī novērtēts. Sprinta pārskata sapulcē tiek veidota atskaite ar analizētiem statistiskiem mērījumiem
Veidot un atjaunot organizācijas plānošanas un mērījumu veikšanas un analīzes politiku	Definēt specifiskus mērķus politikas plānošanas laikā organizācijā, kas saistītos ar projektu plānu un plānu, kurā būtu uzskaitīti veicamie mērījumi dažādos procesu apgabalos
Mērījumu izpildes un analīzes plānošanas process	Atbilstoši iterācijas sākuma un reližu veidošanas fāzē Definēti mērījumu un uzkrājamo datu pieturpunkti
Deleģēt atbilstošus resursus, lai veiktu procesu mērījumus un analīzi	Scrum plānošanas process, kurā tiek deleģēta komanda un noteikti tās dalībnieki, lai veiktu resursu plānošanu, atbilstoši definētajām lomām un atbildības
Deleģēt atbildības jomas visiem dalībniekiem, kā arī noteikt atbildīgo personu par procesu uzlabošanu, produktu darbu izpildi un ieviešanu, kā arī servisu nodrošināšanu	Resursu lomas ir noteiktas, lai nodrošinātu procesu dokumentācijas papildināšanu. Tiek noteiktas atbildības lomas, piemēram, mērījumu uzkrāšanai, plānošanai un pārvaldīšanai
Veikt cilvēku apmācību	Tiek īstenota komandas apmācību veidošana, analizējot visas nepieciešamības, balstoties uz veiktajiem mērījumiem, analīzes rezultātiem (tiek novērtēti projekta uzdevumi un noteiktas atbilstošas zināšanas, prasmes. Katra komandas dalībnieka zināšanas, izglītības iespējas un darbu uzdevumu izpildes efektivitāte tiek atsevišķi novērtēta.

<p>Darba pakotnes atbilstoši novērtēt pēc veiktajiem mērījumiem, noteikt to prioritātes, kā arī nodrošināt atbilstošu kontroli, izvērtējot definētās prioritātes</p>	<p>Veidot atsevišķas datnes un loģisku datņu struktūru atbilstoši konfigurācijas pārvaldības plānam, izmantot, piemēram, MS Excel kā mērījumu uzkrāšanas rīku, taču nodrošināt tā pieejamību visām iesaistītajām pusēm vienotajā dokumentu sistēmā. Kā papildus izmantojams rīks dokumentu versiju vienotai pārvaldībai, kā arī versiju pārraudzībai un uzkrāšanai, iespējams izmantot Sharepoint, VersionOne u.c. rīkus Uzkrātajai informācijai jāsaturs galvenie projektu mērījumi robežšķirtņu veidā</p>
<p>Identificēt visas iesaistītās puses, veidot to savstarpēju ciešu komunikāciju un līdzdalību mērījumu veikšanas procesos, kā arī projektu plānošanā</p>	<p>Scrum komandas dalībnieki, viņu lomas un atbildības Komandas atskaišu veidošana piedaloties iesaistītajām pusēm kā rezultāts pēc veikto mērījumu analīzes veikšanas</p>
<p>Veikt mērījumu un analīzes procesu pārraudzību un kontroli atbilstoši projekta plānam, nepieciešamības gadījumā tiek veiktas preventīvas un koriģējošas darbības</p>	<p>Scrum skolotājs seko līdzi projekta plāna izpildei un definē veicamos mērījumus kā arī uzkrājamo datu veidus un apjomus, nodrošinot mērījumu un analīzes datu atbilstību projekta plānam</p>
<p>Objektīvi novērtēt un stingri ievērot mērījumu uzkrāšanas un analīzes veikšanas procesu atbilstoši katram procesu aprakstam</p>	<p>Nodrošināt mērījumu veikšanu un rezultātus sprinta izpildes laikā un pārskata veidošanas procesā Nepieciešams pilnībā dokumentēt datu uzkrāšanas un uzglabāšanas procedūru</p>
<p>Pārskatīt aktivitātes, rezultātus un mērījumus ar augstāka līmeņa vadību</p>	<p>Vadībai ir skaidrāks priekšstats par organizācijā notiekošo un saistītajiem procesiem un aktivitātēm, lai atbilstoši reaģētu uz kādu darbību vai bezdarbību</p>

Veicot sprintu plānošanu problēmu pieteikumu atrisināšanas procesā, nepieciešams precīzāk novērtēt darba uzdevumu izpildes termiņus. Situācijā, kad novērtējums nav

iespējams vai ir neprecīzs, sekojot Scrum praksei, nepieciešams iesaistīt komandu, lai tiktu veidots darbu uzdevumu darbietilpības vidējais novērtējums. Šāda prakse dod precīzāku sākotnējo problēmu pieteikumu darbietilpības novērtējumu un attiecīgi uzlabo sprinta plānošanas procesu. Augstāk aprakstīto salīdzinošo metožu savstarpējā mijiedarbība un to ievērošana ir nozīmīga, jo vienlaicīgi tiek ievērota CMMI procesu standartu un procesu labās prakses piemēri, nosakot precīzas aktivitātes un sekojot to ieviešanai, kā arī projektu vadības metodes Scrum un izstrādātās metodoloģijas prakses piemēri, kas palīdz organizēt komandas darbu projektu ieviešanas un uzturēšanas kontekstā.

### ***5.2.8. Pielietotā inženierijas metode***

Ņemot vērā iepriekš definētās prioritātes, tiek ievērotas labās prakses principi kvalitātes nodrošināšanā, galveno uzvaru liekot uz pielietotām inženierijas metodēm:

#### ***Kodēšanas standarti un to ievērošana***

Lai nodrošinātu kvalitātīvu kodēšanas procesu kopumā, piegādājot kvalitatīvu produktu, nepieciešams ievērot kodēšanas standartus uzņēmumā. Kodēšanas standartu ievērošana veicina vieglu koda pārskatāmību, vienotu metožu, klašu un dažādu atribūtu definēšanas veidu, kā arī turpmākajā koda uzturēšanas procesā, vieglāku kļūdu atrašanu un to atrisināšanu.

#### ***Nepārtraukts integrācijas process***

Programmatūras papildinājumu ieviešana ir kā nepārtraukts produkta integrācijas process. Nepārtraukts integrācijas process samazina iespējamību, ka veicot izstrādājamā koda ietveršanu pēdējā brīdī relīzes sagatavošanas procesā, kaut kas varētu nestrādāt vai noiet greizi. Lai atrisinātu šādas iespējamības rašanos, tiek izmantota koda uzturēšanas un pārbaudes veikšanas sistēmas SVN un Code Collaborator. Automatizēta koda pievienošana un apvienošana un nepārtraukta programmatūras veidošana soli pa solim automātiskā režīmā nodrošina relīzes veiksmīgu un ātru sagatavošanu cikla beigās. Uzņēmumā ir ieviesta nepārtraukta programmatūras veidošanas sistēma, kas apkopo visas ietvertās koda izmaiņas, katru vakaru būvējot sistēmu un izpildot relīzes sagatavošanas automatizēta procesa funkciju, kas ir atbilstoši sagatavota, lai veiktu testēšanu.

#### ***Testēšana***

Vienību testus izstrādā programmētāji pirms koda izstrādes, tādā veidā nodrošinot izstrādātā koda atbilstību definētajām prasībām. Izveidotie vienību testi pārklāj visus

izstrādātā koda funkcionalitātes apgabalus. Kā būtisks integrācijas procesa nodrošināšanas veids ir automātisko testu sistēmas esamība uzņēmumā. Automatizēto un sistēmas testu izstrāde nodrošina regresa testu veikšanai īsāka laika nepieciešamību. Automātiskajiem testiem nav jāpārklāj pilnīgi visa funkcionalitāte, kā arī testiem nav jāatkārto lietotāja darbību veikšana. Automātiskie testi pārklāj funkciju testēšanu produkta funkcionalitātes līmenī. Veidojot augsta līmeņa testus, iespējams veikt labāku to pielietošanu nākotnē, piemēram, jau sākotnēji definējot testus bibliotēkās un augstāka līmeņa klasēs.

### **5.3. Procesu novērtējums un uzlabojumi**

#### ***Projektu plānošana un vadība***

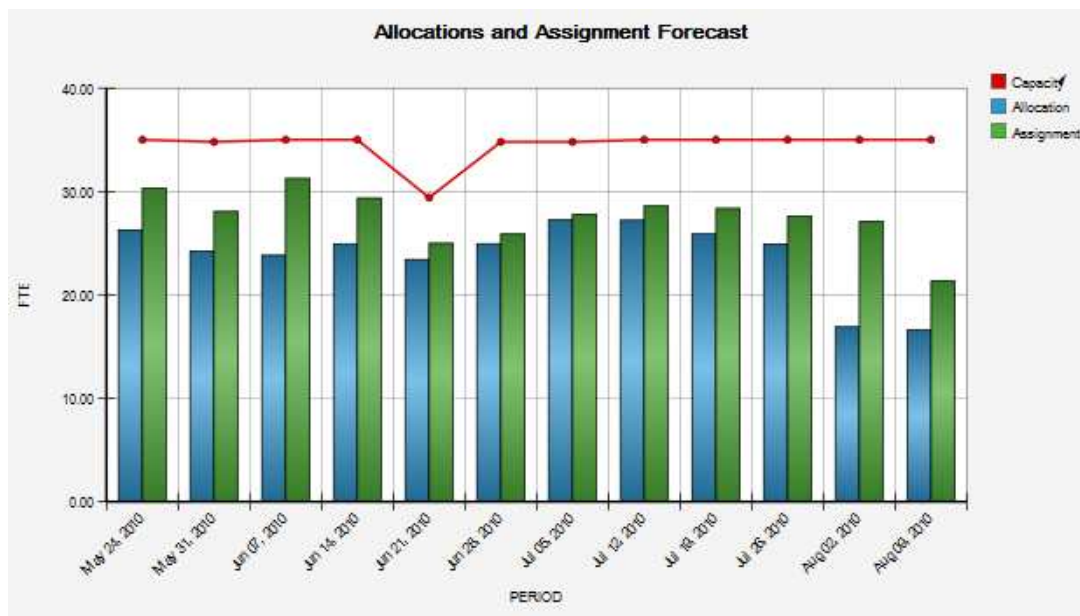
Projektu plānošanā tiek izmantota Clarity programmatūra, kas nodrošina integrētu projektu pārvaldības, finansu uzskaites un kontroles, risku uzskaites un datņu uzglabāšanas funkcijas, kas pilnībā apmierina lielākās izvirzītās definētās prasības. Papildus nepieciešami uzlabojumi dažādu standartu eksportēšanas iespējā, kā arī „viens pret viens” pārveidošana uz MS Project rīku, kur novērojamas vairākas problēmas, kā, piemēram, resursu noslodzes izlīdzināšanas opcijā, kas nestrādā, jo Clarity rīks izmanto vienotu resursu uzskaites sistēmas pārvaldību. Projektu plānošanas procesa nodrošināšanai nepieciešams izvērtēt papildus rīku izmantošanas iespējamību, kā, piemēram, ScrumWorks Pro vai arī papildinājumu integrācija Clarity rīkā, kas nodrošinātu spējas pārvaldības principus projektu plānošanā. Plānošanas process tiek veikts iterāciju veidā, tajā iesaistot visus komandas locekļus, kur katrs komandas dalībnieks veic darba uzdevumu objektīvu novērtējumu tikai situācijā, kad darbu pakotnes ir pietiekami detalizēti aprakstītas. Darba uzdevumus izvēlas katrs komandas dalībnieks atbilstoši savām zināšanām. Situācijā, kad komanda neveic pienākumu sadales un deleģēšanas funkciju, komandas vadītājs izvērtē katra komandas dalībnieka zināšanu līmeni un pieredzi konkrētu uzdevumu veikšanā. Uzņēmumā ir ieviesta komandas dalībnieku zināšanu un pieredzes apgabalu uzskaitē, novērtējums (Pielikums Nr.8), kas ir saskaņā ar procesuālo aprakstu un komandas novērtējuma nepieciešamību, un labo praksi.

#### ***Projektu uzraudzība un kontrole***

Projektu pārraudzība tiek nodrošināta izmantojot paveikto darbu sekošanas sistēmu reizi nedēļā, kā arī katru dienu organizētajās sapulcēs, kuru ilgums nav garāks par 15 minūtēm. Projekta pārskatu sagatavošana augstāka līmeņa vadībai notiek reizi nedēļā, prezentācijas pārskata un telekonferences veidā.

Papildus ir ieviesta nostrādāto darba stundu uzskaites sistēma, taču nepieciešami uzlabojumi automātiskas startēšanās un izslēgšanas veidā. Programmas laika uzskate, kas ir tīmekļa aplikācija, tiek startēta manuālā veidā.

Projektu izmaiņas tiek kontrolētas, izmantojot Clarity programmatūru, kur tiek atjaunota visa ar projektu saistītā informācija, un nodrošina integrētu projektu un resursu pārvaldību, veicot arī dažādu pārskatu ģenerēšanu.



Attēls Nr.5.3.1. „Resursu noslodzes ģenerēts pārskats”

Resursu noslodzes ģenerētajā pārskatā iespējams novērtēt resursu pieejamību pa dažādiem reģioniem, to deleģējumu dažādos projektos.

### ***Konfigurācijas pārvaldība***

Nepieciešams noteikt un vienoties par kopīgu pieeju datu uzglabāšanas un pārvaldības procesā. Tas nepieciešams, lai nepieļautu vienu un to pašu dokumentu uzglabāšanu dažādās vietnēs, kas ir nelietderīga resursu izmantošana. Dokumentācijas uzglabāšanā jāievēro sistēmu hierarhijas un saistīto procesu, un integrācijas nodrošināšana ar citām sistēmām, kā arī versiju kontroles mehānisms. Papildus nepieciešams definēt dokumentu procesu plūsmas mehānismu vienotā dokumentu uzskaites sistēmā.

### ***Kvalitātes pārvaldība***

Starpreģionālā līmenī vienojoties par procesuālajām atkarībām un savstarpējām to saistībām, iespējams veikt atbilstošu procesu dokumentēšanu. Projektu līmenī nepieciešams ievērot kompānijā definētos kodēšanas standartus, kā arī ievērot apstiprinātās dokumentu paraugformas, veidnes, kā, piemēram, Prasību specifikācijas dokuments (Pielikums Nr.1) un tā apstiprināšanas kārtība (Pielikums Nr.2), Prasību izsekojamības matricas dokuments (Pielikums Nr.3), Programmatūras projekta pārvaldības plāns (Pielikums Nr.4), kā arī Pārraudzības un kontroles nodrošināšana, kas ir būtisks kvalitātes procesu ievērošanas pamata princips uzņēmumā. Funkcionalitātes papildināšanas vai problēmu pieteikumu gadījumā tiek izstrādāta atbilstoša dokumentācija, lai atvieglotu darbu tālāku nodošanu

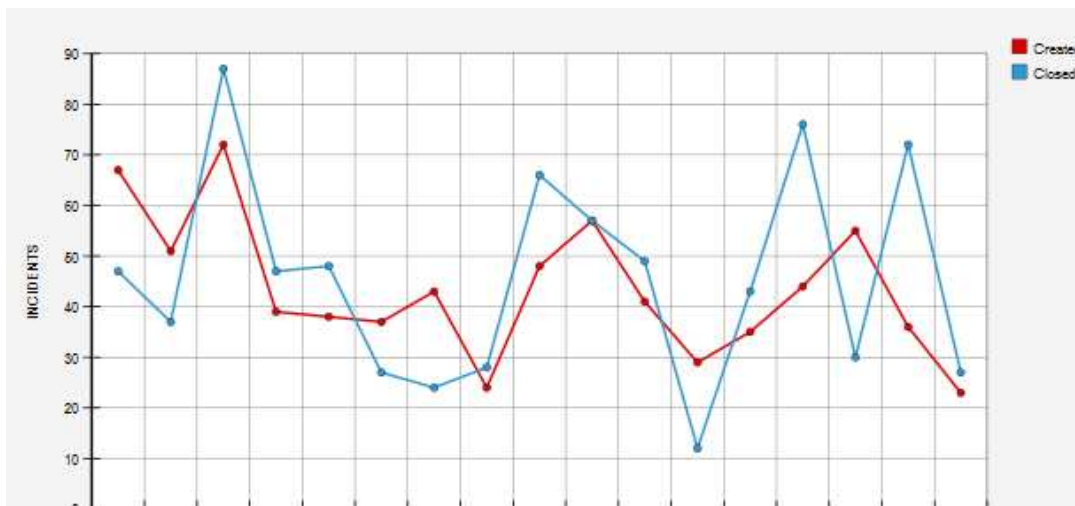
citiem reģioniem (pēc nepieciešamības). Programmatūras kods tiek pārbaudīts un tajā tiek iekļauti skaidrojoši un izprotami komentāri. Komanda darbojas pašorganizēti un pilda tai uzticētos darba pienākumus aktivitāšu un darba pakotņu veidā. Papildus vēl jāizstrādā procesu virzības dokuments ar atbildīgajām personām konkrētos pieturpunktos. Veicot mērījumu novērtēšanu, tiek veikti loģiski un pamatoti secinājumi un pieņemti atbilstoši lēmumi, kvalitatīvi izstrādātu procesu ietvaros.

### ***Piegādātāju izvēle un pārvaldība***

Piegādātāju izvēlē pagaidām tiek ievērot „zemākās cenas” princips. Nav nepieciešamības veikt piegādātāju izvēli pēc „saimnieciski izdevīgākās” metodes. Ir izstrādāta, taču vēl nav apstiprināta darba līgumu un vienošanās paraugforma (no angļu val. – SOW (Statement of Work template). Turpmāko procesu attīstībā nepieciešams veikt šīs formas apstiprināšanu ar augstāka līmeņa vadību.

### ***Mērījumu veikšana un analīze***

Tiek veikti un uzkrāti sākotnējie un paši nepieciešamākie statistiskie mērījumi. Mērījumu uzkrāšana tiek nodrošināta, izmantojot Clarity programmatūru.



Attēls Nr.5.3.2. „Atvērtu pret aizvērtu problēmu pieteikumu skaita analīze pa dienām”

Novērtējot apkopoto statistiku kā veikto mērījumu ieviešanas rezultātu iespējams secināt vai resursu noslodze ir bijusi pietiekama konkrētu problēmu risināšanā. Papildus analīzi iespējams veikt izmantojot problēmu pieteikumu definētajās prioritātēs, kas kalpo kā metrikas nepieciešamo atskaišu ģenerēšanā un secinājumu vai lēmumu pieņemšanas procesā.

### ***Prasību pārvaldība***

Prasību integrētā pārvaldībā nepieciešams identificēt rīku un vienoties par tā vienotu izmantošanu ar citiem reģioniem. Prasību pārvaldības rīka izvēlē būtiskākais princips ir prasību tālāka integrēta saistība ar projekta plānošanas procesu.

### ***Risku pārvaldība***

Riski tiek analizēti konkrēta projekta ietvaros, identificējot cilvēkresursu un aparatūras pieejamību kādas problēmu pieteikuma vai projekta izstrādē. Papildus tiek identificētas arī citas problēmas un riski iterācijas ikdienas 15 minūšu sanāksmēs.

### ***Apmācības***

Tiek uzkrātas visas komandas dalībnieku zināšanas un iepriekšējā pieredze. Projektu plānošanas kontekstā tiek identificēti zināšanu apgabali, kuros nepieciešamas papildus apmācības. Apmācību procesu apstiprina augstāka līmeņa vadība pēc atbilstošas vajadzības un pieprasījuma.

Papildus ir secināms, ka vienas metodoloģijas un standarta ieviešana uzņēmumā vienpersoniski nav iespējama, līdz ar to procesu uzlabošanā nepieciešams iesaistīt vairāku reģionu pārstāvjus. Uzņēmumā nepieciešams veikt komunikācijas uzlabošanas pasākumus dažādos vadības līmeņos. Ieviešot spējās metodes, tiek novērota savstarpēja komunikācijas uzlabošanās komandas iekšienē, komanda ir kļuvusi pašorganizēta.

Procesu attīstība un koordinēta to tālāka virzība ir lielā mērā atkarīga no pareizu un prasībām atbilstošu rīku izmantošanas starp visām iesaistītajām pusēm dažādos reģionos. Jābūt vienotai izpratnei, ar dokumentāru pamatojumu, par rīku izmantošanas veidu, nosakot precīzu, piemēram, lauku aizpildīšanas kārtību – kvalitātes principu ievērošana procesu kontekstā.

CMMI procesu apgabalu novērtēšanā un uzlabošanā visoptimālāko rezultātu iespējams iegūt, analizējot brieduma līmeņus nevis pakāpeniskā, uzskatot to, piemēram, par vertikālu pieeju, veidā konkrēti ievērojot līmeņu robežas, bet veicot visu līmeņu apskatu pa procesu apgabaliem un definējot precīzas vajadzības, kas tieši atbilst kompānijas vajadzībām. Tādā veidā sākuma procesā tiek definētas nelielas un vispārīgas prasības, kas tiek attīstītas visu procesu attīstības ciklu. Šajā gadījumā procesu attīstība tiek virzīta horizontālas līknes novērtējuma kontekstā un, pēc autora domām, ir labāk piemērojama katrai kompānijai atsevišķi, identificējot problemātiskos procesu apgabalus un prioritātes, un uzlabojot tos laika gaitā.

## NOBEIGUMS UN SECINĀJUMI

Izstrādājot šo darbu, autors ir veicis uzņēmumu programmatūras izstrādes un sistēmu uzturēšanas procesu analīzi, atspoguļojot un novērtējot visus CMMI (Capability Maturity Model Integration) pārvaldības brieduma līmeņus un populārākās spējās metodes. Autors ir izstrādājis mijiedarbojošu metodi, kas uzlabo produktu piegādes termiņus, izmaksas un paaugstina kvalitātes nodrošināšanas pasākumus uzņēmumā.

Darba gaitā tika izstrādāti projektu pārvaldības un kontroles procesam svarīgākie dokumentu paraugi, kā, piemēram, Programmatūras projekta pārvaldības plāns, Prasību apstiprināšanas procedūra, Pārraudzības un kontroles nodrošināšana, u.c., kas ir apstiprināti un pielietojami starpreģionālā līmenī.

CMMI definētie procesu apgabali papildina spējās programmatūras ieviešanas metodes un prakses, papildus nosakot nepieciešamību dokumentācijas eksistencei un tās pielietošanā saistītajos procesos.

## IZMANTOTĀ LITERATŪRA

1. **Joseph H. Little**, „The Nokia test”, 2007 [tiešsaiste] – [atsauce 05.05.2010].  
Pieejams: <http://agileconsortium.blogspot.com/2007/12/nokia-test.html>
2. **Artūrs Grīnhofs**, „Uzņēmuma informācijas sistēmu ieviešanas un uzturēšanas procesu analīze”, 2010 [citēts 07.05.2010]
3. **National Defense Industrial Association**, 2010 [tiešsaiste] – [atsauce 29.03.2010].  
Pieejams: <http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Pages/>
4. **SEI, Carnegie Mellon University**, 2008 [tiešsaiste] – [atsauce 15.02.2010].  
Pieejams: <http://www.sei.cmu.edu/cmml/>
5. **NDIA Systems Engineering Division**, “The Effective Use of CMMI®”, 2009 [tiešsaiste] – [atsauce 21.03.2010]. Pieejams:  
<http://www.ndia.org/Divisions/Divisions/SystemsEngineering/>
6. **Wikimedia Foundation**, „Agile software development”, 2004 [tiešsaiste] – [atsauce 01.05.2010]. Pieejams:  
[http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development)
7. **Cockburn Alistair, A. W. Longman**, „Agile Software Development”, 2001 [atsauce 12.04.2010]
8. **Wikimedia Foundation**, „Software development process”, 2001 [tiešsaiste] – [atsauce 15.05.2010]. Pieejams:  
[http://en.wikipedia.org/wiki/Software\\_development\\_process](http://en.wikipedia.org/wiki/Software_development_process)
9. **Bill Krebs**, „XP Evaluation Framework”, 2006 [tiešsaiste] – [atsauce 11.05.2010].  
Pieejams: <http://c2.com/cgi/wiki?XpEvaluationFramework>
10. **Wikimedia Foundation**, „Agile Manifesto”, 2001 [tiešsaiste] – [atsauce 04.04.2010]. Pieejams: [http://en.wikipedia.org/wiki/Agile\\_Manifesto](http://en.wikipedia.org/wiki/Agile_Manifesto)
11. **Ward Cunningham**, „Principles behind the Agile Manifesto”, 2001 [tiešsaiste] – [atsauce 02.04.2010]. Pieejams: <http://agilemanifesto.org/principles.html>
12. **Mike Cohn**, „Toward a Catalog of Scrum Smells”, 2007 [atsauce 12.04.2010]
13. **Wikimedia Foundation**, „Extreme Programming”, 2001 [tiešsaiste] – [atsauce 01.05.2010]. Pieejams: [http://en.wikipedia.org/wiki/Extreme\\_Programming](http://en.wikipedia.org/wiki/Extreme_Programming)
14. **CMMI Product Team**, „CMMI for Development, Version 1.2”, 2006 [atsauce 04.02.2010]

## **PIELIKUMI**

## **Pielikums Nr.1 – Prasību specifikācijas dokumenta apstiprināšana (SRS)**

**Author: A.Grinhofs**

### **Approve SRS**

---

<b>Purpose</b>	The primary purpose of Approve SRS is to ensure that we have full agreement of the critical stakeholders on the content of the SRS. Further, the process is used to clearly mark the document as approved. Future changes to the SRS will only be incorporated when approved by a change process.
<b>Entry Criteria</b>	<ul style="list-style-type: none"><li>• A draft SRS document is ready for review</li></ul>
<b>Inputs</b>	<ul style="list-style-type: none"><li>• The SRS document</li><li>• A list of any known issues</li><li>• A list of approvers. As a default list, the stakeholders who will approve key project documents are:<ul style="list-style-type: none"><li>• Software Engineering Management</li><li>• Software Project Manager (SPM)</li><li>• Applications Lead / Manager</li><li>• Program Management</li><li>• Quality</li><li>• Security</li></ul></li></ul>
<b>Activities</b>	<ol style="list-style-type: none"><li>1. Distribute / send a pointer to the document and any known issues for review to the approvers</li><li>2. Hold a review (either email or meeting), identifying any remaining issues or defects</li><li>3. Address issues and defects</li><li>4. Distribute an updated document (repeat steps 2-4 until all are satisfied with the results)</li><li>5. Gather approvals, either via a signed version of the document or via email.</li><li>6. Store the approved version of the document in the SharePoint Project Folder</li><li>7. Publish a new “Major Version” of the document</li><li>8. Notify approvers of the availability of the approved version of the document</li></ol>

---

---

**Outputs**

- An approved document
- 

**Roles &****Responsibilities**

- SPM: Ensure the document is distributed for review to the stakeholder, manage the review and update process, store the updated version in SharePoint Project Folder, publish the new Major Version, Notify stakeholders of the release.
  - Approvers – Read the document, provide comments, review updates and when the document is ready provide timely approval.
- 

**Tools &****Techniques**

- SharePoint for document storage and management
- 

**Exit Criteria**

- An approved document in SharePoint with a new Major Version number
-

## Pielikums Nr.2 – Prasību specifikācijas dokuments (SRS)

**Author: A.Grinhofs**

	<<Product Name>>  Software Requirements Specification
<b>Document Version #</b>	
<b>Author:</b>	
<b>Corresponding Market Requirements Document (including version)</b>	
<b>Other Referenced Source Documents</b>	

<b>Approvals:</b>	
<b>Engineering Management</b>	
<b>Product Marketing Applications</b>	
<b>Quality</b>	
<b>Release Date:</b>	

Requirement ID	Requirement statement	Group	Sub-Group	Date of Last Change	Priority	Source

## Pielikums Nr.3 – Prasību izsekojamības matricas dokuments

(Traceability matrix)

Author: A.Grinhofs

	<b style="color: red;">Market Requirments Document to Software Requirements Specification Traceability Matrix</b>
<b>Purpose</b>	The Requirements Traceability Matrices provides a means of ensuring that all of the items from the MRD that are allocated to software are in fact addressed in the SRS (no missed requiriements) and likewise that all of the SRS requirements correspond to MRD requirements (no extra features)
<b>Product (including phase):</b>	
<b>Traceability Matrix Version:</b>	
<b>Date:</b>	
<b>Author:</b>	
<b>Source Market Requirements Document (including version)</b>	
<b>Source Software Requirements Specification (including version)</b>	

### Market Requirments Document to Software Requirements Specification

<b style="color: red;">Market Requirements Document</b>	
<b>MRD Section #</b>	<b>Text Description of MRD Requirement</b>
<b style="color: blue;">Software Requirments Specification</b>	
<b>Corresponding SRS Requirement ID(s)</b>	<b>Text Description of SRS Requiriement</b>

## Pielikums Nr.4 – Programmatūras projekta pārvaldības plāns

Author: A.Grinhofs

<Project Name>

### Software Project Management Plan

Version : <Version #>

Document Id : SPMP-XXXX

Revision Date : MM/DD/YY

### Document History & Approvals

Revision	Date	Author	Description of Major Changes

### Release Revision Being Approved:

Role	Name	Signature (indication of email approval is acceptable)	Date
Software Project Manager			
Project Management			
Senior Director of Software			
VP of Engineering			

## **Contents**

1.	Introduction	75
1.1	Project Overview	75
1.2	Project Deliverables	76
1.3	Management Objectives and Priorities	76
1.4	Assumptions, Dependencies, and Constraints	77
1.5	Definitions and Terms	77
2.	Project Organization	77
2.1	Organizational Structure, Roles, & Responsibilities	77
2.2	Organizational Boundaries and Interfaces	77
2.3	Stakeholders & Stakeholder Involvement	78
3.	Resources	79
3.1	Staffing Plan	79
3.2	Knowledge, Skills, & Training Required	80
3.3	Other Resources	80
4.	Monitor and Control	81
4.1	Regular Reviews	81
4.2	Milestones Reviews	81
5.	Risk & Issue Management	81
5.1	On-going Risk & Issue Management	81
6.	Technical Processes	82
6.1	Life Cycle Model	82
6.2	Methods, Tools, and Techniques	82
6.3	Project Support Functions	82
6.3.1.	Configuration Management	82
7.	Schedule and Budget	83
7.1	Schedule	83
7.2	Budget	83
8.	References	83

## **Introduction**

### **Project Overview**

< Provide a brief summary overview of the project, including items such as the name of the project, its intended market(s), and the key market needs that are being addressed by this work. If appropriate, reference the MRD that this project is addressing. Briefly identify the

major work activities that are included in the span of this project. Identify the expected duration and target date for completion of the project.>

**Project Deliverables**

< Identify the outputs of the project, the recipients, and how the deliverable will be provided. Include major intermediate deliverables relied on by other organizations (e.g., Applications, Quality / Validation, Pilot, Project Management), as well the final deliverables. Typically, this will include items such as documentation (for other HC organizations & the customer), source code, executable code, and status information for Project Management. This can be provided in a table format, as below.

Project Deliverable	Recipient(s)	Method of Delivery

>

**Management Objectives and Priorities**

< Describe the objectives and priorities for the project. Topics may include, but are not limited to, the relative priorities among, and trade offs between, requirements, schedule and budget. Issues such as long term maintainability or the ability of the resulting product to serve as a foundation platform for further work can also be addressed, as appropriate.

Sometimes the following table is useful (note that only one item can be the top priority, one item the second level priority, etc.) :

Tradeoff Priorities		
	Rank Order Priority	Notes / Comments
Schedule		
Delivered Functionality		
Development Costs		
Product Costs		
<addition items, e.g., Maintainability, Reusability		

>

>

### **Assumptions, Dependencies, and Constraints**

< State the assumptions on which the project plan is based, the external events the project is dependent upon and the constraints under which the project is to be implemented.

Assumptions and Dependencies may include items such as availability of internal resources, the ability to hire new people in a timely fashion, or the relative difficulty of key technical tasks. They may also include the ability to acquire, modify, or use existing Software. If there are assumptions that the product functionality will be delivered in phases, identify this and at least point to where the stages are defined. Technical Constraints are key environmental technical or managerial conditions that the Software that the project is assuming are “givens” that are limiting choices in some significant way (e.g., the use of the Linux OS, the Marketing Window of Opportunity). >

### **Definitions and Terms**

< Define key terms and acronyms used in the document. >

### **Project Organization**

Organizational Structure, Roles, & Responsibilities

<Describe the internal organization structure of the project. Identify who will be performing the Software Project Manager role and any other project-related leadership roles (e.g., Technical Lead / Architect, Test Lead, CM Lead, Site Leads, Integration Lead, Outsourcing Lead). Briefly identify the responsibilities and key deliverables for each of these roles. If there are key roles from the larger project, for example a System Architect, or overall project / program manager these people can also be identified.

Organizational charts, matrix diagrams, or tables may be used to depict the information.>

### **Organizational Boundaries and Interfaces**

<Describe the boundaries between the Platform Software project team and each of the following entities:

Marketing

Applications

Quality (Validation)

Project Management

Hardware

Subcontracted organizations

Any other organizations that impact the ability to deliver the project successfully, (e.g., HR if hiring, Legal if contracts are required, etc.)

Identify the point of contact within the project team for each of these organizations and who in those organizations are the Software Project Team’s point(s) of contact. Either identify the expected information deliverables to and from the other organization or, as may be the case with subcontracted organizations, provide a link to the document that describes the respective deliverables. If the information is needed by a particular date, include this with the expected deliverable. Note that one deliverable can be regular status information.

A table, such as the following can be used to capture this information:

Organization	Project / Organization POCs	Project Deliverables to Organization	Organization Deliverables to the Project
Marketing			
Applications			
Quality			
Project Management			
Hardware			
Sub – Wind			
Sub - Datamat			
...			

POCs = Points of Contact

>

### **Stakeholders & Stakeholder Involvement**

<This section identifies the key stakeholders and how these stakeholders are expected to participate in document reviews & approvals and project status and milestone reviews.

Pay particular attention to the participation of people from other Organizations such as Applications, Quality, Project Management, Hardware. You can use tables such as the following to capture this information:

Stakeholder Involvement							
Stakeholder / Doc & Reviews	SRS	FDS	SPMP	Test Plans	...	Project Status Reviews	Mileston e Reviews
Engineering VP – Bob							
SW Director – Nassrin							
SPM – xxx							
Project Manager – xxxx							
Marketing - xxx							
Quality– xxx							
Applications – xxx							
...							
...							

Suggested Involvement Codes: R=Review, A=Approve, C=Contribute, I=Inform, P=Participate

>

## Resources

### Staffing Plan

<Identify the HC sites that will provide resources to the project and the main focus of each of the sites. For each site, identify the specific people who will be participating in the project (both HC employees and contractors) and the percentage of their time that is allocated to work on this project (full time, with no other project or sustaining responsibilities = 100%). If there are staffing ramp up and ramp down profiles, use a table or other graphical representation to capture this information.

Identify any subcontracted organizations, the number of people that they are providing (if this is known) and the main project work focus of the organizations.

If the staffing plan information will be changing on a regular basis, it may be advantageous to keep this information in another document, which can be updated independently of this overall Plan. If you decide to do this, you can reference the other document in this section.

>

**Knowledge, Skills, & Training Required**

<Identify the Knowledge and Skill gaps or deficiencies that must be addressed to allow the identified team members (both HC employees and contractors) to be effective, contributing project members, given the tasks they are going to be assigned. Identify the individuals (or groups) and their gaps. Consider product technology, tools, and process gaps / shortfalls. For each of the knowledge and skill gaps, indicate how this will be addressed (e.g., formal training, reading, tutoring / mentoring). If the method used will require time and effort by an existing team member, as in teaching a class or tutoring / mentoring identify the person from the project who will provide it and an approximate estimate of the number of hours that will be involved (ensure that this is incorporated into the schedule and resource loading) You can use tables such as the following to capture this information:

Team Member / Group	Skill or Knowledge Gap Description	Method to Address	Provider	Planned Dates
aaaa				
bbbb				
cccc				
dddd				
eeee				

>

**Other Resources**

< Identify other (e.g., hardware, development equipment, servers, software) resources that are required to successfully specify, design, build, integrate, test, document, and deliver the project. Highlight any of these that are not currently available and when they will become available. >

## **Monitor and Control**

### **Regular Reviews**

< Identify how often project status reviews will be held (note that attendees for the project status reviews were identified in the stakeholder section).

Identify the standard information that will be presented. If available, this can simply be a link to the standard presentation template. Note that a standard template is currently under construction.>

### **Milestones Reviews**

< Identify the major milestone reviews that will be held during the life of the project. The number and types of milestones are dependent on the project life cycle used.

Identify the Readiness checklist that will be used to determine if the Milestone is complete. If available, this can simply a link to the standard checklist.>

## **Risk & Issue Management**

### **On-going Risk & Issue Management**

< Risks and Issues will occur and be resolved over the life of the project. This section identifies how project risks and issues will be managed by the project.

Standard risk source categories that should be considered when identifying risks include, but are not limited to:

Staffing

Product Hardware availability

Supplier / Subcontractor

Technical (product technologies)

Equipment & other development or testing environment resources

Software required to perform the project

Standards & Interfaces (e.g., changing or unstable)

The standard approach is for risks and issues to be captured in Clarity, monitored as a regular part of the standard project reviews, and worked to closure. Risk and issue management is currently an element of the Monitor and Control process (which is currently being defined). If you will be handling risks and issues as described in the standard organizational process, simply indicate that here by referencing the organizational process. If you will be using an alternative approach or process, describe that here.>

## **Technical Processes**

### **Life Cycle Model**

<Define the overall project life cycle model or models that will be used for the project. Typical examples are Iterative, Agile, “V”, Prototype, Spiral, and Waterfall. >

### **Methods, Tools, and Techniques**

< Identify the hardware and software environments, development methodology (if any), programming language(s) and other notations, tools (e.g., design tools, compilers, static code analyzers, integration and test tools) and techniques (e.g., requirements, design, and code reviews) that will be used to specify, design, build, integrate, test, document, and deliver the project deliverables.

Identify what intermediate work products will be produced during specification, design, building, integration, testing, documentation, and delivery of the project. Typical examples include items such as the Software Requirements Specification, a Functional Design Specification, Unit Test Cases, Integration & Test Plans, Test Cases, and Test Reports.

In general, you should include a section on how testing and integration will be performed within the software project. As a part of this section, ensure that you address the approach that will be used to integrate and test any outsourced work.>

## **Project Support Functions**

### **Configuration Management**

< Define how configuration management will be done for the project.

Identify what work products will be controlled and their level of control (will they be formally approved and released, or will they be maintained and updated at the discretion of the project.) Also identify where the different work products will be kept, and if appropriate made visible to the project (e.g., folders in SharePoint, Subversion, Clarity). Identify who should be notified when changes are made.

Identify what baselines will be established and what is included within each of the baselines (e.g., source code, executables, internal work products associated with code, build tools including their versions, user documentation)

If a more formal change control process is identified for a work product, identify how this change control will be managed. Will a “change approval board” review and approve changes before they are implemented? If so, who is on the board (it could be as few as 2 people) and how will change requests be submitted and addressed.>

## **Schedule and Budget**

### **Schedule**

<Capture the major milestones and their planned dates for the project. Typically, there are 3 – 7 milestones. Do not put in a detailed schedule. Instead, reference where the detailed Project schedule is maintained. >

### **Budget**

<If there is one, provide a high level summary of the defined budget for the project, possibly broken down by categories (e.g., labor, expensed equipment, software, ...) If this is described in another document, simply include a reference to the document. If there is no specific budget for the project, identify this.>

### **References**

<Capture the full name and version of any referenced documents. if possible, include a hyperlink to where the document can be obtained. >

## Pielikums Nr.5 – Pārraudzības un kontroles nodrošināšana

Author: A.Grinhofs

### Monitor and Control the Project Process Description Draft Version 0.1; March 11, 2010

This document contains the description of the Monitor and Control process.

#### Monitor and Control the Project – Process Description

---

<b>Purpose</b>	The purpose of Project Monitoring and Control is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan
<b>Entry Criteria</b>	<ul style="list-style-type: none"><li>• The Project Plan is complete and under appropriate CM Control</li><li>• Schedule and resource information is in Clarity</li></ul>
<b>Inputs</b>	<ul style="list-style-type: none"><li>• Project status information</li><li>• Current Action Item list &amp; status</li><li>• Current Risk list &amp; status</li><li>• Improvement Opportunities &amp; Recommendations</li><li>• Readiness &amp; Completion Checklists (Milestone reviews)</li></ul>
<b>Activities</b>	<ul style="list-style-type: none"><li>• Hold Weekly Project Reviews</li><li>• Hold Milestone Reviews</li><li>• Hold Management Reviews (Senior Management, CEO)</li><li>• Manage Actions to Closure</li></ul>
<b>Outputs</b>	<ul style="list-style-type: none"><li>• Collected project status information</li><li>• Updated Action Item &amp; Risk lists</li><li>• Minutes capturing decisions</li></ul>
<b>Roles &amp; Responsibilities</b>	<ul style="list-style-type: none"><li>• <u>Developers</u>: Enter time / task status information into Clarity; Provide other information as requested by SPM</li><li>• <u>SPM</u>: Gather and present inputs during review meeting; Participate in decision making</li><li>• <u>Management</u> – Review information; Provide Guidance; Make</li></ul>

---

	<p>decisions; Provide support where required; Present summary status to higher levels of management</p> <ul style="list-style-type: none"> <li>• <u>Action item assignees</u> – work item to resolution and report the status</li> <li>• <u>Minute taker</u>: Capture action items &amp; meeting minutes</li> </ul>
<b>Tools &amp; Techniques</b>	<ul style="list-style-type: none"> <li>• Clarity, for tracking schedule status, action items, risk items, incidents / defects</li> <li>• Project Review Template for presentation (PowerPoint)</li> <li>• SharePoint for storage of status information and minutes</li> </ul>
<b>Exit Criteria</b>	<p>Project work complete and delivered with the planned levels of functionality and quality</p>

### 1. Hold Weekly Status Reviews – Activity Description

<b>Activity Description</b>	<p>To ensure that management gets a consistent and complete view of the project status, the Weekly Project Review template should be used as the basis of the presentation. Additional detail information can be provided, as appropriate to clearly communicate the status or details of program or technical issues.</p> <p>Use Clarity reports to generate the schedule detail, Action Item, Issue, and Risk information slides</p>
<b>Required Practices</b>	<p>SPM responsibilities:</p> <p>Process Startup:</p> <ul style="list-style-type: none"> <li>• Establish, with management, a set of required and optional review participants</li> <li>• Establish, with management, a schedule and location for the meeting and distribute to required and optional attendees</li> <li>• Establish (or identify if one already exists) a SharePoint folder to store status slides and minutes</li> </ul> <p>Prior to each meeting</p> <ul style="list-style-type: none"> <li>• Gather information described in the Activity description into a PowerPoint presentation.</li> <li>• Identify someone to capture meeting minutes, and new Action Items, Issues, and Risks</li> </ul>

---

Following each meeting

- Ensure that final slides and minutes are stored in SharePoint
  - Ensure that new Action Items, Issues, and Risks are stored in Clarity
- 

**Optional**

**Practices**

**Implementation**

**Guidance**

---

Clarity is an essential resource for the information presented during the meeting. It is therefore important to ensure that the project members understands that the following information and activities are captured in the tool:

- Task completion data
- Action items and action item status
- Issues and associated action item status
- Risks and associated risk

If Clarity is not used to capture this information, the accuracy of communications with management will be severely hampered.

---

## Pielikums Nr.6 – Procesu novērtējums SIA „HC Spania”

**Author: B.Delgado**

Šajā pielikumā esošo procesu apraksts saņemts no Spānijas HC biroja, projektu vadītāja B.Delgado. Pielikums iekļauts kā salīdzinājums starp Latvijā un citos reģionālajos birojos identificētajiem, analizētajiem procesiem.

Below you can find a list with important processes, which are in use or missing in R&D Applications Spain team. Processes used by other Spanish departments are not considered.

We have to fulfil ISO 9001:2000 certification that HC Spain has, which is audited every year. Process to be followed by R&D Applications Spain is defined in PGC-02. Any modification to the process or tools specified in it has to be managed with Spanish Quality & Local Operations team.

<b>Tool</b>	<b>Remark</b>
Requirements management	
PGC-02 (ISO 9001:2000 certification): D-140-K form	Project requirements gathering. According to PGC-02, initial and new requirements can be specified in some tables of D-140-K form or through documents attached to it.
Planning	
PGC-02 (ISO 9001:2000 certification): D-140-K form APPD1 form MS Project 2003	Project management. According to PGC-02: D-140-K form has to be used mandatorily APPD1 form has to be used and signed for Application Development Commitment if project exceeds 160 h MS Project has to be used for project planning if the project exceeds 320 h. If less than 320 h are required, planning can be managed either using MS Project or with a tasks table inside D-140-K form These documents and other documents related to the project

	have to be stored in the local network in a specific folder per project
Development	
SW backup process	Missing process According to PGC-02, it is mandatory to do a backup of the SW delivered as a result of any project, but the way to do it or where to store it is not specified.
Measurement and Analysis	
Euro 6000 - 1	Control of EURO 6000 Application Incidences reported by Clients (according to PGC-02 process related to Spanish ISO 9001:2000 certification)
KPI 2-1	Percentage of projects closed according to its plan (according to PGC-02 process related to Spanish ISO 9001:2000 certification)
KPI 2-2	Customer validation control (according to PGC-02 process related to Spanish ISO 9001:2000 certification)
KPI 2-3	Percentage of customer deliveries (internal and external) according to their plan (according to PGC-02 process related to Spanish ISO 9001:2000 certification)
Project Monitoring and Control	
PGC-02 (ISO 9001:2000 certification): D-140-K form	The D-140-K form has a part dedicated to project revision and according to PGC-02, it is mandatory to have a regular project revision with a periodicity of at least one month. As the D-140-K of each project has to be stored in the local network, this kind of report is available for anybody interested in it.
Specific project reports	Outside of PGC-02 guidelines, there are weekly reports of specific projects according to high management requests. Currently R&D Applications Spain provides weekly reports about Vertical Markets projects and about Spanish Applications Porting project.

Issue tracking	
S EMEA Issues priority management	Process established by Simon Rossi to set priorities to S EMEA Optimum incidents, with a regular review (around each 2 weeks)

## **Pielikums Nr.7 – Darbu izpildes novērtējuma anketa projektu ietvaros**

### *Task Progress Accuracy Survey*

Question 1 - Without referencing your timesheet or task list in Clarity, please list your current workload. What projects are you working on and what are you doing for those projects?

Question 2 - After listing this work, look at your timesheet. Is all of this accounted for your Clarity timesheet in some way? If not, how is it different? Missing tasks? Too high level? Too detailed?

Question 3 - Does the amount of hours you are putting on a task and the amount of hours remaining on that task seem to accurately reflect your task's progress? Is the ETC (remaining work) value usually correct?

Question 4 - Do you work on your tasks in the timeframe indicated on the timesheet as the Task Start and Task Finish, or do you generally ignore those dates and just get the work done as you can?

Question 5 - How are your task priorities communicated to you? Your manager? Clarity? Please explain.

## Pielikums Nr.8 – Darbinieku zināšanu un prasmju dokuments

Name	Employment since			Education Level			Language knowledge				Experience					Programming Languages			
	Years in Hvc	Bachelor	Masters	Doctor	Latvian	English	Russian	other	AppMngnr	HFC	HSDK	AAE	Middleware	Linux	C/C++	Java	Perl	Assembler	
	12.11.2003	7	x			x	x	x							x	x		x	
	02.03.2009	1	x			x	x	x	x						x				
	21.12.2009	1		x		x	x	Hebrew							x				
	30.03.2009	1		x		x	x		x				x		x				
	25.10.2006	4	x			x	x								x			x	
	29.10.2007	3		x		x	x		x				x					x	
	23.03.2009	1	x			x	x							x	x				
	06.10.2006	4	x			x	x						x		x				
	03.10.2005	5	x				x								x				
	01.08.2005	5		x			x												
	06.04.2009	1	x			x	x							x	x			x	
	01.09.2005	5	x			x	x							x					
	04.05.2004	6		x		x	x							x	x			x	
	28.10.2002	8		x		x	x							x				x	
	06.12.2006	4	x				x		x					x					
	25.09.2006	4	x				x						x					x	
	09.10.2006	4		x			x							x					
	19.09.2005	5		x			x							x				x	
	25.09.2006	4		x			x							x				x	
	01.11.2006	4	x				x											x	
	09.11.2009	1				x	x		x					x				x	

## DOKUMENTĀRĀ LAPA

Maģistra darbs: **CMMI un AGILE metodoloģijas ieviešana uzņēmumā**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: \_\_\_\_\_ / A.Grīnhofs

(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **piemērotu/nepiemērotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs(-ja): \_\_\_\_\_ / M.Treimanis

(Vadītāja paraksts)

Darbs iesniegts **maģistrantūras sekretariātā** \_\_\_\_\_.

(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: \_\_\_\_\_.

(Metodiķes paraksts)

Recenzents: \_\_\_\_\_ / M.Linde

(Recenzenta paraksts)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

\_\_\_\_\_ prot. Nr. \_\_\_\_\_, vērtējums \_\_\_\_\_

(Darba aizstāvēšanas datums)

Komisijas sekretārs: \_\_\_\_\_

(Sekretāra paraksts)