TALLINNA TEHNIKAÜLIKOOL
Informaatika teaduskond
Informaatika instituut
Teadmussüsteemide õppetool

# Scrum testimisprotsessi struktureerimine Ignite OÜ näitel

Magistritöö

Üliõpilane: Nelli Licht
Üliõpilaskood: 111700IABM
Juhendaja: Jaak Tepandi

Tallinn
2015

# Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

---------- ----------

(*kuupäev*) (*allkiri*)

# Annotatsioon

Käesoleva töö eesmärgiks on luua agiilset tarkvaraarendusmetoodikat Scrum kasutavale ettevõttele struktureeritud testimise protsess. Lõputöö kirjutamise hetkel on ettevõttes juurutatud Scrum metoodika ning projektipõhiselt on kasutusel erinevad kvaliteeditagamise praktikad, mis on kohandatud vastavalt projektidele, kuid puudub ühine struktuurne lähenemisviis. Seega on tekkinud seis, kus uutel meeskondadel ja ka uute projektidega alustavatel meeskondadel võtab kohanemine palju aega ning selle aja vältel langeb pakutava teenuse kvaliteet. Samuti on kvaliteedispetsialistide koolitamine raskendatud, kuna testimistegevused ei ole läbipaistvad ja puudub ühtne visioon kvaliteeditagamise metoodikast.

Ettevõtte testimisprotsessi struktureerimiseks on antud töös valitud Demingi ringist lähtuv metoodika ning protsessi ja tulemite mõõtmiseks valiti TPI *Next* mudel. Mudeli alusel kaardistati ettevõtte testimisprotsessi probleemkohad ning loodi parendamise skoop. Lähtudes valitud skoobist koostati praktikate analüüs probleemkohtadele lahenduste leidmiseks ning seejärel rakendati neid reaalse projekti näitel.

Lõputöö raames on kirjeldatud Ignite OÜ testimisprotsess, mis on loodud lõputöö mahus läbitöötatud kirjanduse ja olemasoleva protsessi alusel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 99 leheküljel, 9 peatükki, 8 joonist, 12 tabelit.

# Abstract

The purpose of this thesis is to create a structured testing process for a company which has introduced Scrum methodology. Scrum methodology does not describe how testing discipline should integrate into the development process. Therefore the development teams in the company have been practicing the usage of different testing approaches or they have been missing in general. All in all this situation has created problems with the quality of the product under development and in the beginning of new projects quality assurance discipline is always created and adapted again. Also, the training of quality specialists has become a problem because the company is using a variety of different approaches and does not have one specific vision about quality assurance methodology. To overcome the described problems a structured testing process has to be created.

In order to create a structured process, a methodology which is based on Deming ring was chosen. TPI Next model was used to evaluate the existing process and based on the goals set by the company, the scope for the improvement was created. On the basis of the chosen scope, a research of best practices was conducted and the improvement proposals for the problematic areas were put together. Created proposals were put into practice on the example of a project and a case study was created.

As a result of the case study and the conducted research a description of a suitable test process for Ignite OÜ was created.

The thesis is in Estonian language and contains 99 pages of text, 9 chapters, 8 figures, 12 tables.

# Lühendite ja mõistete sõnastik

**TPI *Next*** ***Test Process Improvement Next***
Testimisprotsessi parendamise mudel ja metoodika.

**XP** ***Extreme programming***
Iteratiivse arenduse metoodika [19].

**RUP** ***Rational Unified Process***
Iteratiivse arenduse metoodika ja raamistik.

**CMMI** ***Capability Maturity Model Integration***
Küpsusmudel ja raamistik, mille abil on võimalik hinnata arenduse küpsustaset.

**TMMi** ***Test Maturity Model integration***
Testimisprotsesside parendusmudel.

**ISTQB** ***International Software Testing Qualifications Board***
Rahvusvaheline organisatsioon, mis tegeleb kvaliteedispetsialistide koolitusprogrammide loomise ja sertifitseerimisega.

**ITIL** ***Information Technology Infrastructure Library***
Infotehnoloogia teenusehalduse raamistik.

**SLA** ***Service Level Agreement***
Teenusepakkuja ja kasutaja vaheline leping, mis fikseerib pakutava teenuse kvaliteedinäitajad.

# Jooniste nimekiri

# Tabelite nimekiri

# Sisukord

# 1. Sissejuhatus

Tarkvaraarenduse tempo on muutunud üha kiiremaks ja ärivajadused muutlikumaks, seega on infotehnoloogia üksustele esitatud väljakutse, kuidas toetada nõudlikku äripoolt. Konkurentsis püsimiseks peavad ettevõtted projekte kiirelt ja efektiivselt ellu viima. Antud probleemi lahendamiseks on laialdaselt kasutusele võetud agiilsed tarkvaraarendusmetoodikad [26]. Seda tendentsi näitab aina kasvav ja laienev agiilne kommuun, kui ka suurenev nõudlus spetsialistide järgi, kellel oleks kogemust mõne agiilse metoodikaga. Eestis propageerib agiilsust suur osa erasektori IT ettevõtetest ning samuti on muutunud tavapärasemaks, et ka avalik sektor eelistab agiilmetoodikaid, näiteks üha sagedamini nõutakse projektihangetes agiilsete arendusmetoodikate kasutamist ja teadmuse olemasolu.

Viimastel aastatel on sõna „agiilne“ olnud väga populaarne. Erinevate konverentside, töötubade ja koolituse kaudu on laiale kuulajaskonnale edastatud infot, millist kasu pakub agiilne tarkvaraarendus ettevõtete äripooltele ja IT üksustele. Samuti leidub palju erialast kirjandust ja laialdaselt parimate praktikate soovitusi, kuidas muuta efektiivsemaks ja kvaliteetsemaks arendajate tööd. Samas vähe on kirjeldatud, kuidas jagunevad ümber meeskonna rollid ja kohustused kvaliteedi tagamise ja testimise perspektiivist, võrreldes traditsionaalsete ehk iteratiivsete, inkrementaalsete või kosk-mudeli (*waterfall)* metoodikatega. Näiteks on traditsionaalsed kvaliteeditagamise protsessid selgelt struktureeritud ja sõltuvad tugevalt dokumentatsioonist ning nõuavad rangelt selle loomist ja järgmist, kuid agiilsete põhitõdede alusel luuakse dokumentatsiooni täpselt nii palju kui vaja, kuid nii vähe kui võimalik. Seega võib väita, et võttes kasutusele agiilsed metoodikad, peavad seni kasutusel olnud testimisprotsessid muutuma ja kohanema uue keskkonnaga.

Kuid see ei ole triviaalne ülesanne, kuna agiilsed metoodikad (seal hulgas Scrum) ei defineeri, kuidas ümber korraldada teadaolevaid testimisprotsesse, seega on tekkinud hägune seis ja segadus, milline peaks olema kvaliteedispetsialistide roll. Eriti tugevalt tuleb probleem esile Scrum metoodikaga, kuna metoodika kirjelduse kontekstis ei ole kvaliteeditagamise tegevusi peaaegu üldse puudutatud ning keskendutakse meeskonna kompetentsile. Näiteks Scrum metoodika alusel on meeskond multifunktsionaalne ning eraldiseisvat kvaliteedispetsialisti rolli tiimis ei eksisteeri, samuti on mitmed valdkonna spetsialistid väitnud, et antud protsessis ei olegi kvaliteedispetsialiste vaja, kuna efektiivne protsess tagab kvaliteetse arenduse ka ilma

kvaliteedispetsialistita [15]. Testimist viiakse läbi pidevalt, kogu projekti vältel, mitte projekti lõpus, samuti on ette nähtud, et arendajad viivad läbi testimist ning toetavad erinevaid kvaliteeditagamise tegevusi. Testimine on arendusprotsessi üks põhilisi kvaliteeditagamise vahendeid. Sellest vajadusest lähtudes vaatleb käesolev lõputöö konkreetset tarkvara arendusmetoodikat testimise perspektiivist.

## 1.1 Taust ja probleem

Käesolev töö on vajalik, et luua Scrum tarkvaraarendusmetoodikat kasutavale meeskonnale sobiv testimisprotsess, kuna antud metoodika ei kirjelda testimist ega selgita kvaliteedispetsialisti rolli. Töö põhiprobleemiks on ettevõtte testimisprotsesside projektipõhisus ja ühtse kontrollitud korralduse puudumine. See tingib probleemid arendatava tarkvara kvaliteediga, raskused uute meeskonnaliikmete kaasamisel ning koolitamisel. Lõputöö raames leitakse meetod ettevõtte testimisprotsessi analüüsiks ja struktureerimiseks. Analüüsitakse ettevõtte olemasolevat testimisprotsessi, hinnatakse seda ning koostatakse parendamise soovitused ja nende alusel parendatakse ühe meeskonna protsessi. Lisaks luuakse antud töö käigus Scrum meeskonna kvaliteedispetsialisti juhend, mille hulgas on kirjeldatud erinevad testimisprotsessi osad, kvaliteedispetsialisti roll ja tema kohustused. Kaardistatud on ka vajalike oskuste profiil, mis on tarvilikud selle rolli edukaks täitmiseks.

Uurimus viiakse läbi ettevõtte Ignite OÜ aastal 2014 - 2015 (esimene kvartal) jooksvate projektide ning meeskondade näitel.

Ühtlasi on testimisprotsessi struktureerimine aktuaalne probleem, kuna puuduvad eestikeelsed uuringud, materjalid ja juhtumianalüüsid, mis kirjeldaksid testimise protsessi kohanemist Scrum metoodikat jälgivas arendusmeeskonnas. Seega käesolevast tööst saavad kasu keskmise ja väikese suurusega ettevõtted, kes soovivad kasutusele võtta Scrum arendusmetoodikat ning soovivad saada ülevaadet kvaliteedispetsialisti rollist ja protsessidest.

Käesoleva lõputöö eesmärgiks on struktureerida Ignite OÜ testimisprotsess ning luua selle alusel ettevõtte testimisprotsessi ja kvaliteedispetsialisti rolli kirjeldused. Selle kaudu muudetakse kvaliteedispetsialisti tegevused läbipaistvamaks nii kliendile kui meeskonnale.

## 1.2 Ülesande püstitus

Lõputöö eesmärkideks on:

1. Leida meetodika, kuidas struktureerida testimisprotsessi Scrum arendusmetoodikat kasutavas ettevõttes;
2. Identifitseerida ettevõtte testimisprotsessi probleemkohad ja nende prioriteedid;
3. Koostada leitud probleemidele lahendusettepanekud;
4. Koostada Scrum meeskonna testimisprotsessi kirjeldus ja juhised töötajatele täitmiseks;

Praktiline osa seisneb ettevõtte olemasoleva testimisprotsessi hindamises ja analüüsis, protsessi parenduseks soovitusettepanekute koostamises ning nende rakendamises reaalse projekti näitel. Lisaks, luuakse magistritöö käigus testimisprotsessi ja kvaliteedispetsialisti rolli kirjeldused, mida saab kasutada olemasolevate ja uute kvaliteedispetsialistide koolitamiseks.

## 1.3 Ülevaade tööst

Lõputöö teine peatükk kirjeldab vaadeldavat ettevõtet ja seal kasutatavat metoodikat – Scrum. Metoodikat on kirjeldatud põhinedes agiilmetoodikate alasele kirjandusele ning ettevõtte kogemusele. Lisaks on analüüsitud kvaliteedispetsialisti rolli vajadust Scrum tarkvaraarendusmetoodikas ja antud ettevõttes.

Kolmandas peatükis on kirjeldatud testimise parendamise metoodika. Testimisalase kirjanduse ja varasemate uurimustööde alusel on koostatud analüüs sobiva mudeli valikuks, et toetada valitud metoodikat. Eelneva põhjal luuakse raamistik, mille alusel viiakse läbi testimisprotsessi parendamine ja struktureerimine.

Neljandas peatükis on kirjeldatud ettevõtte testimisprotsessi puudutavad probleemkohad ning seatud nende parendamiseks eesmärgid.

Viiendas peatükis on teostatud ettevõtte olemasoleva protsessi analüüs. Olemasoleva protsessi analüüsi koostamisel on kasutatud kvalitatiivset uurimust (kvaliteedispetsialistide küsitlust)

ning kastutatud saadud tulemit valitud mudeli sisendina. Eesmärkide püstitamisel on lähtutud ettevõtte poolt kirjeldatud probleemkohtadest ja valitud mudelist tulenevatest juhistest.

Kuuendas peatükis on kirjeldatud ja analüüsitud agiilse testimise parimaid praktikaid ning nende alusel koostatud parendussoovitused. Parimaid praktikaid on kirjeldatud nende loojate ja testimisvaldkonna ekspertide loodud kirjanduse ja/või nende internetilehekülgedel leiduvate materjalide põhjal. Samuti on parimate praktikate analüüsiks vaadeldud erinevaid olemasolevaid testimise metoodikaid – V-mudel [21], *ISTQB* (*International Software Testing Qualifications Board*) [18]. Analüüsi tulemustest tehakse järeldused, milliseid praktikaid tuleks kasutada, et täita Ignite OÜ vajadused.

Seitsmendas peatükis on kirjeldatud parimate praktikate rakendamine ja parendatud protsessi tulemusteni jõudmine. Antud peatükis on ka hinnatud parendamise protsessi tulemust ning kirjeldatud muutmise ettepanekud tulevikuks.

Kaheksandas peatükis on kirjeldatud Ignite OÜ struktureeritud testimisprotsess ja kvaliteedispetsialisti rolli oskuste profiil ja vastutused.

Üheksandas peatükis on antud autoripoolne hinnang antud töös välja pakutud testimise struktureerimise metoodikale ja selle rakendamisele.

# 2. Ettevõtte ja Scrum metoodika ülevaade

Järgnevas peatükis on antud ülevaade ettevõttest, mille testimisprotsessi parendatakse. Samuti on kirjeldatud agiilmetoodikate ja Scrum metoodika omavaheline seos ning toodud lühike ülevaade Scrum metoodikast.

## 2.1 Ettevõtte kirjeldus

Ignite OÜ on 2010. aastal asutatud ja Eesti kapitalil loodud tarkvara arendusteenust pakkuv ettevõte. Ettevõttes on kasutusele võetud Scrum metoodika ning arendust viivad läbi meeskonnad. Iga meeskond koosneb 2 või enamast arendajast ning ühest kvaliteedispetsialistist.

Ettevõte on teinud koostööd partneritega paljudest erinevatest valdkondadest:

- Finants;
- Haridus;
- Tervishoid;
- Kasiinomängud;
- Telekommunikatsioon;

Ignite arendusmeeskonnad on loonud projekte nii algusest kui teinud edasiarendusi keerukatele pärandsüsteemidele. Enamasti on projektide läbiviimisel kasutatud Scrumi, mis on eelistatud väiksemate ning keskmise suurusega projektide korral, kus on tähtis, et toode valmiks kiiresti. Scrum keskendub põhiliselt projektijuhtimisele, kus on raske pikalt ette planeerida. [10] Seetõttu on antud metoodika ettevõttele väga hästi sobiv ning on ennast ka aja jooksul tõestanud.

## 2.2 Agiilsete tarkvaraarendusmetoodikate ülevaade

Käesoleva lõputöö konteksti paremaks selgitamiseks annab autor siinkohal ülevaate agiilsete arendusmetoodikate omavahelisest seosest ning kokkusobivusest, kuna antud töö mahus on kasutatud materjale, mis koonduvad üldise „agiilse testimise“ nimetuse alla.

Agiilne manifest [17] kirjeldab järgmist:

„Tarkvara luues ning teisi tarkvara loomise juures aidates oleme leidnud selleks tööks paremaid viise.

Oleme hakanud hindama:

- **inimesi ja nendevahelist suhtlust** rohkem, kui protsesse ja arendusvahendeid
- **töötavat tarkvara** rohkem, kui kõikehõlmavat dokumentatsiooni
- **koostööd kliendiga** rohkem, kui läbirääkimisi lepingute üle
- **reageerimist muutunud** oludele rohkem, kui algse plaani järgimist

Ka parempoolsetel teguritel on väärtus, kuid me hindame vasakpoolseid tegureid kõrgemalt.“

Agiilsed metoodikad on loodud arendajate poolt ning seega on laialdaselt esitatud nende perspektiiv. Agiilne manifest loodi 2001. aastal pealkirjaga „*Manifesto for Agile Software Development*“. Agiilsete metoodikate kirjeldused ei ole kunagi põhjalikult kirjeldanud, kuidas testimine integreerub uude arendusmetoodikasse. Seetõttu on ka tekkinud eespool kirjeldatud tõrked testimisprotsesside integreerimisel agiilsesse maailma. [2]

Agiilse filosoofia aluseks on idee, et ärifunktsionaalsust tarnitakse võimalikult väikestes osades ja nii varakult kui võimalik. Mõistagi, peab kõiki neid osi testima, et tagada nende vastavus ärivajadustele. Selle alusel võib väita, et testimine on eluliselt tähtis osa ka agiilsetes projektides. Traditsionaalsetes projektides on tavaliselt skoop fikseeritud ning aeg ja kvaliteet võivad muutuda. Testimine kaasatakse arendusse projekti viimastes etappides. Agiilse arenduse puhul on fikseeritud aeg ja kvaliteet, kuid skoop muutub ning testimine on kaasatud kogu projekti vältel. Selletõttu on tarvilik, et meeskonnad teeksid väga head koostööd ning selle liikmetel oleks laialdane oskuste pagas.

Lisaks on agiilne arendus kiirelt kohanev muutustega, mis aga püstitab uued väljakutsed testimisele, kuna loob vajaduse automatiseerimiseks, efektiivseks dokumenteerimiseks ja otsesuhtluseks. Need erisused on aga piisavalt suured, et nõuavad kvaliteedispetsialistidelt teistsugust lähenemist ja uusi oskusi, et kohaneda agiilse keskkonnaga [12].

Agiilseid metoodikaid on mitmeid:

1. Scrum;
2. Ekstreemprogrammeerimine (XP);
3. *Crystal*;
4. Erisus-juhitud arendus (*Feature-driven development*, FDD);
5. *Rational Unified Process* (RUP);
6. *Lean Development.*

Käesoleva töö mahus ei vaadelda detailsemalt kõiki välja toodud agiilmetoodikaid vaid piirdutakse ettevõtte poolt valitud ja kasutusel oleva metoodikaga – Scrum.

Agiilmetoodikad on loodud lähtudes agiilsest manifestist, seega on nende põhitõed samad ning nad sobituvad omavahel väga hästi kokku [16]. Seetõttu on võimalik kasutada ära erinevate metoodikate parimaid praktikaid nii ühes kui teises metoodikas [9]. Selle alusel on käesolevas magistritöös kasutatud uurimuse läbiviimiseks erinevaid agiilset testimist käsitlevaid materjale ning need on võetud sobiva testimisprotsessi loomise aluseks.

## 2.3 Scrum metoodika ülevaade

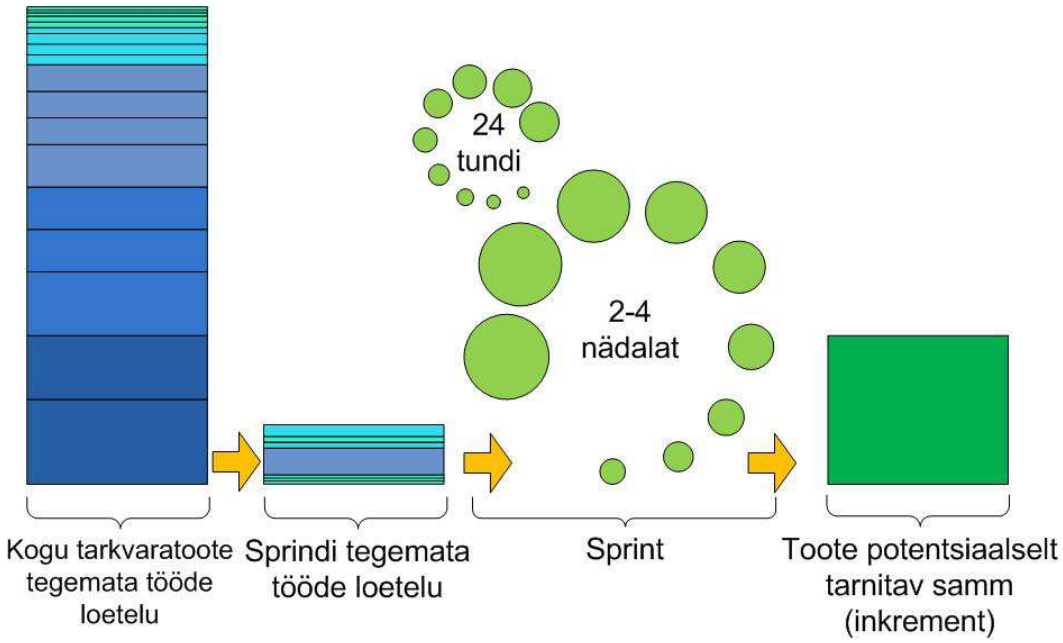
Scrum on empiiriline ja kiirelt kohanev lähenemine projektijuhtimisele. See asendab laiaulatusliku planeerimise kergekaalulise ja lühiajalisega, mis loob protsessi, mis on paindlik ja võimaldab muutustele kiiresti reageerida. Scrum ei dikteeri ette arendusmetoodikat ning tootekvaliteedi tagamiseks kasutatakse meeskonna ühtset teadmusbaasi. Meeskond vastutab ise oma edu eest ning ideaalis toimib ilma konkreetse juhita. Ettemääratud rutiinid tagavad protsessi pideva parendamise projekti tasemel. Agiilne manifest on loonud selles kehtestatud põhitõdedega metoodikale aluse ning Scrum laiendab omalt poolt neid väljatöötatud protsessimudeli ja meeskonna kirjeldusega [4].

Scrumi alusel arendatakse tarkvara inkrementaalselt, kasutades ühte või enamat multi-funktsionaalset meeskonda, mis on iseorganiseeruvad ning koosnevad maksimaalselt kuni seitsmest inimesest. Metoodika kirjeldab struktuuri rollidest, reeglitest ja tehistest. Meeskonnad vastutavad selle eest, et nad looksid ja kasutaksid raamistikus kirjeldatud protsesse. Antud metoodikas kasutatakse arenduseks fikseeritud pikkusega iteratsioone ehk sprinte. Enamasti on need 1-2 nädala pikkused, kuid mitte pikemad kui 30 päeva. Scrum meeskonna eesmärgiks on iga sprindiga luua valmis tükike tootest, mida oleks võimalik toodangusse lasta. Scrum metoodika võimaldab, oma lühikeste iteratsioonide ja paindlike protsesside tõttu, kiiret tagasisidet kliendile ja sellega kaasneb ka varajane võimalike vigade avastamine. [1]

Scrumi põhilised eesmärgid [4]:

1. Tõsta meeskonna efektiivsust;
2. Võimaldada meeskonna arengut läbipaistva protsessi kaudu;
3. Lahendada arendusprotsessis tekkinud takistusi kiirelt;
4. Võimaldada järelvalvet projekti protsessile läbipaistva protsessi kaudu.

Joonisel 1 on kujutatud Scrum arendusprotsess [10].




**Joonis 1. Scrum arendusprotsess**

Põhilised Scrum protsessi juhtimise vahendid on [24]:

1. **Sprint.** Scrum metoodika näeb ette, et projekt jagatakse väiksemateks osadeks ehk sprintideks, millel on kokkulepitud ajaline kestvus. Üldiselt on sprindid kahe kuni nelja nädala pikkused.

2. **Potentsiaalselt tarnitav samm** (*Product Increment*). Iga sprint peab lõppema potentsiaalselt tarnitava tooteosaga, mis tagab, et pidevalt lisatakse tootele vajalikku funktsionaalsust, mille tööd on lõpetatud ja mis loob kliendi jaoks väärtust.

3. **Toote tegemata tööde loetelu** (*Product Backlog*). Tooteomanik haldab ja loob prioritiseeritud nimekirja arendust vajavatest ärivajadustest. Toote tegemata tööde loetelu kirjeldab kõrgel tasemel kogu arendusmahtu. Üldiselt kirjeldatakse ärivajadusi kasutajalugudena.

4. **Sprindi tegemata tööde loetelu** (*Sprint Backlog*). Iga sprindi alguses valib meeskond toote tegemata tööde nimekirjast kõige kõrgema prioriteediga kasutajalood, mida hakatakse arendama. Valitud logi jooksva sprindi jooksul üldiselt ei muudeta.

5. **Valmimise kriteerium** (*Definition of Done*). Potentsiaalselt tarnitava tooteosa valmimise saavutamiseks loob meeskond ühtse nimekirja valmimise kriteeriumitest. Nimekiri kirjeldab tingimusi, mille täitmisel loetakse arendatav funktsionaalsus potentsiaalselt tarnitavaks tooteosaks.

6. **Ajaakende kasutamine** (*Timeboxing*). Ainult need ärivajadused ja funktsionaalsused, mis meeskond on valinud arendamiseks, kuuluvad sprindi tegemata tööde loetellu. Juhul kui valitud töid sprindi jooksul ei lõpetata, siis liigutatakse need tagasi toote tegemata tööde loetellu. Range ajaakendest kinnipidamine on kirjeldatud ka kõikide Scrum protsessis kirjeldatud koosolekute puhul.

7. **Läbipaistvus** (*Transparency*). Kogu protsessi läbipaistvust tagatakse erinevate väikeste koosolekute kaudu. Näiteks peab meeskond igapäevaselt püstjala koosolekuid (*Daily standup*), mille jooksul antakse ülevaade eelnenud päeval tehtud tööst, tekkinud probleemidest ja planeeritud töödest. Selliste koosolekute kaudu tagatakse, et meeskonna liikmed on pidevalt teadlikud teiste liikmete tegemistest ja muudab protsessi läbipaistvamaks. Lisaks hallatakse sprinti võetud tööde progressi kõigile nähtaval kujul, kas kasutades suuri valgeid tahvleid või virtuaalseid tahvleid, mida

kutsutakse Scrum tahvliteks ehk *Scrum board*. Iga sprindi lõpus viiakse läbi retrospektiivi koosolek, mille käigus hindavad meeskonnaliikmed sprinti oma emotsioonide kohaselt ning selle alusel korrigeeritakse meeskonna tööd.

Scrum metoodikas on kirjeldatud järgmised rollid [14]:

1. **Tooteomanik** (*Product Owner*). Selle rolli eesmärk on esindada klienti ja äripoolt. Tema kohustuseks on luua, hallata ja prioritiseerida toote tegemata tööde nimekirja.

2. **Protsessijuht** (*Scrum Master*). Selle rolli põhiülesandeks on vastutada Scrum praktikate kasutamise ja protsessi jälgimise eest. Lisaks peab ta lahendama kõikvõimalikke puudujääke ja probleeme, mis takistavad meeskonnal Scrumi jälgida [13].

3. **Arendusmeeskond** (*Scrum Development Team*). Arendusmeeskonna ülesandeks on arendada ja testida loodavat toodet. Meeskond on multifunktsionaalne ja ise-organiseeruv ning toimib ilma otsese juhita. Otsuseid langetatakse ja võetakse vastu kogu meeskonnaga.

## 2.4 Kvaliteedispetsialisti rolli vajaduse analüüs

Scrum protsessi kirjeldus näeb ette, et arendusmeeskonnas on kõikide meeskonnaliikmete peale kokku esindatud lai oskustepagas. Samas multifunktsionaalsed oskused ei taga kindlust, et üks ja sama inimene suudab erinevate spetsialistide ülesandeid täita samaväärse oskuse ja entusiasmiga. Tarkvara testimine vajab spetsialisti oskuseid ja testimiskogemust, seega võib väita, et igasse meeskonda peaks kuuluma vähemalt üks kvalifitseeritud kvaliteedispetsialist (näiteks *ISQTB* sertifitseeritud). Kvaliteedispetsialist vastutab korrektse disainiga testide loomise eest ja testimise rakendamise eest igas sprindis. Antud roll on vajalik ka näiteks tooteomaniku nõustamiseks, näiteks vastuvõtukriteeriumite loomisel. [4]

Scrum tarkvara arendusmetoodika on üles ehitatud kiire tagasiside andmise ja saamise protsessile. Juhul kui meeskond suudab luua protsessi, kus arendajatele on tagatud kiire ja efektiivne tagasiside, ei ole eraldi kvaliteedispetsialist rolli vaja, kuid selline seis on võimalik vaid lihtsate ja lühikeste projektide puhul, millel on väike risk ning eluiga on lühike. [5] Nagu eespool mainitud, vajab tarkvara testimine spetsiifilisi oskuseid ja kogemust. Arendajad viivad ka läbi testimist, kuid nende esmane fookus on saavutada loodud koodi toimivus ning

lisaks on inimpsühholoogia tõttu keeruline leida oma enda töös tehtud vigu. Paraku kõik inimesed teevad vigu ning mida varem need vead avastatakse, seda odavam on vigade parandamine. Lisaks töötavad arendajad tihtipeale ajalise surve all, seega on keeruline motiveerida arendusspetsialisti viima läbi lisaks kvaliteedispetsialisti ülesandeid ning eeldada, et spetsialist teeb teist tööd sama motiveeritult. Seetõttu on vajalik, et protsessi oleks kaasatud kvaliteedispetsialist, kelle põhitööks on erinevate testimistegevuste läbiviimine, vigade leidmise ja ärahoidmise eesmärgil.

Ettevõtte senine kogemus näitab, et eriti olulist rolli mängivad kvaliteedispetsialistid projektides, milles tegeletakse pärandsüsteemidega (*legacy systems*), kuna sellises seisus tuleb tegeleda keerukamate testimistegevuste, nende planeerimise ja muude sõltuvuste lahendamisega. Lisaks täidavad ettevõtte kvaliteedispetsialistid sageli ka protsessijuhi (*Scrum Master*) rolli ning toetavad meeskonda Scrum protsesside juurutamisel. Seega võttes arvesse neid fakte, on siiski soovitatav igasse arendusprojekti kaasata kvalifitseeritud ja kogenud kvaliteedispetsialist.

# 3. Testimisprotsessi struktureerimise metoodika

Antud peatükis kirjeldatakse metoodika, kuidas saavutatakse lõputööle seatud eesmärkide täitmine.

Testimisprotsessi struktureerimise metoodika aluseks on autor võtnud tervikliku kvaliteedijuhtimise põhimõtted ja selle teoreetilise käsitluse [8]. Lähtudes pideva parendamise mudelist ehk Demingi ringist [3], mis jagab kvaliteedijuhtimise protsessi neljaks üksteisega seotud etapiks, on käesolevas magistritöös läbi viidud kõik neli osa:

- Planeerimine (*Plan*) – Plaani koostamiseks on tarvilik teada, mis seisus ettevõtte testimisprotsess hetkel on. Protsessi hindamiseks valitud sobiv mudel ning kaardistatud ettevõtte probleemid, mida soovitakse lahendada.

- Teostamine (*Do*) – Planeerimise käigus valitud mudeli alusel viiakse läbi uurimus olemasoleva protsessi hindamiseks. Püstitatud eesmärkide ja protsessi analüüsi alusel viiakse läbi uurimus parimate agiilsete testimispraktikate analüüsiks ning nende alusel koostatakse soovitused protsessi parendamiseks.

- Kontrollimine (*Check*) - Koostatud praktikate analüüsi ja selle alusel tehtud soovitusi piloteeritakse ühe meeskonna tööd parendades (4 sprindi jooksul) ning seejärel hinnatakse saavutatud tulemit.

- Korrigeerimine (*Analyze/Act*) – Kontrollimise etapis läbitud hindamise tulemusena selgitatakse välja, kuidas on õnnestunud eesmärkide saavutamine, millised on ilmnenud takistused ja kuidas olemasolevaid plaane tuleb korrigeerida, et saavutada soovitud tulem.

## 3.1 Mudeli valik

Järgnevalt on esitatud planeerimise etapi esimene osa ehk analüüs sobiva mudeli valikuks, mis oleks kooskõlas eespool kirjeldatud testimise struktureerimise metoodikaga ning toetaks Scrum tarkvaraarendusmetoodikat kasutava ettevõtte testimisprotsessi parendamist.

Demingi ringi eeskujul on loodud mitmeid mudeleid ning lähenemisi testimisprotsessi parendamiseks. Mudelipõhised protsesside parendamise metoodikad on loodud lähtudes parimatest praktikatest ning võimaldavad nende alusel hinnata testimisprotsessi hetkeseisu ja kirjeldavad, kuidas protsessi samm-sammult parendada. Mudelid on infotehnoloogias laialdaselt kasutusel ning on kooskõlas rahvusvaheliste standardite ja raamistikega (CMMI, ITIL, TMMI jm), seetõttu on nad ennast ka oma valdkonnas tõestanud. Muud parendamise lähenemised kohandavad tavaliselt olemasolevat protsessi vastavalt konverentsidel ja koolitustel õpitule ning seetõttu ei ole need piisavalt tõestatud, et võtta kasutusele laialdasemalt. Võttes arvesse eespool kirjeldatud fakte valiti testimisprotsessi analüüsimiseks mudelipõhine lähenemine.

Sobiva mudelipõhise lähenemise leidmiseks oli vajalik uurida olemasolevaid testimisprotsessi parendamise mudeleid. Käesolev töö lähtus olemasolevast uurimusest, mille koostas Pavan Kumar, kes analüüsis olemasolevaid testimisprotsesside parendamise mudeleid ning koostas selle alusel ülevaatliku võrdluse „*Test Process Improvement – Evaluation of Available Models*“ [6]. Antud uurimuses ta analüüsis ja kirjeldas erinevaid mudeleid ning tõi välja nende tugevusi ja nõrkusi. Uurimuses vaadeldi mudeleid, mis olid loodud alates aastast 1990. Mudeleid hinnati nelja kriteeriumi alusel, mis mõõtsid nende asjakohasust, täiuslikkust ja sobivust. Kriteeriumite püstitamisel oli lähtutud Frits Philips Instituudi poolt koostatud testimismudelite võrdluse uurimusest. Mudelite võrdlemiseks püsitati neli küsimust [14].

1. Kas mudeli eesmärk on parendada kogu testimisprotsess?
2. Kas mudelil on olemas küpsusstruktuur?
3. Kas mudeli ja selle kasutamise kohta leidub avalikku informatsiooni?
4. Kas mudel on ajakohane ja seda uuendatakse pidevalt?

Püstitatud kriteeriumite alusel leiti, et enamik olemasolevaid mudeleid on kas aegunud või on nende edasiarendamine lõpetatud. Kriteeriumitele vastasid aga täielikult kaks mudelit: TMMi (*Testing Maturity Model Integrated*) ja TPI *Next* (*Test Process Improvement Next*), mis on internetis leiduvate materjalide põhjal ka ühed enamkasutatavad mudelid. Sama uurimuse käigus hinnati ka nende mudelite rakendatavuse lihtsust ning leiti, et TPI *Next* võimaldab kiiremini ja lihtsamalt kaardistada ettevõtte testimisprotsessi probleeme.

Lisaks toetab antud väidet ka rahvusvaheliselt tunnustatud kvaliteedikonsultandi Erik van Veenendaali artikkel „*Test Process Improvement and Agile: Friends or Foes?*“ [7], kus ta analüüsib testimise parendamismudeleid ning toob välja samade mudelite kasutuse. Artiklis on ka kinnitatud, et nii TMMI kui ka TPI *Next* on kasutatavad agiilsete arendusmetoodikate puhul, mis on oluline aspekt antud lõputöö raames.

Erik van Veenendaal on samas artiklis välja toonud, et ametlikult toetab agiilset arendusprotsessi vaid TPI *Next* ning TMMI on küll kasutatav, kuid vajab põhjalikumat kohandamist vastavalt ettevõttele ning metoodika rakendamise juhend agiilse arendusprotsessi puhul on alles väljatöötamisel. Seega lähtudes eespool kirjeldatud tulemitest valis autor testimisprotsessi parendamiseks TPI *Next* mudeli.

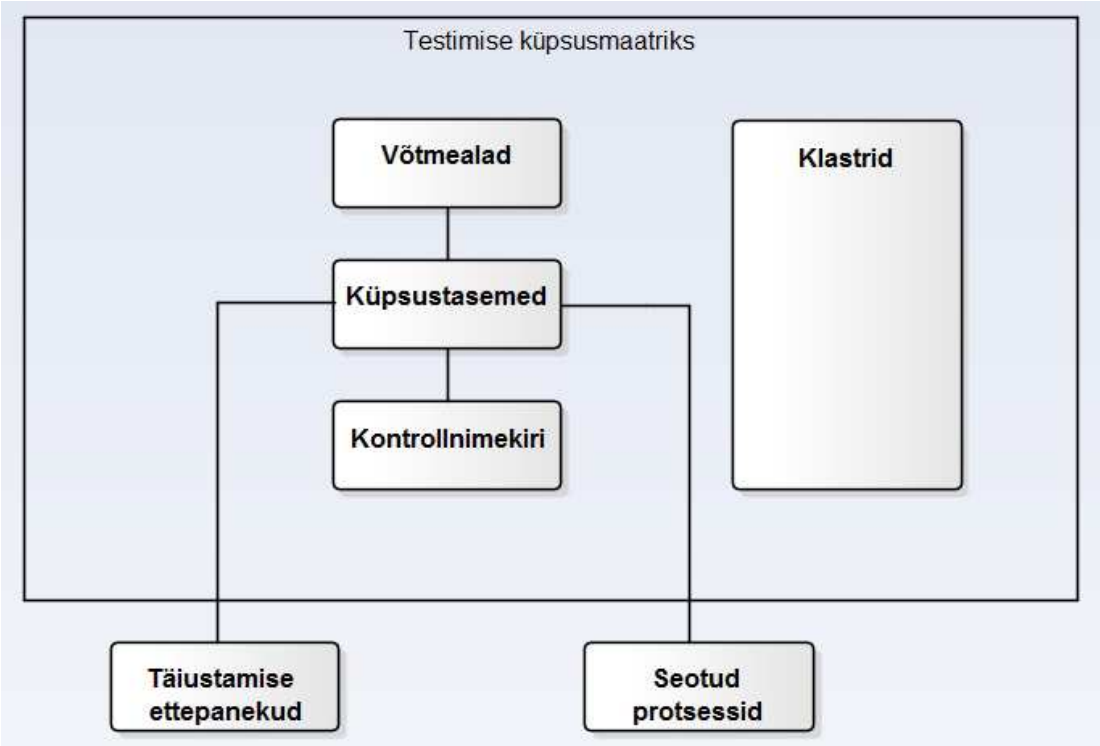
## 3.2 TPI Next mudeli kirjeldus

TPI *Next* on edasiarendus 1997. aastal Tim Koomeni ja Martin Poli poolt loodud TPI (*Test Process Improvement*) mudelile. Edukaks osutunud mudelit muudeti ja täiendati aastate jooksul kogutud kogemuse alusel nii, et seda oleks võimalik kasutada paljudes erinevat tüüpi projektides (sh agiilsed projektid) ning fookus muudeti ärivajadustest lähtuvaks (*Business driven*). Mudeli põhimõtted on kirjeldatud järgnevalt.

TPI *Next* mudel koosneb seitsmest osast:

- Testimise küpsusmaatriks (*Test Maturity matrix*);
- Võtmealad (*Key areas*);
- Küpsustasemed (*Maturity levels*) – Esialgne, Kontrollitud, Efektiivne, Optimeeritud;
- Kontrollnimekiri (*Checkpoints*);
- Klastrid (*Clusters*)
- Seotud protsessid (*Enablers*);
- Täiustamise ettepanekud (*Improvement suggestions*).

Joonisel 2 on toodud kõik nimetatud osad ja nende seosed.




**Joonis 2. TPI *Next* mudel**

Mudel jagab testimisprotsessi 16 võtmealaks, mis omakorda jagunevad kolme gruppi – Kliendisuhtlus (KS), Testijuhtimine (TJ) ja Kvaliteedispetsialisti oskused (KO). Igat võtmeala hinnatakse eraldiseisvalt vastavalt kontrollnimekirjale ning väärtustatakse küpsustasemega. Küpsustasemeid on 4:

- Baas (*Ad-hoc* tegevused);
- Kontrollitud (struktureeritud ja organiseeritud tegevused);
- Efektiivne (varajane ja efektiivne testimine);
- Optimeeritud (kohanev igas situatsioonis).

Igat küpsustaset on võimalik saavutada vaid siis, kui sellele eelnev tase on saavutatud. Kõikide võtmealade väärtuste alusel koostatakse küpsusmaatriks, mis kirjeldab tervet testimisprotsessi. Vaikimisi omab antud metoodika alusel iga võtmeala baastaset. Võtmealale väärtustatakse kõrgem (kui baastaseme) küpsustase vaid siis, kui see vastab kõikidele selle taseme kontrollnimekirja punktidele. Sama reegli alusel hinnatakse ka üldist testimisprotsessi küpsustaset. Tabelis 1 on toodud TPI *Next* mudeli küpsusmaatriks.

**Tabel 1. TPI *Next* küpsusmaatriks**

| | | Võtmeala | Baas | Kontrollitud | | | | Efektiivne | | | | Optimeeritud | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KS | Kliendi pühendumine | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | 3 |
| 2 | KS | Testimise kaasatus protsessis | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | |
| 3 | KS | Testimise strateegia | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | |
| 4 | KS | Testimise organisatsioon | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 5 | KS | Suhtlemine | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | |
| 6 | KS | Raporteerimine | + | 1 | 2 | 3 | | 1 | 2 | 3 | | 1 | 2 | |
| 7 | TJ | Testimise protsessi haldus | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | |
| 8 | TJ | Hindamine ja planeerimine | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 9 | TJ | Meetrika | + | 1 | 2 | 3 | | 1 | 2 | 3 | 4 | 1 | 2 | |
| 10 | TJ | Vigade haldus | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 11 | TJ | Testimisvahendite haldus | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | | 1 | 2 | |
| 12 | KO | Metoodika | + | 1 | 2 | 3 | | 1 | 2 | 3 | 4 | 1 | 2 | |
| 13 | KO | Kvaliteedispetsialisti oskused | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 14 | KO | Testidisain | + | 1 | 2 | 3 | | 1 | 2 | 3 | 4 | 1 | 2 | |
| 15 | KO | Testimisvahendid | + | 1 | 2 | 3 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 16 | KO | Testkeskkond | + | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |

Järgnevalt on kirjeldatud osa võtmealadest, täielik kirjeldus on toodud Lisas 1.

**Kliendi pühendumine** on väga oluline osa efektiivsest kommunikatsioonist ja koostööst kliendi ja meeskonna vahel, mis on omakorda eelduseks eduka testimise läbiviimiseks. Testimine sõltub otseselt kliendi osapoolest, kuna tema loob nii ajalise ja rahalise võimaluse testimiseks. Agiilses arendusprotsessis kliendi kaasatus kriitilise kaaluga, kuna kogu protsess on üles ehitatud otsesuhtlusele kahe osapoole vahel ning klient juhib arendust.

**Testimise kaasatus protsessis** on agiilse metoodika puhul väga tähtis osa, kuna metoodika üks olulistest komponentidest on kiire tagasiside saamine ning testimine on üks põhivahenditest selle võimaldamiseks.

**Testimise strateegia** on arendusprotsessis iga testimistegevuse vundament, mis loob ülevaate mida ja kuidas testitakse. Strateegia kirjeldab testimise eesmärgid, riskid ja lähenemise meetodid.

**Testimisvahendite haldus**. Agiilses keskkonnas on muutustele reageerimine olulisem, kui plaani järgimine. Seega peab meeskond olema valmis mitte ainult nõuete muutumiseks vaid ka näiteks meeskonna liikmete muutumiseks. Testimisvahendite (näiteks testraamistike ja -skriptide) korrektne loomine ja haldamine tõstab nende hooldatavust, taaskasutatavust ning on seetõttu äärmiselt oluline kogu meeskonna jaoks.

**Kvaliteedispetsialisti oskused** loovad aluse efektiivseks testimiseks ja edukaks koostööks meeskonnaga. Iga meeskonnaliige on vastutav oma töö ja meeskonna edukuse eest. Seega on äärmiselt oluline, et kõik liikmed suudaksid omavahel koostööd teha ning vajadusel täita või täiendada teineteise ülesandeid.

Optimeeritud taseme saavutamine on väga hea saavutus, kuid ei peaks olema eesmärk omaette. Kõige efektiivsem on saavutada arendus- ja testimisprotsessi vaheline tasakaal ning parendada neid jooksvalt ja pidevalt. Seetõttu on mudelis ka eraldi välja toodud „Seotud protsessid“, mis loovad mudelis otsese seose testimise- ja arendusprotsessi vahel, kuna osad arenduspraktikad toetavad juba eoses efektiivset testimist.

### 3.2.1 Klastrid

Testimisprotsessi astmelise arengu lihtsustamiseks on mudelisse lisatud klastrid. Klastrid on mudeli loojate poolt koostatud ning loogika on kõikide küpsusmaatriksite puhul samasugune. Üks klaster koosneb hulgast kontrollnimekirja punktidest, mis on kombineeritud erinevatest võtmealadest ning moodustavad ühe tervikliku parendusastme. Klastreid tähistatakse tähtedega („A“, „B“ kuni täheni „M“). Kõik ühe ja sama tähega märgistatud kontrollpunktid kuuluvad ühte klastrisse. Klastrite kasutamine võimaldab jagada parendamise sammud väiksemateks osadeks ning korraga tegeleda üksikute puudujääkide parendamisega ning teha seda mudeli loojate poolt soovitatavas järjekorras. Tabel 2 kirjeldab TPI *Next* küpsusmaatriksit koos metoodikas määratud klastritega.

**Tabel 2. Küpsusmaatriksi klastrid**

| | | Võtmeala | Baas | Kontrollitud | | | | Efektiivne | | | | Optimeeritud | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KP | Kliendi pühendumine | + | A | B | B | C | F | H | H | | K | M | M |
| 2 | KP | Testimise kaasatus protsessis | + | A | B | C | E | H | H | J | | L | L | |
| 3 | KP | Testimise strateegia | + | A | A | B | E | F | F | H | | K | L | |
| 4 | KP | Testimise organisatsioon | + | A | D | D | E | I | I | J | J | K | L | L |
| 5 | KP | Suhtlemine | + | B | C | C | D | F | F | J | | M | M | |
| 6 | KP | Raporteerimine | + | A | C | D | | F | G | G | | K | K | |
| 7 | TH | Testimise protsessi haldus | + | A | A | B | B | G | H | J | | K | M | |
| 8 | TH | Hindamine ja planeerimine | + | B | B | C | C | G | H | I | I | K | L | L |
| 9 | TH | Meetrika | + | C | C | D | | G | H | H | I | K | K | |
| 10 | TH | Vigade haldus | + | A | A | B | D | F | F | H | J | K | L | L |
| 11 | TH | Testimisvahendite haldus | + | B | B | D | E | I | I | J | | L | L | L |
| 12 | KO | Metoodika | + | C | D | E | | F | H | J | J | M | M | |
| 13 | KO | Kvaliteedispetsialisti oskused | + | D | D | E | E | G | G | I | I | K | K | M |
| 14 | KO | Testidisain | + | A | A | E | | F | I | I | J | K | K | M |
| 15 | KO | Testimisvahendid | + | E | E | E | | F | G | G | I | L | M | M |
| 16 | KO | Testkeskkond | + | C | D | D | E | G | H | J | J | L | M | M |

# 4. Eesmärkide kaardistamine

Testimisprotsessi parendamiseks vastavalt ettevõtte vajadustele kaardistati olulisemad testimist puudutavad probleemid. Antud tegevus on struktureerimise metoodika esimese sammu (planeerimise) osa. Ettevõtte probleemide kaardistamisel lähtuti meeskondade ja juhtkonna senisest projektide läbiviimisele antud tagasisidest.

Kaardistamisel selgusid järgnevad probleemkohad:

**Kliendi rahulolu ja protsessi läbipaistvus** – vajadus anda kliendile selge ülevaade, milliseid ülesandeid katab kvaliteedispetsialist ning millist lisaväärtust see roll lisab arendusmeeskonnale;

***Ad-hoc* testimisprotsess** – ebakindlus kvaliteedispetsialistide seas, kuna puuduvad tööjuhised ja on ebaselge, millised on kvaliteedispetsialisti tööle püstitatud eesmärgid ja nõudmised. Seetõttu väheneb testimise efektiivsus ning lühikeste sprintide ja ajasurve tõttu puuduvad sageli:

- Kokkulepped projekti osapooltega (tarnimiseks või tööde ümbertegemiseks);
- Testiplaanid ja hindamismeetodid – puudub ülevaade, mida testiti ja mis aja jooksul;
- Riskide hinnang;
- Testistrateegia;

**Meeskonna rahulolu** – puudub struktureeritud protsess testimiseks, seega on vähenenud meeskondade kindlustunne tarnitava toote suhtes ning iga projekti puhul alustatakse protsessi väljatöötamist uuesti või lähtutakse varasemast kogemusest. Uute liikmete koolitamine on selletõttu keeruline ja ajakulukas.

Ettevõtte vajadustest lähtuvalt püstitati üldised eesmärgid testimisprotsessi parendamisele 2015 aasta lõpuks:

- Testimise muutmine efektiivsemaks – tõsta testimise küpsustaset, suurendada arenduse käigus leitavate vigade arvu. Mõõdik – 50% võrra väheneb garantiitööde maht ja igas sprindis eelmise sprindi vigade parandamise arv. Igal projektil on testiplaan ja -strateegia. Meeskondade ja liikmete retrospektiivides ei mainita testimise korrapäratust.

- Testimise muutmine läbipaistvamaks – võimaldada pidevalt kliendile ja meeskonnale hea ülevaade arenduses oleva toote kvaliteedist. Mõõdik – 80% klientidest annavad regulaarse tagasiside kaudu positiivse hinnangu meeskonna töö kvaliteedile. Meeskondade ja kvaliteedispetsialistide regulaarsetel tagasiside koosolekutel ei kajastu ebakindlus testimise vajalikkuse osas.

- Testimise muutmine korratavaks protsessiks – luua alus ühisele testimisprotsessile ja koostada sellest terviklik ülevaade kvaliteedispetsialistide koolitamiseks. Mõõdik – kõik meeskonnad (100%) kasutavad kirjeldatud testimisprotsessi või on loonud selle alusel kohandatud versiooni vastavalt oma vajadustele.

Eespool kirjeldatud probleemid viitavad, et ettevõttes on ositi kasutusel *Ad-hoc* testimisprotsess (ehk vastab vähemalt TPI *Next* mudeli baastasemele). Ettevõtte poolt püstitatud eesmärke toetab TPI *Next* mudeli „Kontrollitud“ taseme saavutamine. TPI *Next* mudelis [12] on kirjeldatud, et „Kontrollitud“ küpsustaseme kontrollpunktide täitmine tagab, et protsessis oleks olemas minimaalse vajalikul tasemel erinevad testimistegevused ning selle alusel on võimalik luua esmane testimisprotsess, mis on korratav erinevates projektides. „Kontrollitud“ taseme protsess tähendab, et testimist on võimalik juhtida ning see annab piisavat infot testitava toode kvaliteedist. Seega võib testimisprotsessi parendamise eesmärgiks seada ettevõtte testimisprotsessi „Kontrollitud“ tasemele viimise.

# 5. Olemasoleva protsessi analüüs

Järgnevalt on toodud ettevõtte olemasoleva testimisprotsessi analüüsimise protsessi ülevaade ja tulem. Lisaks on antud peatükis valitud skoop, mis on aluseks testimisprotsessi parendamisel. Kirjeldatud sammud on struktureerimise metoodika planeerimise etapi teine osa.

Käesolevas lõputöös on ettevõtte olemasoleva testimisprotsessi hindamiseks kasutatud kvalitatiivset uurimust, kuna eesmärgiks oli kaardistada protsess, mis on juba olemas. Uurimuse käigus viidi läbi küsitlus nelja arendusmeeskonna kvaliteedispetsialisti hulgas. Küsimused koostati lähtuvalt protsessi parendamiseks valitud mudelist ning praktik Tilo Linz'i kirjeldatud küsimustest raamatus „*Testing in Scrum*“ [3], mis võimaldasid kaardistada erinevaid testimist puudutavaid aspekte ja tuvastada võimalikke puudujääke. Küsimustik on loodud sellisena, et saada ülevaade erinevatest testimisprotsessi osadest. Küsimustik koostati inglise keeles, kuna antud ettevõte on rahvusvahelise töötajaskonnaga. Küsimustik on välja toodud lisas 4. Küsitlus ja materjalide analüüs viidi läbi paralleelselt ning vajadusel täpsustati segased kohad ja puudujäägid suuliselt.

Lisaks küsimustikule analüüsiti olemasolevat dokumentatsiooni ning koguti andmeid töötajatega vabas õhkkonnas suheldes. Analüüsi käigus hinnati iga meeskonna testimisprotsessi terviklikult ning igat võtmeala võrreldi vastu mudelis püstitatud kontrollpunkti.

Selleks, et anda ülevaadet, kuidas toimus ettevõtte testimisprotsessi hindamine on järgnevalt kirjeldatud osa võtmealade analüüsi ja hindamise tulemused. Analüüsi kogu lõpptulem on toodud ettevõtte küpsusmaatriksis tabelis 3. Loetavuse huvides on toodud välja iga võtmeala number mudelis, mis on samaväärne lisas 6 toodud tabelis oleva võtmeala numbriga. Selguse mõttes on analüüsi tulemuste kirjeldamiseks kasutatud viitamist numbri kujul olevatele kontrollpunktidele, nagu on toodud joonisel 1.

**Kliendi pühendumine.** (Võtmeala 1) Ettevõtte teenuseid kasutavad kliendid on kaasatud arendustegevusse läbi Scrum protsessi praktikate. Kõikides meeskondades on tagasiside põhjal hea ja efektiivne kommunikatsioon kliendiga. Igas meeskonnas on kvaliteeditegevuste ja testimise eest vastutav isik. Tooteomanikud annavad planeerimiskoosolekutel sisendit testimisprotsessile (riskid ja prioriteedid) ning kaasavad arendusse ka teisi huvitatud osapooli. Kliendid/tooteomanikud ja teised kaasatud osapooled esitavad küsimusi ja avaldavad arvamust meeskonna poolt läbiviidavatel esitlustel. Pidevalt antakse tooteomanikule ülevaadet testimise seisust (püstjalakoosolekud). Tooteomanik arvestab kvaliteedispetsialisti poolt välja toodud riskidega ning võtab neid arvesse. Retrospektiivide kaudu tuuakse välja puudujääke testimisprotsessis ning leitakse nendele ühine lahendus.

**Tulem:** Antud võtmealal on täidetud kõik kontrollitud, efektiivse ja optimeeritud taseme kontrollpunktid. Antud võtmeala on optimeeritud tasemel.

**Testimise kaasatus protsessis.** (Võtmeala 2) Projekti alguses luuakse meeskonnaga koos ülevaade, kes on huvitatud osapooled, millised on rollid ja vastutused. Uue keskkonnaga kohanemisel ei looda koheselt suurt pilti testimise skoobist ja riskidest, vaid lähtutakse Scrumi praktikatest ning kasutatakse nende kaardistamiseks planeerimise koosolekuid. Kvaliteedispetsialist kaasatakse planeerimise koosolekutel projekti algusest peale. Üldise pildi ja riskide selgitamine tekitab projekti alguses segadust ning puudub süstematiseeritud lähenemine testimisele, selle eesmärgi ja skoobi kokkuleppimiseks. Lisaks selginevad meeskonna kohustused ja kasutatav protsess alles projekti esimeste sprintide jooksul. Esimestel sprintidel teostatakse funktsionaalset testimist tooteomaniku poolt kirjeldatud kasutajalugude alusel ning meenutab osaliselt *ad-hoc* testimist, kuid toimub pidevalt ning on integreeritud protsessi. Kvaliteedispetsialist teostab leitud vigadele analüüsi ning kaardistab nende põhjused ja esitab need tooteomanikule, see tulem on sisendiks võimalike tooteriskide hindamisel. Testimise sisenddokumente ja vahetulemeid hoiab kvaliteedispetsialist enda töökeskkonnas ning enamasti kasutatakse nende haldamiseks tabelarvutuse tarkvara, seega testilugude optimeerimisele ja taaskasutusele olulist rõhku ei pöörata.

**Tulem:** Täidetud on kontrollitud taseme 3 ja 4 punkt ning efektiivse taseme 1 ja 2 punkt. Täielikult täidetud ei ole ükski küpsustase, seega antud võtmeala on baastasemel.

**Testimise strateegia.** (Võtmeala 3) Teststrateegia loob kvaliteedispetsialist projekti alguses lähtuvalt kliendi ettevõtte vajadustest või loob uue strateegia lähtudes üldistest projekti

vajadustest. Strateegia kirjeldamisele ei ole loodud malli ning enamasti ei ole tarvilik kliendi nõusolekut testistrateegia kasutamiseks, seega on ettevõttes kasutusel erinevaid lähenemisi. Meeskonnad kasutavad harjumuspäraseid strateegiaid, kus enamlevinud on musta- ja valge kasti testimine ning madala taseme testide kaetus on määratud vähemalt 50%.

**Tulem:** Täidetud on kontrollitud taseme 3 punkt, seega antud võtmeala vastab baastasemele.

**Testimise organisatsioon.** (Võtmeala 4) Ettevõttes on määratud kvaliteedispetsialistide arengu eest vastutav ametikoht. Lisaks tunnustatakse pideva õppimise protsessi. Kvaliteedispetsialiste koolitatakse (nt *ISTQB* koolitused ja konverentsid) ning arendatakse läbi tiheda koostöö arendajatega. Ettevõttes on kirjeldatud kvaliteedispetsialisti vastutused ja vajalikud oskused. Igasse meeskonda on kaasatud kvaliteedispetsialist. Lisaks toetab testimistegevusi sageli ka kliendipool ja tooteomanikud. Testimisorganisatsioonil puudub struktureeritud pikaajaline arenguplaan.

**Tulem:** Täidetud on kontrollitud taseme 1 ja 3 punkt. Antud võtmeala vastab baastasemele.

**Testimisvahendite haldus.** (Võtmeala 11) Testilood kirjeldatakse meeskonnale kättesaadavates vahendites, sinna hulka kuuluvad nii tabelarvutus- kui ka tekstitöötlustarkvarad. Loodud dokumente jagatakse meeskonnaga läbi interneti. Iga testilugu omab seost tooteomaniku poolt kirjeldatud kasutajalooga. Parema läbipaistvuse tagamiseks kasutatakse ka näiteks maatrikseid kasutaja- ja testilugude seoste kujutamiseks. Kvaliteedispetsialistid teostavad testimist arenduskeskkondades, mille tagamist toetavad arendajad. Lisaks hoitakse tooteomaniku poolt prioritiseeritavat toote tegemata tööde nimekirja (*Product Backlog*) virtuaalselt kogu meeskonnale kättesaadavas keskkonnas. Ühine struktuur testimisvahendite säilitamiseks ja haldamiseks aga puudub. Kvaliteedispetsialistide poolt loodavaid automatiseeritud teste hoiavad osad tiimid versioonihaldussüsteemides.

**Tulem:** Täidetud on kontrollitud taseme punktid 1-4. Võtmeala vastab kontrollitud tasemele.

**Kvaliteedispetsialisti oskused.** (Võtmeala 13) Ettevõttes on kirjeldatud kvaliteedispetsialistide vastutused kirjeldatud, kuid sageli leiavad olulisi vigu just äripoole inimesed. Üldiselt on kvaliteedispetsialistid kogemusega, kuid puudub struktuurne lähenemine agiilsele keskkonnale sobivale testimisele. Ettevõttes pole kehtestatud kohustuslikke testimismeetodeid, seega neid ka ei jälgita. Suurt rõhku pannakse töötajate enesearengule ning igal kvaliteedispetsialistil on oma arengukava, kuid oskuste hindamist

nende alusel läbi ei viida. Iga meeskond töötab omaette üksusena, olulist koostööd ei tehta (nt kogemuste jagamist). Iga kvaliteedispetsialist kujundab testimisprotsessi vastavalt oma oskustele.

**Tulem:** Täidetud on kontrollitud taseme 1 punkt. Antud võtmeala on baastasemel.

Analüüsi käigus selgus, et nelja meeskonna küpsustasemed olid enam-vähem sarnased, kuid küpsusmaatriks koostati kõige nõrgema tulemi järgi. Tabel 3 annab ülevaate antud ettevõtte testimisprotsessi küpsust. Rohelisega taustaga on tähistatud täidetud kontrollpunktid.

**Tabel 3. Ettevõtte testimisprotsessi küpsusmaatriks**

| | | Võtmeala | Baas | Kontrollitud | | | | Efektiivne | | | | Optimeeritud | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KP | Kliendi pühendumine | + | A | B | B | C | F | H | H | | K | M | M |
| 2 | KP | Testimise kaasatus protsessis | + | A | B | C | E | H | H | J | | L | L | |
| 3 | KP | Testimise strateegia | + | A | A | B | E | F | F | H | | K | L | |
| 4 | KP | Testimise organisatsioon | + | A | D | D | E | I | I | J | J | K | L | L |
| 5 | KP | Suhtlemine | + | B | C | C | D | F | F | J | | M | M | |
| 6 | KP | Raporteerimine | + | A | C | D | | F | G | G | | K | K | |
| 7 | TH | Testimise protsessi haldus | + | A | A | B | B | G | H | J | | K | M | |
| 8 | TH | Hindamine ja planeerimine | + | B | B | C | C | G | H | I | I | K | L | L |
| 9 | TH | Meetrika | + | C | C | D | | G | H | H | I | K | K | |
| 10 | TH | Vigade haldus | + | A | A | B | D | F | F | H | J | K | L | L |
| 11 | TH | Testimisvahendite haldus | + | B | B | D | E | I | I | J | | L | L | L |
| 12 | KO | Metoodika | + | C | D | E | | F | H | J | J | M | M | |
| 13 | KO | Kvaliteedispetsialisti oskused | + | D | D | E | E | G | G | I | I | K | K | M |
| 14 | KO | Testidisain | + | A | A | E | | F | I | I | J | K | K | M |
| 15 | KO | Testimisvahendid | + | E | E | E | | F | G | G | I | L | M | M |
| 16 | KO | Testkeskkond | + | C | D | D | E | G | H | J | J | L | M | M |

Tabeli 3 alusel saab anda hinnangu, et ettevõtte testimisprotsessi üldine küpsustase on hetkel baastasemel. Selleks, et katsetada struktureerimise metoodikat ning kasutada loodud küpsusmaatriksit efektiivse mõõdikuna, otsustati protsessi parendamist piloteerida ühe meeskonna näitel, kelle tulemid vastasid täpselt eespool toodud küpsusmaatriksile.

## 5.1 Võtmealade prioritiseerimine

Kuna võtmealasid on palju, siis on tarvilik otsustada, millised neist on kõige olulisemad ning vajavad parendamist esmajärjekorras. Antud probleemi võimaldab valitud mudel mugavalt lahendada, kasutades selleks võtmealade prioritiseerimist. Prioriteetide määramiseks on vaja valida kõige olulisemad võtmealad, selleks kasutame käesoleva lõputöö mahus metoodikas

välja toodud eraldi kaardistust agiilsete põhimõtete ja võtmealade seostest. Lisas 2 on kirjeldatud, kuidas agiilsed põhitõed on kindlate võtmealadega ning selgitatud, miks need on olulised [12].

Lähtudes peatükis 4 kirjeldatud ettevõtte probleemidest analüüsiti agiilse manifesti põhimõtteid ning seoseid võtmealadega ning selle alusel otsustati pöörata tähelepanu järgnevatele agiilsetele põhimõtetele:

- Kõige olulisem on tagada kliendi rahulolu, tarnides talle vajalikku tarkvara võimalikult kiiresti ja tihti;
- Tehnilist täiuslikkust ja head disaini pideva tähelepanu all hoides tagatakse tarkvaraarenduse kiirus ja paindlikkus;
- Agiilse tarkvaraarenduse protsessid soodustavad jätkusuutlikku arendust. See tähendab, et projektiga saab samas tempos jätkata määramata aja jooksul.

Valitud agiilsete põhimõtete ja testimisprotsessi parendamise metoodikas määratud seoste alusel määrati igale küpsusmaatriksi võtmealale olulisuse kategooria – Kõrge (K), Normaalne (N), Madal (M). „Kõrge“ hinnang määrati võtmealadele, mis on otseselt seotud valitud põhimõtetega. „Madal“ hinnang määrati võtmealadele, mis ei ole nii olulised ettevõtte testimisprotsessi jaoks, kuna on hetkel rahuldaval tasemel ning „Normaalne“ hinnang määrati ülejäänud võtmealadele. Tabelis 4 on välja toodud kategooriatesse jaotatud võtmealad.

**Tabel 4. Võtmealade jaotus olulisuse kategooriatesse**

| Kõrge | Normaalne | Madal |
|---|---|---|
| Testimise kaasatus protsessis | Kliendi pühendumine | Testimise organisatsioon |
| Testimise strateegia | Suhtlemine | Raporteerimine |
| Kvaliteedispetsialisti oskused | Hindamine ja planeerimine | Vigade haldus |
| Testidisain | Testimisvahendite haldus | Testkeskkond |
| Testimise protsessi haldus | Metoodika | Meetrika |
| | Testimisvahendid | |

„Kõrge“ hinnangu saanud võtmealade klastrite väärtusi muudeti olulisemaks, ehk „A“-le lähemale. „Normaalse“ hinnanguga võtmealade klastrid jäid samaks ning „Madala“ hinnanguga klastrite väärtuseid muudeti „A“-st kaugemaks, ehk vähem olulisemaks.

Tabelis 5 on toodud uuendatud võtmealade klastrite väärtused vastavalt määratud olulisusele. Rohelisel taustal on toodud juba saavutatud kontrollpunktid, kollase tausta on tähistatud parendamise skoopi valitud võtmalad ja „A“-taseme klastri täitmiseks vajalikud kontrollpunktid.

**Tabel 5. Ettevõtte testimisprotsessi küpsusmaatriks koos eesmärkidega**

| | | Võtmeala | K | N | M | Baas | Kontrollitud | | | | Efektiivne | | | | Optimeeritud | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KP | Kliendi pühendumine | | x | | + | A | B | B | C | F | H | H | | K | M | M | |
| 2 | KP | Testimise kaasatus protsessis | x | | | + | A | A | C | D | G | G | H | | K | K | | |
| 3 | KP | Testimise strateegia | x | | | + | A | A | A | D | E | E | G | | J | K | | |
| 4 | KP | Testimise organisatsioon | | | x | + | B | E | E | I | J | J | K | K | L | M | M | |
| 5 | KP | Suhtlemine | | x | | + | B | C | C | D | F | F | J | | M | M | | |
| 6 | KP | Raporteerimine | | | x | + | C | D | F | | G | K | K | | L | L | | |
| 7 | TH | Testimise protsessi haldus | x | | | + | A | A | A | A | H | J | K | | M | L | | |
| 8 | TH | Hindamine ja planeerimine | | x | | + | B | B | C | C | G | H | I | I | K | L | L | |
| 9 | TH | Meetrika | | | x | + | D | D | E | | F | G | G | H | L | L | | |
| 10 | TH | Vigade haldus | | | x | + | B | C | C | F | H | G | I | K | L | M | M | |
| 11 | TH | Testimisvahendite haldus | | x | | + | B | B | D | F | I | I | J | | L | L | L | |
| 12 | KO | Metoodika | | x | | + | C | D | E | | F | H | J | J | M | M | | |
| 13 | KO | Kvaliteedispetsialisti oskused | x | | | + | C | C | D | D | E | E | G | G | I | I | K | |
| 14 | KO | Testidisain | x | | | + | A | A | C | | D | E | E | F | I | I | J | |
| 15 | KO | Testimisvahendid | | x | | + | E | E | E | | F | G | G | I | L | M | M | |
| 16 | KO | Testkeskkond | | | x | + | D | F | F | G | H | H | J | J | L | M | M | |

Protsessi küpsusmaatriksi analüüsist selgub, et kontrollitud tasemele jõudmiseks on arenguruumi päris palju. Kontrollitud tase tagaks ettevõttele struktureeritud ja organiseeritud testimisprotsessi. Lähtuvalt eesmärkidest tuleb parendamist alustada „A“-klastri võtmealade kontrollpunktide (tähistatud kollasega) täitmisega. Seega on antud lõputöö raames keskendutud järgmiste võtmealade analüüsimisele ja parendamisele:

1. Testimise kaasatus protsessis;
2. Testimise strateegia;
3. Testimise protsessi haldus;
4. Testidisain.

## 5.2 Parendamise skoop

Järgnevalt on kirjeldatud, milliseid kontrollpunkte on vaja täita, et saavutada „A“ – taseme klaster valituks osutunud võtmealadel.

**1. Testimise kaasatus protsessis**. Pidev testimise kaasamine tarkvara arendusprotsessi jooksul võimaldab hoida fookust toote kvaliteedil juba projekti algusest peale. Samuti võimaldab see vähendada riski, et testimine muutub arendusprotsessi „pudelikaelaks“.

Kontrollitud taseme saavutamiseks on antud võtmealal vajalik täita kaks „A“ - taseme kontrollpunkti:

1. Testimise eesmärk, skoop, riskid ja metoodika arutatakse projekti põhiliste osapooltega läbi juba varakult projekti alguses;

2. Testimistegevusi viiakse läbi juba enne testide jooksutamise tegevusi.

Metoodika alusel tuleks testimistegevusi alustada koheselt projekti algusetapis või kasutajalugude loomise faasis. Konkreetsete tegevuste kaardistamiseks ja juhendi loomiseks on vaja uurida, kuidas kirjeldatakse agiilses keskkonnas testimisprotsessi ja milliseid tegevusi ning kuidas on võimalik läbi viia enne testide jooksutamist.

**2. Testimise strateegia.** Testimise strateegia võimaldab testimisprotsessi juhtida optimaalselt ehk ta kirjeldab milliseid ressursse ja teste kasutades on võimalik arendatav süsteem vajalikul määral ära testida. Mis omakorda tagab, et kõige riskantsemad vead leitakse võimalikult vara.

Kontrollitud taseme poole pürgimiseks on antud võtmealal vajalik esialgu täita kaks „A“ - taseme kontrollpunkti :

1. Projekti põhilised osapooled nõustuvad testistrateegiaga;

2. Testistrateegia luuakse põhinedes toote riskianalüüsil.

Metoodika soovitab luua testistrateegia lähtudes tooteriskidest, hallata teste ja leitud vigu vastavalt nende riskidele ning luua vajalikud regressiooni testiplaanid uute tarnete jaoks. Ühtlasi on soovitatud kaasata tehnilised osapooled testimise strateegia loomisesse ning tõsta kliendi teadlikkust testimisest. Testidisaini tehnikad peaks olema valitud ja põhjendatud.

Seega on vajalik detailsemalt uurida agiilse testimise strateegia loomist ning testidisaini tehnikaid.

**3. Testimise protsessi haldus.** Protsessi haldamine toetab efektiivset testimist teatud aja ja kuluga. Testimise eesmärgid on piisavalt keerulised, et neid ei ole võimalik saavutada ühe lihtsa tegevusega, vaid tuleb läbida erinevate tegevuste jada, seega tuleb seda juhtida kui protsessi. Proaktiivne protsessi juhtimine võimaldab testimise eesmärkide saavutamist. Seega tegevused peavad olema juba ette läbi mõeldud ja juhitud.

Kontrollitud taseme saavutamiseks on tarvilik täita neli „A“ kontrollpunkti:

1. Projekti alguses luuakse testiplaan (mis kirjeldab vähemalt kvaliteedispetsialisti rolli ja kohustused);
2. Testiplaan lepitakse kokku kliendiga (täpsustatakse, mida kliendile üle antakse);
3. Testimistegevusi jälgitakse ning vajadusel muudetakse;
4. Testiplaan lepitakse kokku teiste oluliste osapooltega.

Kirjeldatud kontrollpunktide täitmiseks on tarvilik uurida ja kirjeldada testiplaan.

**4. Testidisain.** Testilugude disain võimaldab testide jooksutamisel leida võimalikke vigu ning võimaldab struktureeritud lähenemist. Praktika näitab, et kõike ei ole võimalik testida ning seetõttu tuleb teha valikuid lähtuvalt toote riskidest. Testidisain võimaldab tagada vajalikku testide katvust ja põhjalikkust.

Kontrollitud taseme saavutamiseks on vajalik kõigepealt täita kaks „A“-klastri kontrollpunkti:

1. Testilugusid luuakse ja kirjeldatakse;
2. Testilood koosnevad sobivast kokkulepitud kirjeldusest.

Metoodika soovitab luua praktiline testilugude kirjutamise juhend, kuna liiga suur vabadus testide disainimisel võib viia ebaefektiivsete testilugudeni. Näiteks juhul, kui keegi teine peab loodud testilugusid läbima, siis võivad need võõra kasutaja jaoks olla liiga ebatäpsed või infoga üle küllastatud. Antud kontrollpunktide täitmiseks uuritakse ja kirjeldatakse parimaid praktikaid testilugude loomisel.

# 6. Agiilse testimise parimad praktikad

Antud peatükis kirjeldatakse agiilse testimise parimaid praktikaid vastavalt eelmises peatükis püstitatud parendamise skoobile. Kuna parendamismudel kirjeldab protsessi kontrollpunktid, kuid ei määra, kuidas neid saavutada, on vajalik läbi viia eraldi analüüs konkreetsete praktikate kasutusele võtmiseks. Parimate praktikate kirjelduse alusel koostas autor parendamise ettepanekud. Kirjeldatud sammud on struktureerimise metoodika teostamise etapi esimene osa.

## 6.1 Testimise kaasatus protsessis

Võtmeala „Testimise kaasatus protsessis“ vajalike kontrollpunktide täitmiseks on vajalik selgitada välja, millised on parimad testimisprotsessi praktikad ning kuidas on võimalik alustada kvaliteedi tagamise tegevustega enne testide jooksutamise etappi.

*International Software Testing Qualifications Board* (ISTQB) on kirjeldanud, et traditsionaalne testimise protsess koosneb järgnevatest osadest [18]:

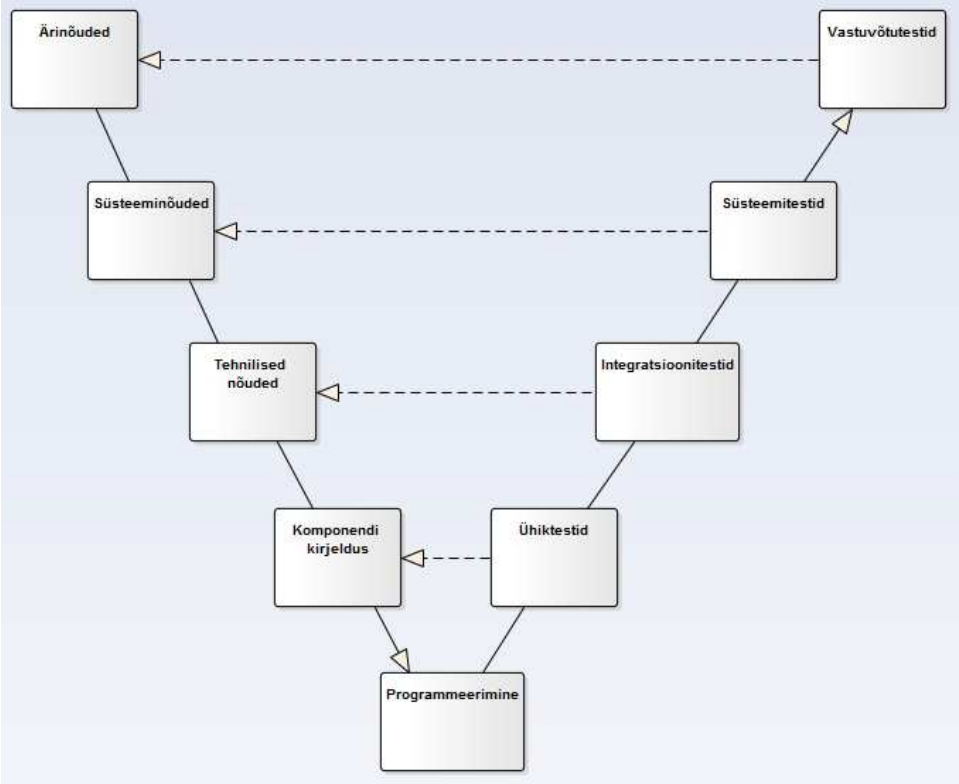
1. Planeerimine ja kontroll;
2. Analüüs ja disain;
3. Testide loomine ja läbiviimine;
4. Lõpetamise kriteeriumi hindamine ja raporteerimine;
5. Testimise lõpetamise tegevused.

Lisaks on täpsustatud, et kuigi tegevused on loogiliselt järjestatud, on tavapärane, et nimetatud tegevused võivad toimuda samaaegselt ning võivad tekkida ülekatted. Samad punktid kehtivad ka agiilse testimise puhul. Protsess oma olemuselt ei ole muutunud, küll on muutunud selle läbiviimine, vahendid ja prioriteedid [3]. Järgnevalt vaatleme lähemalt planeerimist, analüüsi ja disaini osi, kuna need on kõige enam seotud vajalike kontrollpunktide täitmiseks.

### 6.1.1 Testimise tasemed ja protsess

Traditsionaalse tarkvara arendusprotsessi testimise osa ja agiilse protsessi testimise erisuste mõistmiseks vaatleme neid lähemalt. Traditsionaalsed protsesside mudelid jaotavad projekti konkreetsete eesmärkide ja tähtaegadega eraldiseisvateks tükkideks. Samuti on rangelt määratletud projekti meeskonna ülesanded ja kohustused. [3]

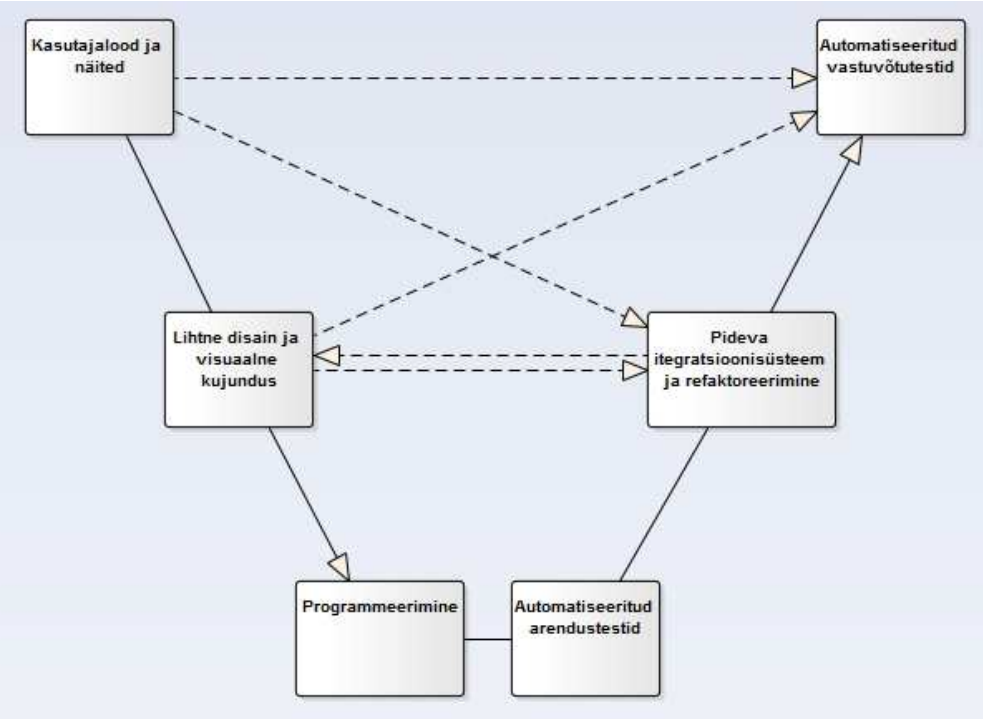
Lisaks tähtaegade kontrollimisele, on defineeritud ka abstraktsioonitasemed, kuidas projekti arendust vaadeldakse. Selle heaks näiteks on Joonisel 3 kujutatud V-mudel [3]. V-mudel on üks enamlevinud verifitseerimise ja valideerimisemudeleid, mis on osaliselt inspireeritud kosk-mudelist. Mudeli alusel koosneb protsess järjestikustest etappidest ning iga eelnev etapp tuleb lõpetada, enne kui järgmist võib alustada. Iga arendusetapi jaoks on planeeritud paralleelne testimisetapp (ühiktestimine, integratsioonitestimine, süsteemitestimine, vastuvõtutestimine).




**Joonis 3. Tarkvara testimise V-mudel**

Scrum läheneb aga arendusprotsessile teisiti, seega kirjeldatud etapilist protsessi kasutada ei saa. Küll aga on tarvilik läbida kõik nimetatud testimise etapid ning kasutada neid pidevalt, igas tsüklis ja paralleelselt. Seega on võimalik kohandada V-mudeli häid külgi ning luua selle alusel uus sobiv mudel [20].

Joonisel 4 kirjeldatud mudel kujutab valideerimis- ja verifitseerimisprotsessi, mis on kohandatud Scrum metoodikale sobivaks.




**Joonis 4. Scrumile kohandatud testimise mudel**

Antud protsessis olulisel kohal järgmised punktid [20]:

- Kasutajalood vastavad *INVEST* kriteeriumile (selgitatud peatükis 6.1.2);
- Pööratakse tähelepanu visuaalse kujunduse loomisele;
- Koodi luuakse lihtsa disainiga;
- Jälgitakse pideva integratsiooni ja refaktoreerimise põhimõtteid;
- Automatiseeritakse nii arendus- kui vastuvõtuteste;

Kohandatud protsess katab ära eelnevalt V-mudelis kirjeldatud testimisetapid ning selle kaudu on võimalik ära kasutada vanema olemasoleva mudeli tugevusi ja tehnikaid ka agiilses arenduskeskkonnas.

### 6.1.2 Kasutajalugude loomine

Üks varajastest protsessi etappidest, kus kvaliteedispetsialist saab oma panuse anda on kasutajalugude loomine. Arendusprotsessi alguses loodud kasutajalugude kvaliteet mõjutab otseselt arenduse käekäiku. *INVEST* kriteerium loodi 2003. aastal Bill Wake'i poolt, eesmärgiga luua süsteem kasutajalugude kvaliteedi kontrollimiseks. Tema poolt loodud praktika kasutamist toetavad ka teised agiilsete raamistike loojad. Üks Scrum arendusmetoodika loojatest, Mike Cohn, on selle praktika rakendamist põhjalikult kirjeldanud ja analüüsinud erinevate agiilmetoodikate raames ning peab seda edukaks meetodiks heade kasutajalugude loomisel [23]. *INVEST* on akronüüm, mille iga täht sõnas kirjeldab üht tingimust, millele hea kasutajalugu peab vastama ning selle alusel on võimalik efektiivselt hinnata kasutajaloo kvaliteeti. Juhul kui kvaliteet ei vasta mõnele tingimusele, peab meeskond analüüsima vajadust antud loo täpsustamiseks või isegi ümberkirjutamiseks.

Kõik head kasutajalood peaksid vastama järgmistele tingimustele [22, 23]:

I – *Independent* ehk sõltumatu teistest kasutajalugudest, see omadus on vajalik prioritiseerimise ja planeerimise probleemide vältimiseks. Näiteks kui kõrge väärtusega kasutajalugu on tugevas sõltuvuses madala väärtusega kasutajaloost, tekib konflikt planeerimisel, kuna arendusi viiakse läbi prioriteetide alusel.

N – *Negotiable* ehk läbiräägitav. Hea kasutajalugu ei kirjelda detailselt ette funktsionaalsust, vaid annab ülevaate loodava funktsionaalsuse sisulisest poolest. Detailid selgitatakse välja ja pannakse paika üheskoos arendusmeeskonnaga ning seda tehakse arendusprotsessi käigus. Kasutajalugude juurde võib küll lisada erinevaid märkmeid ja testiideid, kuid need ei ole hädavajalikud kasutajalugude prioritiseerimiseks ja planeerimiseks, seega pole neile ülemäära palju tähelepanu vaja pöörata. Lisaks võivad liigsed märkmed luua keerukust funktsionaalsuse mõistmiseks.

V – *Valuable* ehk väärtust lisav. Iga arendatav kasutajalugu peab lisama äripoole jaoks väärtust. Seega tuleb vältida olukorda, kus luuakse ainult tehnilist eesmärki kirjeldav kasutajalugu. Näiteks: „Vigade haldamine ja logide loomine teostatakse ühise klassi kaudu." Sellised kasutajalood on tehnilise suunitlusega ning võivad luua olulist väärtust arendusmeeskonna jaoks, kuid võivad sisuliselt jääda äripoole jaoks arusaamatuks ning tekitada selle kaudu probleeme prioritiseerimisel ja planeerimisel. Kuid sellised kasutajalood on tehnilisest vaatest olulised ning neid ei tohiks kõrvale jätta, vaid kirjeldada äripoolele

arusaadavas vormis. Näiteks: „Tekkinud veaolukordi kuvatakse kasutajale ja logitakse süstemaatilisel viisil.“

E – *Estimable* ehk hinnatav. Kuna kasutajalood on sisendiks planeerimisele, siis on vajalik, et neid oleks võimalik hinnata. Hinnatavus on osaliselt läbirääkimise omadusega, kuna raske on hinnata kasutajalugu, mida ei mõisteta. Probleemid tekivad enamasti juhtudel, kus arendusmeeskonnal puudub domeeniteadmine või tehnilised oskused vajaliku töö läbi viimiseks. Samuti tekitavad probleeme liiga suured ja üldised kasutajalood. Juhul kui arendusmeeskonna jaoks on kasutajalugu, väheste domeeniteadmiste või tehniliste oskuste tõttu raskesti arusaadav, on heaks tavaks eraldada selle loo jaoks eraldi ajaaken uurimuse läbi viimiseks ning seejärel määrata selle kasutajaloo täpsem väärtus. Seega muutub määramatu kasutajaloo hinnang kaheks: uurimuse hinnang ja kasutajaloo enda hinnang. Juhul kui kasutajalugu on liiga suur, tuleb see tükeldada väiksemateks kergemini mõistetavateks tükkideks.

S – *Small* ehk väike. Head kasutajalood on väikesed ning on teostatavad maksimaalselt nädala aja kogu meeskonna tööga. Liiga suuri ja keerukaid kasutajalugusid on mõistlik tükeldada näiteks *CRUD* liigituse alusel. Omavahel eraldatakse loomise, vaatamise, uuendamise ja kustutamise kasutajalood. Väikeseid kasutajalugusid on oluliselt lihtsam hinnata. Samas tuleb vältida liiga väikeseid kasutajalugusid, kuna need võivad tekitada meeskonnas tunde, et nende kirjeldamine ja analüüs võtab rohkem aega kui nende realiseerimine. Hea tava väga väikeste kasutajalugude haldamiseks on koguda nad kokku ning moodustada nendest üks veidi suurem kasutajalugu.

T – *Testable* ehk testitav. Testitavus tagab, et kui tooteomanik kirjeldab kasutajalugu, siis ta mõistab, mida ta soovib ja suudab selle jaoks mõne testi kirjutada. Meeskonnad on produktiivsemad, kui kasutajalood on läbimõeldud ja neile on loodud testid. Lisaks, testide loomine võimaldab arendusmeeskonnal mõista, millal kasutajalugu on valmis. Juhul kui äripool ei tea, kuidas midagi testida, võib see viidata, et kasutajalugu ei ole piisavalt läbimõeldud või seal ei ole kirjeldatud midagi äri jaoks piisavalt olulist. Oluline aspekt on ka automatiseerimisel, kui kasutajalugu on kehvasti kirjeldatud, on seda ka halb kui mitte võimatu automatiseerida.

*INVEST* kriteeriumi alusel kasutajalugude hindamine tagab, et nende kaudu luuakse dokumentatsiooni vajalikul määral ja piisava täpsusega.

### 6.1.3 Kasutajalugude täpsustamine

Lisaks kasutajalugude korrektsele loomisele tuleb neid täpsustada, et tagada piisav ja korrektne info edukaks arendustööks. Kasutajalugude täpsustamisel teeb kogu meeskond koostööd toote omanikuga ning aitab tal luua vastuvõtukriteeriumeid ja näiteid. See tegevus aitab paremini defineerida skoopi ja tuua esile ärivajadusi. Scrumile kohandatud testimise mudel (joonis 4) eeldab samuti kasutajalugude täpsustamist ja näidete loomise tehnika kasutamist.

Maailmas tunnustatud testimispraktik Gojko Adzic on kirjeldanud näidetel põhineva spetsifikatsiooni tehnika (*Specification by Example*). Antud praktika keskendub arenduse suunamisele „õige“ toote ehitamiseks, mis vastaks kliendi soovidele ja vajadustele. Kuna tulemid on läbi aegade näidanud, et sageli luuakse tarkvara igati korrektselt ja häid tehnilisi praktikaid kasutades, kuid lõplik tulem ei tee siiski seda, mida klient soovis [27]. Kuna inimkeel on mitmetähenduslik ja tugevalt sõltuvuses kontekstist, siis on suur risk, et erinevad osapooled tõlgendavad kasutajalugu erinevalt. Seetõttu tulebki riski vähendamiseks luua kasutajalugude täpsustamise etapis näiteid, mis defineerivad, mida arendatav toode tegema peab. Loodud näiteid kasutatakse sageli ka vastuvõtukriteeriumitena. Vastuvõtukriteeriumite loomisel tuleb keskenduda „mida“ tarkvara peab tegema, mitte „kuidas“. Soovitatav on luua mõned positiivse stsenaariumiga ja mõned veaolukordade või alternatiivse voo näited.

Kasutajalugude täpsustamiseks ja näidete loomiseks on mitmeid vahendeid [9]:

- Kontrollnimekirjad;
- Mõttekaardid;
- Arvutustabelid;
- Prototüübid;
- Töövoo diagrammid.

Näidete loomist kasutatakse üheskoos teise praktikaga – vastuvõtutestide põhise arendusega (*Acceptance Test Driven Development*) [28]. Ühe kasutajaloo vastuvõtukriteeriumiga võib seotud olla mitu vastuvõtutesti. Vastuvõtuteste luuakse koos meeskonnaga enne koodi kirjutamist ning need kirjeldavad sprinti võetud kasutajaloo oodatud käitumist. Kasutajalugude täpsustamisel antud tehnikat kasutades paraneb meeskonna arusaamine

arendatava kasutajaloo sisust ja väheneb möödarääkimiste risk. Lisaks tekib koos töötades ühisomandi tunne ning seeläbi kasvab ja jaguneb vastutustunne meeskonnas.

Testid kirjeldatakse võimalusel kasutades automatiseerimisvahendeid ning sellise pideva automatiseerimise kaudu luuakse „elav dokumentatsioon“ vastuvõtutestidest. Elava dokumentatsiooni loomiseks kasutatakse sobivaid vahendeid, mis võimaldavad teste kirjeldada inimloetavas keeles ning anda ülevaadet rakenduse funktsionaalsustest. Selline praktika võimaldab teste kirjeldada ja mõista ka äripoolel, kuid senised kogemused on näidanud, et antud praktika võib ka tekitada testide haldamise probleeme [27]. Seetõttu peab iga meeskond hindama vajadust ja riski, milliseid teste ja mis tasemel on tarvilik luua.

### 6.1.4 Testimise kaasatuse parendamise soovitused

Läbi viidud analüüsi alusel koostas autor järgnevad soovitused testimisprotsessi parendamiseks:

1. Kasutajalugude kirjeldamise kvaliteedi hindamiseks kasutada peatükis 6.1.2 kirjeldatud *INVEST* kriteeriumit;
2. Kirjeldada projekti testimisprotsess ning teha kliendiga ja meeskonnaga vastavad kokkulepped, võtta kasutusele Scrumile kohandatud protsess;
3. Leppida kliendiga kokku kvaliteedispetsialisti vastutused ja töö tulemid;
4. Kirjeldada testimise eesmärk, skoop ja riskid nii sprindi kui tarne põhiselt;
5. Võtta kasutusele sprindi planeerimise koosolekutel ja kasutajalugude täpsustamise koosolekutel näidete loomise ning vastuvõtutestide loomise praktika;
6. Luua vastuvõtutestide alusel automatiseeritud testid, millest saab alus „elavaks dokumentatsiooniks“.

## 6.2 Testimise strateegia ja protsessi haldus

„Testimise strateegia“ ja „Testimise protsessi haldus“ on teineteisega tihedalt seotud, seetõttu vaatleme neid käsitlevaid praktikaid koos. Antud võtmealade kontrollpunktide täitmiseks on tarvilik leida lahendus riskianalüüsi, testistrateegia ja –plaani koostamiseks.

### 6.2.1 Testiplaan

Laialt levinud vale arusaam agiilsest arendusest on see, et enam ei vajata planeerimist, kuna agiilses manifestis [17] on kirja pandud põhimõte, mis kirjeldab, et kohanemist vastavalt muutustele väärtustatakse rohkem, kui plaanide järgimist. Tegelikult võib väita, et tendents on pigem vastupidi, kuna edukad agiilsed meeskonnad võivad vahel planeerimisele kulutada isegi rohkem aega, kui traditsionaalseid arendusmetoodikaid jälgivad meeskonnad. Erinevus seisneb aga selles, et planeeritakse väikestes osades ja ainult nii palju ette kui vaja. Sama reegel kehtib testimise planeerimisel [9].

Scrum arendusprotsess näeb ette, et edukas sprint lõppeb potentsiaalselt valmis tarkvaraosaga, mida on võimalik lasta toodangusse. Just see eesmärk toob esile testimise olulisuse sprindi jooksul. Kõik funktsionaalsused ja detailid, mis valmivad sprindi jooksul peavad saama ka testitud sprindi ajaakna sees, seega peavad kõik testimistegevused olema planeeritud tervikliku sprindi osana.

Täidetud ülesandeid võib lugeda valminuks, kui need vastavad Valmimise kriteeriumitele (*Definition of Done*). Iga ülesande valmimisel peab selle täitja veenduma, et kõik valmimise kriteeriumid on täidetud ning need vajadusel täitma, selline protsess võimaldab tähelepanu koondada tegevustele, mis tihipeale jäetakse tegemata. Näiteks koodi läbivaatused, uute testiideede kirjutamine, dokumentatsiooni loomine jne. Selline reegel tõstab küll ülesannete täitmise keerukust ja mahtu, kuid võimaldab tagada, et kõik ülesanded on täielikult valmis, kui sprint on lõppeb. Valmimise kriteeriumis testimisülesannete kirjeldamine on üks enamlevinuid praktikaid testimise fookuses hoidmiseks.

Traditsionaalne testimispraktika ja standardid (nt IEEE-829) näeb ette, et projektile luuakse üldine testiplaan ning vajadusel ka lisa testiplaane. Need dokumendid on aga enamasti suured ja mahukad, ning kuna agiilse arendusprotsessi põhimõtted on oluliselt teistsugused, on need dokumendid kaotanud olulisuse oma esialgsel kujul ning on asendunud näiteks mõttekaartide ja ühe leheliste testiplaanidega [3].

Janet Gregory ja Lisa Crispin [9] on pakkunud testimise planeerimise jaoks detailsustasemed, mida kvaliteedispetsialist peab planeerimisel arvesse võtma. Planeerimiseks nad kirjeldanud neli taset:

- **Toote tarne** (*Product Release*). Antud tase kirjeldab toote visiooni, tarneplaane ja kuupäevi ning tarnitavaid funktsionaalsuseid. Sellel tasemel peab arvestama ja planeerima, milliseid teste on tarvilik läbi viia toote lõpliku tarne korral, samuti on vaja ette mõelda, milliseid ressursse nende testide läbi viimiseks on vajalik kaasata (nt lõppkasutajad, kolmandad osapooled jt.).

- **Funktsionaalsus** (*Feature*). Funktsionaalsused kirjeldavad ärivajadusi, mis tükeldatakse kasutajalugudeks. Sellel tasemel peab võtma arvesse, et terve loodav funktsionaalsus peab olema põhjalikult testitud ning selleks tuleb luua vastavad võimalused. Näiteks kui ühte funktsionaalsust luuakse läbi mitme sprindi, mille jooksul testitakse seda ositi, siis peab lõpuks testima ka tervikvaadet ning hindama mõju kogu süsteemile. Samuti on sellel tasemel hea planeerida võimalikke eritüübilisi testimisi, näiteks koormus- ja turvatestimist.

- **Kasutajalugu** (*Story*). Väike, testitav funktsionaalsuse osa, mis loetakse lõpetatuks, kui see vastab Valmimise kriteeriumile ning enamasti arendatakse seda kaks kuni kolm päeva. Enamasti ei ole kasutajalugu eraldiseisvana tarnitav osa ning selle realiseerimiseks tuleb täita mitu ülesannet. Sellel tasemel testitakse suhteliselt detailselt. Igal lool on defineeritud vastuvõtukriteeriumid, mida võib testida vastavalt kõikidele kvaliteedispetsialisti oskustele. Enamasti kasutatakse sellel tasemel uurivat testimist ja teisi ekspertteadmistel põhinevaid tehnikaid. Kvaliteedispetsialist peaks testilugusid looma hakkama alates sellest hetkest, kui tooteomanik loob kasutajaloo ning lisama teste sprindi jooksul.

- **Ülesanne** (*Task*). Sellele tasemele tuleks planeerida arendaja poolt loodavad testid (*TDD*), kuna need on disaini eesmärgil loodavad. Kvaliteedispetsialist saab aga ülesannete tasemel toetada arendajate tegevust näiteks testandmete loomisega, mis võib keerukamates keskkondades osutuda tõsiseks väljakutseks.

### 6.2.2 Testiplaan ja agiilsed kvadrandid

Testiplaan on aluseks igale testimistegevusele, mis projektis läbi viiakse. See kirjeldab, mida testitakse ja kuidas seda tegema hakatakse. Tabelis 6 on toodud Janet Gregory ja Lisa Crispin'i agiilse testimise kvadrandid [9], mis võimaldavad luua terviklikku testiplaani ja strateegiat.

**Tabel 6. Agiilse testimise kvadrandid**

| Arendust juhtivad testid (vigade ennetamiseks) | Ärilise suunitlusega testid | | Tootepõhised testid (vigade leidmiseks) |
|---|---|---|---|
| | Näitedetel põhinevad testid<br>Kasutajamugavuse testid<br>Prototüübid<br>Simulatsioonid Q2 | Q3 | Uuriv testimine<br>Töövood<br>Süsteemi integratsioon<br>Vastuvõtutestid |
| | Ühiktestid Q1<br>Komponenttestid | Q4 | Koormus- ja jõudlustestid<br>Turvatestid<br>Teised kvaliteediatribuudid |
| | Tehnilised testid | | |

Kirjeldatud kvadrantide tähiste numbrid ei oma eraldi väärtust, need on loodud lihtsalt tähisteks. Ehk testimist ei planeerita nende järjekorranumbri alusel ja need ei kirjelda kvadrantide olulisust. Nagu joonisel näha, jagunevad kvadrandid nelja gruppi.

- Q1 - tehnilised testid, mis juhivad ja toetavad arendajaid;
- Q2 - ärisuunitlusega testid, mille eesmärk on juhtida arendust;
- Q3 - ärisuunitlusega testid, mis uurivad süvitsi valmivat toodet;
- Q4 - tehnilised testid, mis on suunatud toote võimekuse hindamiseks.

Vasak pool maatriksist kirjeldab erinevaid tehnikaid vigade ennetamiseks programmeerimise ajal ja enne seda. Parem pool kirjeldab tehnikaid, mille abil on võimalik vigu ja puudujääke avastada pärast koodi kirjutamist. Maatriksi ülemine pool kirjeldab teste, mis on suunatud äripoole vajadustele ning alumine pool kirjeldab meeskonna ja toote edukuse jaoks loodavaid teste.

Kvadrantide kirjelduse kaudu saab ka kinnitust, et peatükis 6.1.1 kirjeldatud erinevad testimisetapid on olulisel kohal ka agiilsetes praktikates. Kvadrandis Q1 on kirjeldatud nii ühik- ja integratsioonitestid, mida loovad enamasti arendajad ning testid on automatiseeritud.

Kvadrandis Q2 on vastuvõtu- ja regressioonitestid, mida viiakse läbi nii automatiseerimise kui ka manuaalse testimise abil. Kvadrandis Q3 paiknevad erinevad süsteemitestid ja ekspertteadmiste põhised testid ning neid ei ole enamasti võimalik automatiseerida. Kvadrant Q4 on loodud teiste kvaliteediatribuutide (nt jõudlus, turvalisus) testimise kirjeldamiseks ning need testid vajavad alati lisateadmiste ja -vahendite abi.

### 6.2.3 Testistrateegia

Testistrateegia loomisel on vajalik valida, millised tehnikad on piisavad loodava toote kvaliteedi tagamiseks ja millal on vajalik neid läbi viia. Agiilse testimise kvadrandid annavad hea ülevaate võimalustest ning selle alusel saab hinnata vajalikke riske ning langetada otsused, milliseid tehnikaid kasutada. Kõik ettevõtte projektidele sarnased nüansid kirjeldatakse selles dokumendis. Seda dokumenti saab kasutada näiteks kliendile, uutele töötajatele või meeskonna liikmetele kiire ülevaate andmiseks, kuidas testimise protsess toimib.

Testimise strateegia peaks vastama järgmistele küsimustele [9]:

1. Kes vastutab testimise eest? Mis hetkel kvaliteedispetsialist hakkab testima, mis keskkonnas ta testib?
2. Millist testandmestikku kasutatakse (nt luuakse automaatselt testandmed)?
3. Mis tasemel luuakse automaattestid ning milliseid vahendeid selleks kasutatakse?
4. Milliseid testimise tehnikaid kasutatakse?
5. Kuidas luuakse koormus- ja integratsioonitestid? Mis vahenditega?
6. Kuidas riske hinnatakse ja maandatakse?
7. Kuidas hallatakse vigu?

Lisa Crispin ja Janet Gregory on kirjeldanud tabelis 7 oleva testistrateegia, mis katab üldiselt eelnevalt püstitatud küsimuste vajadused.

**Tabel 7. Testistrateegia**

| | |
|---|---|
| Projekti algus | Üleüldise ülevaate ja arusaama saavutamine |
| Tarne planeerimine | Kasutajalugude hindamisel osalemine,<br>Riskide analüüs hindamisel,<br>Testiplaanide loomine |
| Sprint 1 … N | Sprindi planeerimine, tööde hindamine<br>Arenduskeskkonnas testimine,<br>Kasutajalugude testimine,<br>Kliendiga koos arenduse testimine,<br>Automattestide loomine ja jooksutamine,<br>Regressioonitestide loomine,<br>Esitlemine toote omanikule ja teistele osapooltele. |
| Süsteemitestimine | Toodangulaadses keskkonnas testimine,<br>Koormustestid,<br>Täielikud regressioonitestid,<br>Äripoole vastuvõtutestid |
| Tarne toodangusse | Toodangus testimine |

### 6.2.4 Riskianalüüs

Riskianalüüs lähtub põhimõttest: „Pole riski, pole testi!“. Testistrateegia kirjeldab testimise eesmärgi, kuidas need saavutatakse ja kuidas riske maandatakse. Agiilses keskkonnas kasutatakse ühe riskihaldamise vahendina tööde prioritiseerimist. Riski põhine lähenemine on Scrumi juba sisse ehitatud, kuna meeskond lähtub alati sprinti võetavatest töödest, mis on prioritiseeritud ja hindamisel on arvesse võetud ka riske.

Lisa Crispin ja Janet Gregory on oma raamatus „Agile testing“ [29] soovitanud järgnevat riskianalüüsi meetodit. Antud meetodit saab kasutada nii üksiku kasutajaloo hindamiseks kui ka suuremate ja keerukamate puhul. Kasutajaloo hindamisel tuleks koostada nimekiri potentsiaalsetest riskidest. Seejärel tuleks hinnata nende põhjustatava kahju mõju (*impact*) hindega 1 kuni 5, milles 1 tähendab madalat ja 5 kõrget riskitaset. Seejärel tuleks samamoodi hinnata nende toimumise tõenäosust (*probability*). Saadud tulemid tuleks oma vahel korrutada

ning reastada pingeritta. Sellisel meetodil on lihtne valida, milliste riskide maandamisega on vajalik rohkem tegeleda ning millised riskid on vähemolulised.

### 6.2.5 Testistrateegia parendamise soovitused

Ignite OÜ meeskonnad teevad tööd ka välistele partneritele, seega alati ei saa ühesugust testistrateegiat kasutada, vaid kohanetakse kliendi valitud strateegiate ja reeglitega. Samas nendel juhtudel, kui tegemist on mitte tavakliendiga või kliendiga, kellel ei ole loodud testimisstrateegiat, on oma testistrateegia olemasolu vajalik.

Läbi viidud analüüsi alusel koostas autor järgnevad soovitused testistrateegia parendamiseks:

1. Kirjeldada üleüldine testimisstrateegia, mis on leitud ühisosana vaadeldes erinevaid projekte ning kasutades agiilseid testimise kvadrante ja testistrateegia küsimustikku;
2. Arendusprotsessi alguses kooskõlastada koos kliendi ja meeskonnaga üldine testimisstrateegia;
3. Testimistehnikate kasutus tuleb kirjeldada testistrateegias;

Järgnevalt on kirjeldatud autori koostatud soovitused testimise protsessi halduse parendamiseks:

1. Vastuvõtu-, integratsiooni- ja ühiktestide edukas läbimine tuleb kirjeldada Valmimise kriteeriumites;
2. Kvaliteedispetsialist kirjeldab koos tooteomanikuga tarne testiplaani:
    - Toote omanik kirjeldab visiooni ja kuupäevad;
    - Kaardistatakse huvitatud osapooled ja nende vajadused;
    - Plaani täiendatakse pidevalt sprintide jooksul kuni toote tarneni;
    - Võetakse vaatluse alla ka standardi ISO-9216 kvaliteediatribuudid;
3. Valmimise kriteeriumisse (*Definition of Done*) lisatakse testimise läbimine;
4. Iga sprint kirjeldatakse kasutajaloo tasemel testiideed (lähtudes vastuvõtukriteeriumitest):

- Positiivsed stsenaariumid;
- Veaolukorrad;
- Iga sprint kaetakse uus kood ühiktestidega (~80% ulatuses);

5. Sprindi testiplaan:
    - Iga sprint loodud uuendused kirjeldatakse sobival kujul, et hiljem oleks võimalik analüüsida, milliseid funktsionaalsuseid tarnitakse;
    - Iga sprint täiendatakse funktsionaalsuse põhist regressiooniplaani, hinnatakse riske, kuidas võivad teised arendused mõjutada ning milliseid teste tuleks lisaks läbida;
    - Iga sprint valideeritakse uut koodi staatiliste koodianalüüsi vahenditega;
    - Iga sprindi lõpus on kvaliteedispetsialist võimeline andma tooteomanikule ülevaadet arendatud funktsionaalsusest ja selle kvaliteedist;
    - Enne sprindi algust luuakse prototüüpe ning kinnitatakse need äripoolega;

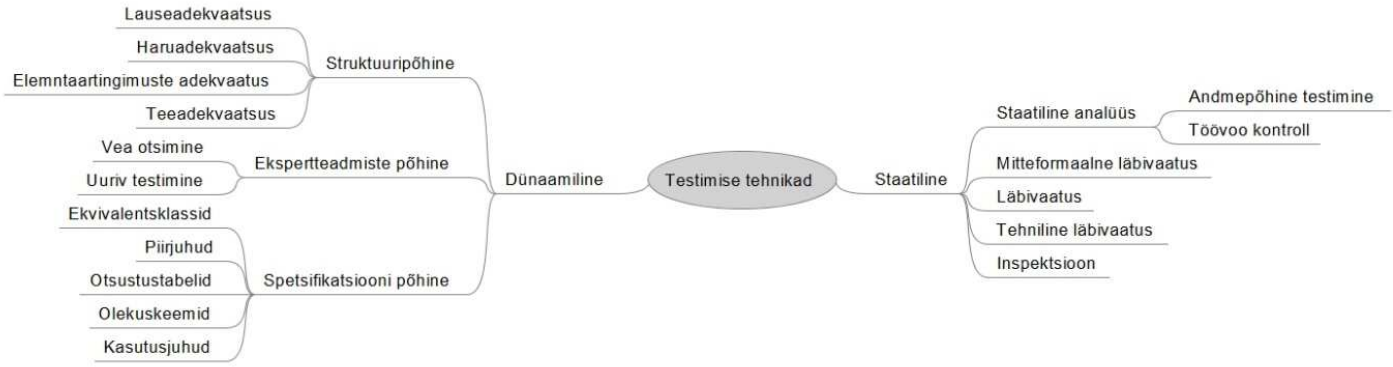
## 6.3 Testidisain

„Testidisaini“ võtmeala „A“-klastri kontrollpunktide täitmiseks on tarvilik kirjeldada parimad praktikad testilugude loomiseks ja kirjeldamiseks.

Süsteemi kvaliteedi hindamiseks tuleb läbi viia erinevaid teste, kuna süsteemid koosnevad enamasti mitmest osast, mis vajavad erinevaid teste ja testimise põhjalikkust. Neid teste tuleb aga läbi viia minimaalse kuluga ja eesmärgiga avastada maksimaalselt olulisi vigu. Seetõttu on testidisainil väga oluline koht ning disainimiseks on võimalik kasutada formaalseid testidisaini tehnikaid [3]. Nende tehnikate korrektsel kasutamisel on mitmeid eeliseid:

1. Keskendutakse süstemaatiliselt võimalike vigade avastamisele kasutades selleks sobivaid tehnikaid;
2. Testimist viiakse läbi erinevatel tasemetel (arendus-, vastuvõtutestid);
3. Testimise täielikkust on võimalik seotud osapooltele selgitada ja põhjendada.

Joonisel 5 on kirjeldatud testimise disainitehnikad [19].




**Joonis 5. Testimise disainitehnikad**

Enamlevinud testimistehnika, mida kvaliteedispetsialistid traditsionaalsetes projektides kasutavad, on dünaamiline testimine (joonisel 5 kujutatud vasakpoolne haru), mille käigus võrreldakse testitava süsteemi käitumist oodatud tulemiga. Scrum meeskonna kvaliteedispetsialistil läheb aga vaja teadmisi nii staatilisest kui dünaamilisest testimisest, et tagada võimalikult kiire tagasiside testide kaudu. Enamasti ei täpsustada planeerimisel väga detailselt kasutajalugude nüansse, seega tuleb spetsifikatsioonipõhiste tehnikate kasutamiseks varasemast rohkem koostööd teha tooteomanikuga ning näha rohkem vaeva. Eksperteadmiste põhised ja struktuuripõhised tehnikad on agiilsetes praktikates laialdaselt kasutusel ja kõrgelt hinnatud. Seega on võimalik kõiki (*ISQTB* kirjeldatuid) olemasolevaid tehnikaid kasutada testide loomiseks [11].

### 6.3.1 Manuaalne testimine

Joonisel 5 on välja toodud eksperteadmistel põhinev tehnika – uuriv testimine. Uuriv testimine on agiilses keskkonnas väga populaarne testimispraktika, mis seob omavahel loodava süsteemi uurimise ja manuaalse testimise. Antud tehnika riskiks on tugev sõltuvus kvaliteedispetsialisti oskustest, kogemustest ja distsipliinist. Lisaks on uuriva testimisega kerge kursilt kõrvale kalduda ning seetõttu langeb oluliselt vigade leidmise efektiivsus [3].

Toodud riskide maandamiseks kasutatakse uuriva testimise juhtimiseks sessioonipõhist testimist. Sessioonipõhine testimine koosneb järgnevatest punktidest:

- Kvaliteedispetsialist valib testimise dokumenteerimiseks vahendi ja kirjeldab lühidalt testimise eesmärgi;

- Sessioon kestab maksimaalselt 90 minutit ning koosneb ettevalmistusest, testi disainist ja läbiviimisest, vigade täpsustamisest ja raporteerimisest;

- Sessiooni raport luuakse lühikese kokkuvõttega tehtud tööst (eesmärk, süsteemi detailid, testide katvus, komponentide kirjeldus, leitud vead, vastuseta jäänud küsimused).

Sellisel meetodil on hiljem lihtsam analüüsida tehtud tööd ning teha järeldusi. Lisaks uurivale testimisele viiakse manuaalselt läbi ka kasutajamugavuse testimist ja süsteemitestimist enne lõplikku tarnet.

#### 6.3.1.1 Testiloo kirjutamine

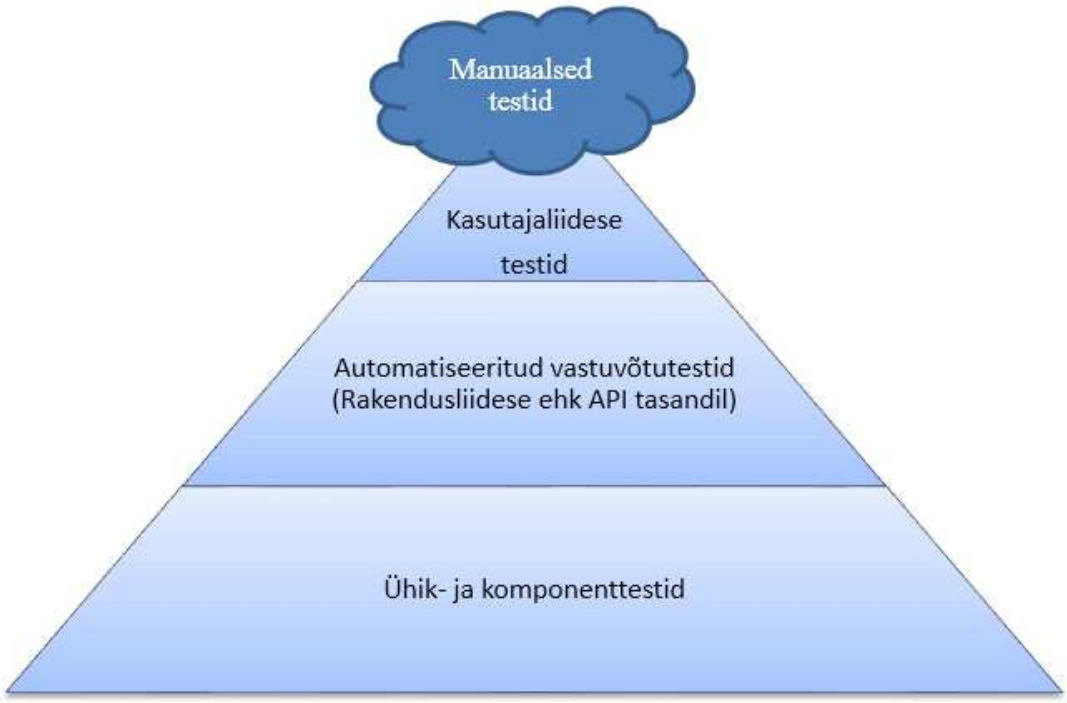
Testilugude loomiseks ja kirjutamiseks on praktikud loonud näidisnimekirju erinevate testiideede ja sisendandmete loomiseks, näiteks Elisabeth Hendrickson, James Lyndsay ja Dale Emery poolt loodud „*Test Heuristics Cheat Sheet*“ [25] ning Michael Hunteri poolt kirjeldatud „*You Are Not Done Yet checklist*“ [31].

Agiilses keskkonnas kirjeldatakse testilugusid kõrgtasemel ning lepitakse kokku meeskonnale sobiv formaat. Testiloos tuleks kirjeldada minimaalne info, mis on vajalik testi läbi viimiseks. Sageli kasutatakse ka loodud vastuvõtukriteeriumite analüüsi ja nende laiendamist. Lisa Crispin’i soovitatud mall [30] kasutajaloo testide kirjeldamiseks on lisatud lisasse 5.

Ideaalis tuleks testilugude kvaliteedi hindamiseks nende loomisel jäädvustada kasutatud testidisaini tehnika, selle alusel on kvaliteedispetsialistil hiljem võimalik analüüsida tulemeid ja teha vajadusel parandusi [11].

### 6.3.2 Testimise automatiseerimine

Mike Cohn'i loodud testide automatiseerimise püramiid (joonis 6) toob esile, et automatiseeritud testide aluseks on ühik- ja komponenttestid. Püramiidis on kolm taset, milles iga kõrgem tase tugineb alumisel.




**Joonis 6. Testide automatiseerimise püramiid**

Kõige alumist kihti iseloomustavad robustsed ja koodilähedased testid, mis luuakse enamasti samas programmeerimise keeles, mida kasutatakse arendamiseks. Nendel testidel on kõige kõrgem investeeringutasuvus (*Return on investment*), kuna testid läbitakse väga kiiresti ning nende kaudu saadakse pidevat ja kiiret tagasisidet, samuti on nende loomise kulud madalad. Agiilses keskkonnas on hädavajalik, et arendust toetaksid testid, mis on loodud nii madalal tasemel (ehk koodi lähedal) kui võimalik. Madalataseme teste automatiseeritakse ülesannete tasemel (*task level*) ehk kui arendaja on lõpetanud ülesande täitmise, peaks sellel olemas olema ka vastav test.

Keskmisel tasandil asuvad testid on loodud äriloogika testimiseks. Kui madalataseme testid on loodud tehniliste nüansside testimiseks, siis sellel tasemel on põhifookuses kasutajalugude ja vastuvõtukriteeriumite põhised testid. Testid luuakse rakendusliidese tasandil ehk ilma kasutajaliidest kasutamata, see võimaldab keskenduda äriloogika testimisele ning hoida testide loomise ja hooldamise kulud madalamad. Võimalusel luuakse need testid inimloetavas keeles (*domain-specific language*), mis võimaldab äripoolel testidest aru saada ja neid ka ise

luua. Nende testide läbimine on oluliselt aeglasem ja testide loomine tehniliselt keerulisem kui ühik- ja komponenttestide puhul, kuna testide arhitektuur on keerulisem. Näiteks kasutatakse neid teste andmebaasi suhtluse ja teiste komponentide integratsiooni testimiseks. Seetõttu ei ole nende testide investeeringutasuvus sama kõrge kui madalamal tasemel, kuid on siiski oluliselt väärtuslikumad kui kasutajaliidese kaudu loodavad testid. Antud testide loomiseks kasutatakse enamasti raamistikke, võimalusel kasutatakse olemasolevaid, kuid sobivate puudumisel luuakse see ise. Keskmisel tasemel automatiseeritakse kasutajaloo teste (*story level*) ehk kui meeskond on lõpetanud kõik kasutajalooga seotud ülesanded, siis peab sellele kasutajaloo funktsionaalsusele looma ka vastavad testid.

Kõige kõrgemal tasemel paiknevad testid, millel on kõige madalam investeeringutasuvus. Need testid simuleerivad kasutajaliidese kasutamist ning on selletõttu suhteliselt haprad, kuna on väga tundlikud kasutajaliidese muutustele, mis on aga pidevas arendusprotsessis väga tavalised. Testide käivitamine on aeglane. Samas omavad kasutajaliidese testid väga olulist rolli kogu testimisprotsessis, kuid nende osakaal kogu automatiseeritud testide hulgas peaks olema võrdlemisi väike ning moodustama ainult püramiidi tipu. Sellel tasemel automatiseeritakse funktsionaalsuse (*Feature level*) taseme teste, mis tähendab, et luuakse kõige üldisema sisuga testid.

Püramiidi tipus on pilvena kujutatud manuaalne testimine, mis kirjeldab, et sõltumata automaattestide rohkusest vajab iga süsteem lisaks veel manuaalset testimist. Samas kehtib reegel, et kui regressioonitestimine ei ole vajalikul määral automatiseeritud, siis ei saa ka manuaalne testimine luua olulist lisaväärtust.

### 6.3.3 Automatiseerimise disain

Automatiseerimise püramiid on hea alguspunkt testimise automatiseerimise juurutamiseks agiilses meeskonnas. Enamasti keskenduvad arendajad kõige madalama taseme testidele ning kvaliteedispetsialistid pühenduvad enamasti kõige kõrgema ehk kasutajaliidese testide automatiseerimisele. Multifunktsionaalses meeskonnas on võimalik tagada efektiivne testide loomine, näiteks kasutades paarisprogrammeerimise võtteid kvaliteedispetsialisti ja arendaja koostöös ning selle kaudu on võimalik saavutada väga hea testide katvus ka keskmisel püramiidi tasemel.

Lähtuvalt agiilsetest kvadrantidest tuleb analüüsida, mida on vajalik automatiseerida ja mida mitte. Lisaks tuleb alati silmas pidada investeeringutasuvust ning riske, mida loodavad testid peaksid vähendama.

Kasutajaliidese testid omavad madalat investeeringuväärtust ning seetõttu tuleb nende loomisel olla väga tähelepanelik. Testimist tuleks alustada nii madalalt kui võimalik, et hoida testid efektiivsete ja robustsetena.

Testimise automatiseerimisega alustamiseks on vajalik mõista, kust alustada ja selleks tuleb leida, millised on arendatava toote kõige kriitilisemad osad. Näiteks valitakse sageli esimesena kas ärikriitiline funktsionaalsus, kasutajate poolt enam kasutatavad funktsioonid või testid ja testiideed, mida käivitatakse paljude sisendandmete või tingimustega.

Automatiseerimise püramiidi kõige alumine tase kuulub peatükis 6.2.1 kirjeldatud agiilse testimise kvadranti Q1. Ilma kvadrant Q1 testideta on väga raske, kui mitte võimatu, tagada teiste kvadrantide testimise edukust. Ideaalses protsessis jooksutatakse ühik- ja integratsiooniteste igal korral, kui koodi muudatus jõuab versioonihaldussüsteemi. Selline protsess võimaldab meeskonnale anda kiiret tagasisidet ning Scrum on üles ehitatud tagamaks kiiret tagasisidet.

Agiilses arenduskeskkonnas on enamlevinud alt-üles testimisstrateegia. Nagu eespool kirjeldati omavad madala taseme testid kõige kõrgemat investeeringutasuvust, seega on äärmiselt oluline, et just neid luuakse korrektselt ja õigeaegselt.

**Ühiktestimine** (*Unit testing*). Testid annavad kiiret tagasisidet. Testid on koodi dokumentatsiooniks ning selle kaudu tagatakse parem testitavus, kuna testide kirjutamine mõjutab otseselt loodud koodi struktuuri. Ühiktestide loomisel saab kasutada järgmiseid testidisaini tehnikaid:

- Olekuskeemide kasutamine;
- Avalike meetodite testimine;
- Teeadekvaatsuse testimine;
- Haruadekvaatsuse testimine;

**Integratsioonitestimine** (*Integration testing*). Võimaldab kontrollida erinevate tarkvara osade koostoimivust. Integratsioonitestide loomisel kasutatakse järgmist disaini protsessi ja tegevusi:

- Analüüsitakse osade omavahelist suhtlust;
- Kaardistatakse ekvivalentsklassid;
- Kaardistatakse sisendandmed;
- Defineeritakse sidend ja väljund andmete võrdlused (oodatud käitumise analüüs);
- Vajadusel disainitakse testid asünkroonse suhtlemise jaoks (vajalikud juhul kui on kasutusel näiteks hajutatud ressursid).

**Automaattestide kirjutamise hea tava [3]:**

1. Testide meetodid on nimetatud vastavalt sooritatava testi sisule, see tagab testide lihtsa loetavuse ja kergema hooldamise;
2. Iga test sisaldab nelja osa: ettevalmistus, testi protsess, kontroll, lõpetamine (*teardown*);
3. Kontroll verifitseerib, kas saadud tuleb vastab oodatud tulemile. Kui need on võrdsed, siis test loetakse läbituks, muudel juhtudel loetakse läbikukkunuks;
4. Teste saab jooksutada ükshaaval või kõiki koos.

### 6.3.4 Testidisaini ja –tehnikate parendamise soovitused

Läbi viidud analüüsi alusel koostas autor järgnevad soovitused testidisaini ja -tehnikate parendamiseks:

1. Testistrateegias ja –plaanis tuleb kirjeldada sobivad testimistehnikad ning neid jälgida;
2. Testilugusid luuakse ühtse malli alusel (nt lisas 5 toodud näide);
3. Testilugude loomisel kasutatakse peatükis 6.3 kirjeldatud struktureeritud testidisainitehnikaid ja peatükis 6.3.1.1 kirjeldatud heuristikaid;

4. Uuriva testimise juhtimiseks kasutatakse peatükis 6.3.1 kirjeldatud sessioonipõhist testimist;

5. Automatiseeritud testide loomisel tuleb lähtuda automatiseerimise püramiidi põhitõdedest ning luua esmalt ühik- ja komponenttestid;

6. Automaattestid tuleb luua ühik- ja integratsioonitestide disainitehnikaid jälgides;

7. Loodavad automaattestid peavad vastama automaattestide heale tavale;

# 7. Parimate praktikate rakendamise analüüs

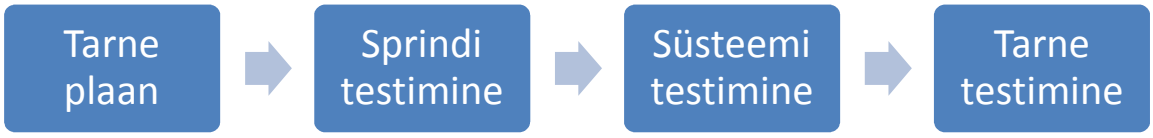
Järgnevalt on analüüsitud eelnevalt toodud parimate praktikate ja autori poolt koostatud soovituste rakendamine reaalse projekti näitel. Loetavuse lihtsustamiseks on toodud lisas 7 eraldi tabel, kus on kirjeldatud iga võtmeala kontrollpunktid, soovitused ja nende rakendamise staatus. Antud peatükk kirjeldab struktureerimise metoodika kontrollimise etappi ning eraldi peatükina on toodud korrigeerimise etapp.

Lähtuvalt küpsusmaatriksi tulemist parendati valitud võtmealasid. Meeskonna retrospektiivide ajal hinnati parendusettepanekuid, nende vajalikkust ja milliseid muutuseid protsessis oleks tarvilik näha ning mida on võimalik rakendada. Meeskonna sprindi pikkus oli 2 nädalat. Igal sprindil võeti eesmärgiks alustada ühe võtmeala parendamisega, et saada ülevaadet, mis aja jooksul on võimalik protsessi antud metoodika alusel parandada. Kokkuvõttes parendati tulemusi 2 kuu ehk 4 sprindi jooksul. Olemasoleva protsessi kaardistamiseks kulus varasemalt 2 sprinti, ehk 1 kuu.

Esimese ja teise sprindi jooksul võeti vaatluse alla testistrateegia ja –plaanide loomine, selleks, et saada ülevaadet, kuidas testimist läbi viia. Antud aspektide parendamine võimaldas võtmelade „Testimise strateegia“ ja „Testimise protsessi haldus“ puudujääkidega tegeleda.

### 7.1.1 Testimise strateegia

Testimise strateegia koostamiseks tehti koosolek, mille vältel selgitati välja põhilised testimise eesmärgid/probleemid ja kuidas neid lahendada. Selle põhjal koostas kvaliteedispetsialist testimisstrateegia, mille kinnitas tooteomanikuga. Testistrateegia katab põhilised punktid testimise läbiviimise, raporteerimise ja tulemite edastamise osas. Joonisel 7 on toodud üldine kokkulepitud strateegia testimistegevuste juhtimiseks.




**Joonis 7. Testimise strateegia**

Järgnevalt on kirjeldatud kasutusele võetud testistrateegia. Testimisstrateegia loomisega loodi selge alus testimistegevustele. Pärast strateegia kasutuselevõttu tõusis meeskonda kuuluva kvaliteedispetsialisti rahulolu.

**Skoop.** Rakendus luuakse ettevõtte siseseks kasutamiseks, mida hakkavad kasutama koolitatud spetsialistid, seega on kasutusmugavus ja tehniline korrektsus äärmiselt oluline. Rakendus on strateegilise olulisusega kliendi jaoks, lisaks puudutab arendus palju teineteisest sõltuvuses olevaid komponente. Testimist viiakse läbi riskipõhise testimise alusel. Sprintide ja tarne testiplaanide loomisel lähtutakse agiilse testimise kvadrantidest ja riskidest. Lisaks tagatakse planeeritud testimine peatükis 6.2.1 toodud planeerimise tasemel.

**Testimistehnikad.** Projekti testimiseks kasutatakse nii dünaamilisi kui staatilisi testimistehnikaid (uurivat testimist, teeadekvaatsuse analüüsi, koodi läbivaatus ja staatilist analüüsi). Uuriva testimise juhtimiseks kasutatakse sessioonipõhist testimist.

**Vastutusalad.** Testimine on kogu meeskonna vastutus. Testimiseks loovad arendajad ühik- ja komponentteste ning kvaliteedispetsialist loob automatiseeritud kasutajaliidese teste (positiivsete testilugude näol) ja aitab integratsioonitestide loomisel. Töövoo juhtimiseks ja läbipaistvaks muutmisel kasutatakse projektijuhtimise tarkvara (*Jira*), milles kvaliteedispetsialist kirjeldab testimisülesanded ja nende mahu.

**Raporteerimine ja meetrika.** Pideva töö tulemustest annab kvaliteedispetsialist ülevaate püstjala koosolekutel (*Daily Scrum*). Iga sprindi jooksul loodud testilood salvestatakse projektijuhtimise tarkvara abil ning selle kaudu luuakse läbipaistvus kasutajalugude ja läbitud testide vahel. Pidevalt jälgitakse sprintide jooksul saavutatud koodikaetust ning loodud testilugude ja leitud vigade arvu. Eraldi mõõdikuna peetakse arvet varasemate sprintide vigade üle, mille alusel on võimalik hinnata testimise efektiivsust. Iga sprindi lõpus annab kvaliteedispetsialist tooteomanikule ülevaate testimise seisust, kasutades tabelarvutuse tarkvara, mille tulem kantakse sisse projektijuhtimise tarkvarasse.

**Vigade haldamine.** Vead parandatakse sama sprindi jooksul, millal need leitakse. Juhul kui ei ole võimalik vigu kohe parandada raporteeritakse need projektijuhtimistarkvaras ning lisatakse tööde nimekirja. Vead prioritiseeritakse planeerimise koosoleku käigus. Kvaliteedispetsialist läbib esitlusel kirjeldatud vastuvõtukriteeriumid ja uuriva testimise käigus loodud testid.

**Keskkonnad ja vahendid.** Loodud testilugude haldamiseks kasutatakse projektijuhtimise tarkvara. Automatiseeritud testide koodi salvestamiseks ja haldamiseks kasutatakse versioonihaldussüsteemi (*SVN*, *Git*). Ühik- ja komponenttestid luuakse kasutades *JUnit* ja *Mockito* raamistikke, integratsioonitestid *JUnit* raamistikul ning kasutajaliidese testid kasutades *Selenium* raamistikku. Kvaliteedispetsialist viib testimist läbi arenduskeskkonnas. Tooteomanik kirjeldab testimiskeskkonna vajadused. Esitlusi viiakse läbi eraldi stabiilses keskkonnas, millele saab tooteomanik soovi korral ligi, samas keskkonnas teostatakse ka süsteemitestimist. Üldiselt kasutatakse testimiseks testandmeid ja hägustatud toodangu andmeid.

**Riskide maandamine.** Olemasoleva koodibaasiga konflikti sattumise riski vähendamiseks kasutatakse olemasolevate testide jooksutamist ja ümberkirjutamist ning lisaks luuakse puhta koodi eesmärgil uuele koodile testid. Lisaks kasutatakse pideva integratsiooni süsteemi (*Jenkins*). Regressioon automatiseeritakse maksimaalsel võimalikul määral, põhirõhuga ühik- ja komponenttestidel. Kvaliteedispetsialist loob ja haldab riskide nimekirja, milles on kirjeldatud põhilised riskikohad arendatava rakenduse tarnimisel. Regressiooniplaani täiendatakse ja kohandatakse iga sprindi jooksul.

**Toote riskid.** Toote riskidena kaardistati järgnevad: kasutajale kuvatakse ebakorrektne info, sisestatud andmete ebakorrektsus, lõppkasutaja sisestab ebakorrektse info, andmeid ei salvestata korrektselt, töövoogu mõjutavad teise projekti arendused, arendatav töövoog ei vasta tegelikule vajadustele, liidestused erinevate süsteemide vahel, spetsiifiline domeeniteadmus.

**Koolitamine.** Kvaliteedispetsialisti töö efektiivsemaks muutmiseks kasutatakse valdkonna spetsialistide abi koolitamisel.

**Testimise lõpetamise tingimused.** Koodibaas vastab seatud kvaliteedinõuetele ja valmimise kriteeriumitele. Testimise käigus ei tuvastata olulisi vigu ning varasemalt leitud vead on parandatud. Tarne puhul kehtib lisaks tingimus, et äripoole suitsutestimine on edukalt läbitud ning vajalik dokumentatsioon koolitamiseks loodud.

Esimesel sprindil töötati välja testistrateegia ning alates teises sprindist hakati seda kasutama. Testistrateegia kasutamist ei saa veel lugeda juurutatuks, kuid kõik eeldused selleks on loodud. Eespool kirjeldatuga kaeti parendamise skoobis püstitatud eesmärk ehk saavutati võtmeala „Testimise strateegia“ A-taseme kontrollpunktide täitmine.

### 7.1.2 Testimise protsessi haldus

Vastavalt peatükis 5.2 kirjeldatule oli antud võtmeala parendamise põhiliseks puudujäägiks testiplaan ning selle loomine. Pärast testistrateegia defineerimist asuti teisel sprindil looma testiplaani, mille sisendiks kasutati testistrateegiat, meeskondade varasemate retrospektiivide tulemusi ja parimaid praktikaid. Efektiivsema testimise planeerimiseks võeti kasutusele kompaktne ja lühike testiplaan, näide antud projekti testiplaanist on toodud lisas 3. Lisaks kinnitati sprindi testiplaan tooteomanikuga ning loodi selle alusel parem ülevaade, mis on testimise fookuses ning millistele riskidele tuleb rohkem tähelepanu pöörata. Tooteomanik tutvustas meeskonnale arenduse visiooni ja oodatavaid tarne kuupäevi. Igal sprindil kaasati tooteomanik testimisse ning tagati talle võimalus arenduse ülevaatamiseks kättesaadavas keskkonnas, mille alusel kiirendati tagasiside saamist äripoolelt. Ühtlasi kaasati äripoole lõppkasutajad ka esitlustele. Tehnilise keerukuse tõttu leiti aga kolmandal sprindil, et pikaajaliselt on keeruline tagada tooteomanikule stabiilset keskkonda ning selle tõttu loodi kokkulepe, et kvaliteedispetsialist loob vajaliku keskkonna ühe päevase ette teatamisega tooteomaniku poolt.

Kasutusele võetud testiplaani valmimiseks loodi peatükis 6.2.5 kirjeldatud testimiskvadrantide alusel kokkuleppeid. Kvadrant 1 (Q1) alusel tehti kokkulepe, et arendajad kirjutavad pidevalt ühik- ja komponententteste, eesmärgiga saavutada vähemalt 80% koodi katvuse tase. Kvaliteedispetsialist aitab kontrollida koodikvaliteedil, kasutades selleks staatilisi vahendeid ning leiab võimaluse testide katvuse analüüsiks integreeritud arendusvahendis. Lisaks analüüsib kvaliteedispetsialist kirjutatud testide sisu ning juhendab võimalike testide loomist. Muudetud koodi jooksutatakse võimalikult tihti pideva integratsiooni süsteemis (*Continuous Integration*), et lahendada võimalikult kiiresti tekkinud konflikte koodibaasis.

Esialgselt planeeris meeskond kasutusele võtta testipõhise arendusmetoodika, kuid esimese sprindi jooksul, kui seda rakendati, selgus, et selline lähenemine ei ole meeskonnale sobiv, kuna aeglustas oluliselt arendust ning tekitas ülemäära probleeme. Probleemide vältimiseks lepiti kokku, et kasutusele võetakse praktika, mis näeb ette testide loomise koheselt pärast koodi loomist, selle kaudu saavutati vajalik koodi kaetuse tase ning praktika kasutamist jätkati järgnevate sprintide jooksul.

Kvadrant 2 (Q2) alusel tehti kokkulepe, et kvaliteedispetsialist loob olemasolevale põhifunktsionaalsusele automatiseeritud testid, mis katavad esialgu ainult positiivseid

stsenaariumeid ning teised juhtumid kaetakse võimalusel madalamal tasemel integratsioonitestidega. Valmimise kriteeriumis kirjeldati ühik- ja komponenttestide läbimine, läbivaatuste läbimine ja testimise edukas läbimine. Lisaks loob meeskond kasutajaliidese prototüüpe ning verifitseerib neid tooteomaniku ja teiste huvipooltega, et katta vähemalt minimaalne kasutajamugavuse vajadus. Lisaks rakendas kvaliteedispetsialist planeerimiskoosolekul vastuvõtukriteeriumite täpsustamiseks *INVEST* kriteeriumit ning küsis selgitusi kasutajalugude kohta näidete põhjal. *INVEST* kriteeriumit rakendades jagati mitmed kasutajalood väiksemateks tükkideks ning selgitati välja puudujääke, mille alusel kasutajalugusid koheselt parendati, kuid üldine kasutajalugude kirjutamise tase oli suhteliselt kõrge nii et oli keeruline veenduda antud praktika kasulikkuses, kuid sellegipoolest jätkatakse selle praktika katsetamist.

Kolmanda kvadrandi (Q3) täitmiseks kasutatakse lähtuvalt testistrateegiast uurivat testimist, mida viiakse läbi sessioonipõhise testimise abil. Neljanda kvadrandi (Q4) testid viiakse läbi vajadusel sprindi jooksul või vahetult enne tarnet. Meeskond otsustas, et antud hetkel on teised kvadrandid olulisemad ning lisatestid tuuakse vajadusel sisse tarne testiplaanis.

Eelpool tooduga täideti vajalikud „A“-klastri kontrollpunktid, kuid autori poolt antud soovitustest jäi esialgu välja standardi ISO-9216 kvaliteediatribuutide analüüs ja kaasamine, mida planeeritakse hiljem uuesti kaaluda.

### 7.1.3 Testimise kaasatus protsessis

Vastavalt peatükis 5.2 kirjeldatule oli antud võtmeala põhiline puudujääk testimise eesmärgi, skoobi ja riskide defineerimine. Samuti puudus struktureeritud lähenemine testimisele enne testide jooksutamise etappi. Nende puudujääkide kõrvaldamiseks võeti edukalt kasutusele testistrateegia ja –plaanide kirjeldamine, mis võimaldas eesmärgi, skoobi ja riskid muute projekti põhilistele osapooltele läbipaistvaks ja arusaadavaks. Lisaks alustati *INVEST* kriteeriumi rakendamist, nagu kirjeldati eelmises peatükis 6.1.2.

Tarne testiplaani esialgselt veel ei loodud, kuid sellele on pandud tugev alus strateegia ja sprindi testiplaanide näol. Lisaks haldab kvaliteedispetsialist tarne testiplaani loomise jaoks riskide ja regressiooni nimekirja, mis lihtsustab vajalikul hetkel testiplaani loomist. Esialgselt kirjeldati testiplaan dokumendi vormis, kuid järgnevates sprintides kaalutakse mõttekaartide kasutamist, et testiplaani oleks kergem visualiseerida ning selle kaudu saab selle muuta kõikidele osapooltele nähtavaks.

Näidetel põhinevate testide praktika „elava dokumentatsiooni“ loomisega tekkis aga probleeme, kuna meeskonna esialgsel hinnangul selgus, et arendatav süsteem ei sobi keeruka ülesehituse tõttu antud praktika rakendamiseks ning selliste testide loomine tekitas meeskonnas selget rahulolematust ja seetõttu selle soovituse täitmist päevakorda ei võetud.

Käesoleva töö mahus kirjeldatud praktikaid kombineeriti olemasoleva protsessiga ning selle alusel loodi Ignite OÜ testimisprotsessi kirjeldus, mis on detailsemalt kirjeldatud peatükis 9.

Sisseviidud muudatuste abil täideti võtmeala „Testimise kaasatus protsessis“ „A“-klastri kontrollpunktid ning peaaegu kõik autori poolt tehtud soovitused, välja arvatud „elava dokumentatsiooni“ soovitus.

### 7.1.4 Testidisain

Sprindi algul töösse võetud kasutajalugudele loodi vastuvõtukriteeriumitest ja testistrateegiast lähtuvalt testilood ning võeti kasutusele lisas 5 kirjeldatud testiloo mall. Testilugusid disainis kvaliteedispetsialist lähtudes oma varasemast kogemusest, olemasoleva funktsionaalsuse kirjeldusest ja riskidest. Lisaks prooviti testiideede loomisel kasutada peatükis 6.3.1.1 soovitatud heuristikaid, millele andis kvaliteedispetsialist positiivse tagasiside ning kinnitas nende kasulikkust. Samas viitas kvaliteedispetsialist, et sooviks struktureeritumalt testilugusid luua, et saavutada paremat katvust ning selleks tuleb kohendada soovitatud malli, et oleks võimalik tehnikate kirjeldamine ja järelanalüüs. Uuriva testimise haldamiseks katsetati sessioonipõhist testimist, mis andis positiivse tulemuse struktureeritud raportite näol, mille alusel oli kerge tooteomanikule anda ülevaadet tehtud tööst.

Kasutajaliidese kaudu automatiseerimiseni läbitud sprintide jooksul ei jõutud, seega ei jõutud katsetada automatiseerimiseks tehtud soovitusi. Samas ühik- ja komponenttestide loomisel lähtusid arendajad automaattestide heast tavast ning loodi kokkulepe, et ka kasutajaliidese testid peavad vastama sellele. Esialgselt soovis meeskond rakendada testidel põhinevat arendust (*Test-driven development*), kuid seda ei suudetud efektiivselt kasutusele võtta, seega lepiti kokku lähenemises, kus luuakse testid koheselt pärast koodi loomist ning see tõestas oma kasulikkust kas disaini vaatenurgast, kuna teste luues avastati koodi disaini kõrvalekaldeid.

Eelpool tooduga täideti kõik võtmeala „Testidisain“ „A“-klastri kontrollpunktid.

### 7.1.5 Praktikate rakendamise tulem

Tabelis 8 on toodud parendamiseks valitud meeskonna testimisprotsessi küpsusmaatriks pärast parimate soovituste ja parimate praktikate rakendamist. Küpsusmaatriksi kirjeldamiseks hindas meeskonna kvaliteedispetsialist muutunud seisu.

**Tabel 8. Testimisprotsessi küpsusmaatriks pärast praktikate rakendamist**

| | | Võtmeala | K | N | M | Baas | Kontrollitud | | | | Efektiivne | | | | Optimeeritud | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KP | Kliendi pühendumine | | x | | + | A | B | B | C | F | H | H | | K | M | M |
| 2 | KP | Testimise kaasatus protsessis | x | | | + | A | A | C | D | G | G | H | | K | K | |
| 3 | KP | Testimise strateegia | x | | | + | A | A | A | D | E | E | G | | J | K | |
| 4 | KP | Testimise organisatsioon | | | x | + | B | E | E | I | J | J | K | K | L | M | M |
| 5 | KP | Suhtlemine | | x | | + | B | C | C | D | F | F | J | | M | M | |
| 6 | KP | Raporteerimine | | | x | + | C | D | F | | G | K | K | | L | L | |
| 7 | TH | Testimise protsessi haldus | x | | | + | A | A | A | A | H | J | K | | M | L | |
| 8 | TH | Hindamine ja planeerimine | | x | | + | B | B | C | C | G | H | I | I | K | L | L |
| 9 | TH | Meetrika | | | x | + | D | D | E | | F | G | G | H | L | L | |
| 10 | TH | Vigade haldus | | | x | + | B | C | C | F | H | G | I | K | L | M | M |
| 11 | TH | Testimisvahendite haldus | | x | | + | B | B | D | F | I | I | J | | L | L | L |
| 12 | KO | Metoodika | | x | | + | C | D | E | | F | H | J | J | M | M | |
| 13 | KO | Kvaliteedispetsialisti oskused | x | | | + | C | C | D | D | E | E | G | G | I | I | K |
| 14 | KO | Testidisain | x | | | + | A | A | C | | D | E | E | F | I | I | J |
| 15 | KO | Testimisvahendid | | x | | + | E | E | E | | F | G | G | I | L | M | M |
| 16 | KO | Testkeskkond | | | x | + | D | F | F | G | H | H | J | J | L | M | M |

Nagu tabelist näha, õnnestus „A“-taseme klastri kontrollpunktide täitmine. Selle meeskonna puhul on „B“-taseme klastris olevad kontrollpunktid juba täidetud, seega tuleb edasi liikuda klastriga „C“, ehk võtta vaatluse alla võtmealad „Metoodika“ ja „Kvaliteedispetsialisti oskused“. Järgmiste võtmealade parendamisele on hea aluse loonud ka peatükis 9 kirjeldatud testimisprotsess ja seal hulgas olev kvaliteedispetsialistide oskuste kirjeldus.

Meeskond andis retrospektiivi jooksul tagasisidet, et kasutatud protsess oli väga intensiivne ja seetõttu väsitav. Lisaks sooviti rakendatud praktikate katsetamiseks igat praktikat kasutada vähemalt 2-3 sprinti, et oleks aega nendega harjuda ja veenduda nende sobivuses. Sprint 4 retrospektiivi koosolekul otsustati rakendada parendamise samme veidi rahulikumas tempos ja pikema aja jooksul. Seega edaspidi on planeeritud 1 võtmeala parendamine 2 sprindi jooksul ning vajadusel pikemalt.

# 8. Ignite OÜ testimisprotsess

Järgnevalt on kirjeldatud struktureerimise metoodika rakendamise käigus loodud testimisprotsess ning nende täitmiseks vajalikud oskused.

Tarkvara testimine on protsess, mille eesmärk on avastada puudused tarkvaras ja võrrelda arvutiprogrammi loodud tulemusi tegelike oodatud tulemustega. Seevastu kvaliteedi tagamise (*quality assurance*) eesmärk on korraldada arendustegevust ja selle põhimõtteid, et vältida juba alguses defektide tekkimist. Efektiivse testimise tagamiseks peab kvaliteedispetsialist omama ülevaadet testimistegevustest ja –tulemitest, mis saavutati eelnevalt, mis on hetkel käsil ja mida hakatakse tulevikus läbi viima.

Käesoleva töö mahus kirjeldatud teooria kinnitab, et kvaliteedispetsialisti võib jagada üldpildis kolmeks:

1. **Testimisepraktikate kasutus.** Kvaliteedispetsialist vastutab parimate testimise praktikate kasutamise eest (sh enda ja teiste koolitamine);

2. **Sprindi sisesed testimistegevused.** Kvaliteedispetsialist vastutab kliendi ja meeskonna testimisprotsessi koordineerimise eest;

3. **Testimine enne tarnet/ toodangusse laskmist.** Kvaliteetse tarne jaoks peab olema tagatud hea ülevaade tehtud tööst. Selles etapis viiakse läbi eritüübilisi testimisi, mis vajavad toodangulaadset keskkonda. Põhifookuses peaksid olema peatükis 6.2.5 kirjeldatud testimise kvadrandid 3 ja 4. Toodangusse lastavat tarkvara tuleb testida lähtuvalt toote testistrateegiast ning selle alusel koostada konkreetse tarne testiplaan. Kõik olulised riskid peavad olema maandatud.

Käesoleva magistritöö mahus kirjeldati erinevaid kvaliteedispetsialisti kohustusi, tehnikaid ja testimisprotsessi. Järgnevalt on kirjeldatud läbi viidud analüüsi ja olemasoleva protsessi kombineerimisel saadud Ignite OÜ kvaliteedispetsialistide kohustused ja sprindi testimise protsess.

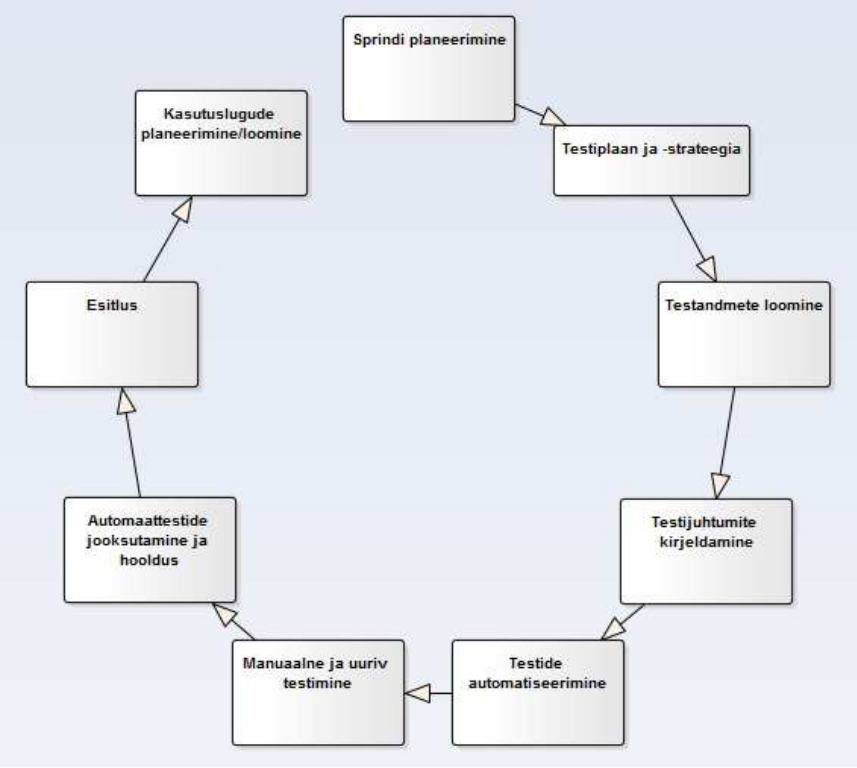
Igas sprindis läbi viidavad tegevused peaksid sisaldama:

1. Koostööd tooteomanikuga, mille käigus analüüsitakse ja viimistletakse järgnevatesse sprintidesse võetavaid kasutajalugusid.

2. Kasutajalugude testimine vastavalt vastuvõtukriteeriumitele ning lisaks valitud testimistehnikate kasutamine;

3. Manuaalset ja automatiseeritud testimist. Ideaalselt lõpetatakse kõik testimistegevused (käsitsi- ja automaattestimine) ühe sprindi jooksul. Kuigi võib juhtuda ka nii, et on vajalik lisaks lõpetada eelneva sprindi töid;

Joonisel 8 on toodud sprindi testimise protsess. Kirjeldatud tegevusi viiakse tavaliselt läbi järjekorras, kuid see ei ole kohustuslik ning nende tegevuste läbiviimist kohandab iga meeskonna kvaliteedispetsialist vastavalt vajadustele.




**Joonis 8. Sprindi testimistegevused**

**Sprindi planeerimine.** Kvaliteedispetsialist saavutab üldise arusaama planeeritavast kasutajaloost ning aitab kaasa vastuvõtukriteeriumite loomisel (küsib õigeid küsimusi, täpsustab näiteid, kaardistab eesmärke). Kvaliteedispetsialist hindab sprinti võetavate kasutajalugude testimise mahtu ja keerukust. Hea on ka praktika, kui kvaliteedispetsialist suudab sprinti võetavaid kasutajalugusid eelnevalt analüüsida, sel juhul saab ta paremini

planeerida testimisstrateegiat. Planeerimisel või hiljemalt sprindi esimeste päevade jooksul luuakse riskianalüüs.

**Testiplaan- ja strateegia**. Testistrateegia loomine, et kirjeldada arenduses kasutusele võetav üldine lähenemine testimisele sealhulgas testimistehnikate valik ja põhjalikkus. Tarne ja sprindi testiplaani loomine ning täpsustamine, mille kaudu kirjeldatakse testimise skoop, prioriteedid ja põhilised riskid. Lisaks antakse ülevaade milliseid vahendeid ja andmeid kasutatakse, kes on lõppkasutajad ning mida on vajalik dokumenteerida (ainult vajalikud osad). Analüüsitakse kõiki agiilseid testimise kvadrante ning otsustatakse, kes on vastutavad teatud testimistasemete läbimise eest.

**Testandmete loomine.** Kvaliteedispetsialist toetab vajalike testandmete loomist, mis on äärmiselt vajalik eduka arenduse jaoks. Andmeid luuakse kas manuaalselt või automaatselt, kasutades erinevatest allikatest pärit andmeid (nt hägustatud testandmed, toodangu andmed). Kvaliteedispetsialist peab looma erinevaid andmeid ning katma nii arendajate kui testimise vajadusi. Antud punkti alla kuulub ka näiteks eelmise iteratsiooni andmete kustutamine või eemaldamine süsteemist.

**Testlugude kirjeldamine**. Kasutades testidisaini tehnikaid ja praktikute poolt loodud heuristikat. Luues struktureeritud kujul testilugusid, näiteks lähtudes lisas 5 olevast mallist.

**Testide automatiseerimine**. Meeskonnaga koos tuleb määrata, millisel tasandil kvaliteedispetsialist automatiseerida saab ning millises mahus seda on tarvilik teha. Automatiseerimisel jälgitakse investeeringutasuvust. Leitakse õiged vahendid ja tööriistad testimise korraldamiseks. Testide loomisel peab jälgima automatiseerimise „head tava“ ning testikoodi tuleb käsitleda samasuguse hoolega nagu toote koodi. Tuleb kasutada objekt-orienteeritud koodi, mitte duplitseerida koodi ning luua korrektne struktuur. Selle punkti alla kuulub ka olemasolevate testide jooksutamine ja hooldamine ning testiraportite jälgimine ja vigade raporteerimine. Automatiseeritud testid peaksid jooksma vähemalt mõned korrad sprindi jooksul, eelistatavalt pideva integratsiooni süsteemi kaudu. Lisaks peab toetama kvaliteedispetsialist arendajate poolt teostatavat automatiseeritud testimist (ühik- ja komponenttestimine).

**Manuaalne ja uuriv testimine**. Testimist viiakse läbi arenduskeskkonnas ning testimise põhifookuses on kasutajaloo vastuvõtukriteeriumid ja loodud testilood. Kvaliteedispetsialist

peab tegema koostööd tooteomanikuga ning võimalusel testima ka eraldi keskkonnas, kuhu on paigaldatud lõplik versioon tarkvarast.

**Esitlus (*Demo*)**. Kvaliteedispetsialist valmistab ette ja viib läbi esitluse, mille käigus tutvustatakse äripoolele valminud kasutajalugusid. Esitlusel presenteeritakse ainult lõpetatud kasutajalugusid, mis vastavad täielikult Valmimise kriteeriumitele. Esitlust viiakse läbi lõplikul tarkvara versioonil ning stabiilses keskkonnas.

**Järgmise sprindi tööde analüüs ja planeerimine**. Antud tegevuse hulka kuulub kasutajalugude täpsustamine ja hindamine. Täpsustamisel kasutatakse *INVEST* kriteeriumit ja näiteid ning tööde mahu hindamisel võetakse arvesse testimise keerukust ja võimalikke riske.

Lisaks eelnevalt nimetatud tegevustele tuleb paralleelselt tegeleda vigade haldusega ja luua läbipaistvus kasutajalugude ja läbiviidud testide vahel.

**Vigade haldamine.** Vigade haldussüsteemid ei oma agiilses arenduses olulist rolli. Sprindi jooksul kaetakse tekkinud vead automaatsete testimisvahenditega ning meeskonnaliikmete tiheda koostöö tõttu ollakse enamasti teadlikud vigadest ja need parandatakse koheselt. Üldiselt kasutatakse sprindi jooksul suuliselt ja näidetel põhinevat lahendust. Samas ei õnnestu alati kõiki vigu avastada sprindi jooksul ning sellisel juhul on vaja vigu hallata. Meeskond peab otsustama, kuidas käsitletakse järgnevaid juhtumeid:

1. Viga avastatakse hilisema manuaaltestimise käigus (Vaja teada, kuidas korrata);
2. Vea parandamiseks on vaja lisaressursse või uurimust (Vea raporteerimine tagab, et kõik vajalikud osalised on sellest teadlikud);
3. Viga ei parandata sprindi jooksul (Vaja registreerida viga, et seda ei unustataks).

**Läbipaistvus.** Kvaliteedispetsialist on vastutav selle eest, et kogu meeskond ja tooteomanik oleks teadlik, mis seisus on arenduses oleva toote testimine, selleks on võimalik kasutada erinevaid mõõdikuid, mida on hea kujutada nii virtuaalsetel kui füüsilistel tahvlitel:

1. Kasutajalugude arv, mis on testimiseks valmis;
2. Kasutajalugude arv, mis on testimises;
3. Parandamist ootavate vigade arv;

4. Ületestimist ootavate vigade arv;
5. Automatiseerimist ootavate kasutajalugude arv.

Kõige olulisemad kvaliteedispetsialisti poolt loodavad tehised on järgnevad dokumendid (lühikesed ja ülevaatlikud):

1. Testistrateegia – ja plaan (sprint, tarne);
    a. Tulemused/raportid;
2. Kasutajalugude vastuvõtukriteeriumid (kas koostöös tooteomanikuga või iseseisvalt);
3. Riskide nimekiri;
4. Ülevaade süsteemist;
    a. Komponentide omavahelised seosed;
    b. Andmebaasi diagrammid;
    c. Vajalik dokumentatsioon ja juhendid;

## 8.1 Kvaliteedispetsialisti oskused

Kvaliteedispetsialist on täisväärtuslik liige arendusmeeskonnas ja töötab tihedas koostöös arendajatega, seetõttu on tekkinud selge vajadus, et ta mõistaks arendajate tööd ja terminoloogiat ning omaks üldisi teadmisi IT valdkonnas. Kuna Scrum tarkvaraarendusmetoodikat jälgiv meeskond on multifunktsionaalne, siis igal liikmel peaks olema vähemalt ühes valdkonnas sügavad teadmised ning teistes valdkondades üldised teadmised. Seetõttu on vaja, et ka kvaliteedispetsialistidel kujuneks „T-kujuline“ oskuste kogum, mis lisaks testimisteadmusele sisaldaks ka üldist arusaama järgnevatest teemadest: [9]

1. Süsteemiarhitektuur;
2. Üldised programmeerimise ja disaini põhitõed;
3. Andmebaaside baasteadmised;
4. Integreeritud arendusvahendid ja –keskkonnad;

5. Pideva tarnimise keskkonnad;

6. Rakenduste arendus- ja paigaldusprotsess.

Selleks, et toetada arendajaid kiires ja pidevalt muutuvas keskkonnas peab kvaliteedispetsialist olema iseseisev ning suutma viia läbi järgmiseid tegevusi:

1. Rakenduste ja keskkondade (sh mälukasutus) protsesside jälgimine;

2. Logifailide lugemine;

3. Testkeskkondade haldamine;

4. Andmebaaside kasutamine.

# 9. Testimisprotsessi struktureerimise metoodika hinnang

Käesoleva töö mahus läbiti peatükis 3 kirjeldatud Demingi ringi etapid:

**Planeerimine**. Peatükis 3.1 teostati testimise parendamise mudeli valik ning peatükis 4 kirjeldati ettevõtte probleeme ning eesmärke.

**Teostamine.** Peatükis 5 teostati olemasoleva testiprotsessi analüüs ning koostati parendamise skoop. Kuuendas peatükis analüüsiti maailmas kasutatavaid parimaid praktikaid ning nende alusel loodi konkreetsed soovitusettepanekud, mille alusel viidi sisse vastavad parendused protsessis.

**Kontrollimine.** Peatükis 7 kirjeldati parimate praktikate ja autori antud soovituste rakendamist ning lõpptulem kirjeldati uuel küpsusmaatriksil.

**Korrigeerimine.** Peatükis 7.1.5 kirjeldati parendamise käigus tekkinud metoodika muutmise ja korrigeerimise ideed.

Antud metoodikat rakendati reaalse projekti näitel ning saavutati positiivseid tulemusi. Kasutatud metoodika negatiivseks küljeks võib lugeda seda, et võtmealade parendamiseks on tarvilik viia läbi mahukas analüüs olemasolevatest praktikatest, mis teeb analüüsi osa aeglaseks ja ajamahukaks, samas metoodika kasuks räägivad saavutatud positiivsed tulemused.

Konsulteerides ettevõtte teiste meeskondade kvaliteedispetsialistidega, hinnati loodud metoodikat ja saavutatud tulemit positiivselt. Struktureerimise metoodika hinnati heaks ning piisavalt paindlikuks, et rakendada ka teiste meeskondade töö korraldamises. Selleks, et tõeliselt veenduda antud metoodika toimivuses tuleb parendamise protsessi korrata seni, kuni kõik meeskonnad on saavutanud peatükis 4 kirjeldatud TPI *Next* mudeli „Kontrollitud“ taseme täitmise eesmärgi.

Seega on autori hinnangul antud metoodika sobiv testimisprotsessi struktureerimiseks Scrum tarkvaraarendusmetoodikat kasutavas ettevõttes.

# 10. Kokkuvõte

Magistritöö eesmärgiks oli välja pakkuda metoodika, kuidas struktureerida testimisprotsessi Scrum arendusmetoodikat kasutavas ettevõttes ning selle metoodika abil identifitseerida testimisprotsessi probleemkohad ning luua neile parendusettepanekud.

Selleks, et eesmärke saavutada kasutati testimisprotsessi struktureerimiseks testimise parendamise mudelit TPI *Next.* Mudeli rakendamise käigus analüüsiti ettevõtte vajadusi ja eesmärke ning olemasolevat protsessi. Olemasoleva protsessi analüüsi alusel kaardistati prioritiseeritud probleemkohad. Kõige olulisemateks osutusid: testimise kaasatus protsessis, testidisain, testimise strateegia ja testimisprotsessi haldamine.

Protsessi reaalseks parendamiseks oli aga tarvilik viia läbi analüüs, et selgitada välja konkreetsed praktikad, mida kasutatakse tänapäeval antud probleemide lahendamiseks. Selle eesmärgiga töötas autor läbi palju agiilset testimist käsitlevaid materjale, et leida tunnustatud arusaama, millised on antud valdkonnas levinud praktikad ja kuidas neid rakendatakse ning koostas nende alusel parendusettepanekud.

Selleks, et hinnata parendusettepanekute toimivust katsetati neid ühe meeskonna töö näitel ning saavutati positiivne tulem. Metoodika alusel parendatud meeskonna testimisprotsess paranes võrreldes varasemaga ning seega tõestati, et selle metoodika alusel on võimalik parendada ettevõtte testimisprotsessi, kuid lõpliku veendumuse saavutamiseks on tarvilik sama protsessi läbida kõikide meeskondadega.

Kuna Scrum testimisprotsessi struktureerimisega tegelevaid juhtumianalüüse ei ole avalikult kätte saada, siis ei saanud läbi viia metoodika võrdlust teiste analoogsete tööde tulemitega. Metoodika rakendamise positiivse tulemuse alusel võib järeldada, et seda metoodikat saab tulemuslikult kasutada Scrum testimisprotsessi parendamiseks. Lisaks kirjeldati läbitud analüüsi ja olemasoleva protsessi kaardistuse alusel Ignite OÜ testimisprotsess, mida saab kasutada aluseks juhiste andmiseks töötajatele.

Kuid välja pakutud metoodika kasutamine testimisprotsessi parendamiseks ei ole ühekordne tegevus. Parendamistsükleid tuleb jätkata seni, kuni soovitud tulem on saavutatud. Antud lõputööst tuleb edasi liikuda ettevõtte teiste meeskondade protsesside parendamisega, mille

eest vastutab igas meeskonnas olev kvaliteedispetsialist. Sellisel juhul on võimalik soovitud tulemi saavutamine.

Kirjeldatud tulemitega täideti lõputööle püstitatud eesmärgid, kuna leiti metoodika, kuidas on hea struktureerida testimisprotsessi ning identifitseeriti probleemkohad, nende prioriteedid ja kaardistati võimalikud lahendused. Lisaks kontrolliti metoodika toimivust ühe meeskonna tööd parendades ning selle alusel kirjeldati ettevõtte testimisprotsess. Kuna vaadeldud ettevõttes on kasutusel Scrum muutmata kujul, võib eeldada, et ka sama metoodika rakendamine teistes sarnastes ettevõtetes saavutab positiivse tulemi.

# Summary

The main goal of this thesis was to find a solution how to structure the test process of a company who has implemented Scrum software development methodology. The aim on finding the solution was to identify the weaknesses of the existing testing process and to create improvement suggestions how to eliminate them.

In order to assess the current state of the testing process an evaluation model had to be chosen. The model was chosen in accordance with the company goals and Scrum methodology. TPI Next was chosen to create an analysis of the testing process. During the analysis the goals and the problem areas of the testing were described and documented. Using the maturity matrix with mapping of the goals of the company created the scope for improvement. The chosen scope included the degree of involvement of testing, test strategy, test process management and test design.

Based on the scope the author conducted wide search of existing literature and best practices and created description of possible solutions and improvement suggestions according to the findings. To validate the created improvement suggestions and methodology a case study was conducted. As a result the testing process improved and used methodology gave positive results.

Based on the conducted research and the analysis of existing process a description of Ignite OÜ testing process was created. Also the list of necessary skills for quality specialist was created. Those results can be used to train existing and new employees.

In order to verify fully the proposed methodology it has to be implemented by all the teams in Ignite OÜ. The usage of the proposed methodology is not a one time act. The improvement cycle has to be repeated until the sufficient level of testing process and company goals are reached.

In summary, all this thesis targets were reached. Master thesis could be used for improving testing process in software companies which have already introduced Scrum methodology.

# Kasutatud kirjandus


1. James, M. Scrum reference card. [WWW] http://scrumreferencecard.com/scrum-reference-card/ (17.03.2015)

2. Gil-Santamaria, M. Agile and Scrum Methodologies from a Testing/QA Perspective. [WWW] http://www.stickyminds.com/article/agile-and-scrum-methodologies-testingqa-perspective?page=0%2C2 (17.03.2015)

3. Linz, T. (2014). Testing in Scrum - A Guide for Software Quality Assurance in the Agile World. Rocky Nook Computing.

4. Mengerink, J., Knol, E. Test improvement for Agile. [WWW] http://www.polteq.com/wp-content/uploads/2012/10/20121018-TIfA-presentatie-NNOT.pdf (21.03.2015)

5. Ambler, S. Agile Testing and Quality Strategies: Discipline Over Rhetoric. [WWW] http://www.ambysoft.com/essays/agileTesting.html#AgileSoftwareDevelopment (21.03.2015)

6. Kumar, Pavan. Test Process Improvement – Evalutation of Available Models.[WWW] http://maveric-systems.com/pdf/whitepapers/White_paper_Vol%208_TPI.pdf (31.03.2015)

7. Veenendaal, E. Test Process Improvement and Agile: Friends or Foes? [WWW] http://www.erikvanveenendaal.nl/NL/files/testingexperience27_09_14_van_Veenendaal.pdf (15.03.2015)

8. Hansen, E., Jaanisk, M. Kvaliteedijuhtimine. [WWW] https://www.nooruse.ee/e-ope/opiobjektid/kvaliteediaine/kvaliteedijuhtimisssteemi_kavandamine_ja_rakendamine.html [WWW] (01.04. 2015)

9. Gregory, J., Crispin, L. (2015). More Agile Testing. Addison Wesley.

10. Miks Scrum? [WWW] http://www. scrum.ee (27.04.2015)

11. International Software Testing Qualifications Board. Agile Tester Extension Syllabus. [WWW] http://www.istqb.org/downloads/syllabi/agile-tester-extension-syllabus.html (27.04.2015)

12. Sogeti TPI Next : Sogeti, 2009.

13. Axis Agile, 'Chief ScrumMasters' and ScrumMasters. http://www.axisagile.com.au/blog/scrum-roles/chief-scrummasters-and-scrummasters/ (27.04.2015)

14. Ron Swinkels „A comparison of TMM and other Test Process Improvement" 17-11-00 (27.04.2015)

15. Jõgi, E. Agiilne tarkvaraarendus kui tarkvara kvaliteedi tõstmise vahend. [WWW] http://www.slideserve.com/valeria/erik-j-c3-b5gi-0berik-jogi-40swedbank-ee (15.03.2015)

16. Agile Methodology. [WWW] http://agilemethodology.org/ (03.04.2015)

17. Agiilse tarkvaraarenduse manifest. [WWW] http://agilemanifesto.org/iso/et/ (03.04.2015)

18. International Software Testing Qualifications Board. Foundation Level Syllabus [WWW] http://www.istqb.org/downloads/syllabi/foundation-level-syllabus.html (03.04.2015)

19. Tepandi, J. Tarkvara kvaliteet ja standardid (IDX5721, IDX5722) [WWW] http://deepthought.ttu.ee/users/tepandi/pdf/tks-loeng.pdf (03.04.2015)

20. Expedith, M. Agile Testing: Key Points for Unlearning [WWW] https://www.scrumalliance.org/community/articles/2012/january/agile-testing-key-points-for-unlearning (03.04.2015)

21. What is V-model - advantages, disadvantages and when to use it? [WWW] http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/ (03.04.2015)

22. Wake, B. Invest in good stories, and smart tasks. [WWW] http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ (19.04.2015)

23. Cohn, M. (2004). User stories applied. Addison-Wesley.

24. Schwaber, K. Beedle, M. Agile Software Development with Scrum. (2001) Prentice Hall.

25. Hendrickson, E. Test Heuristics Cheat Sheet. [WWW] http://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf (02.05.2015)

26. VersionOne 2013. 7th Annual State of Agile Development Survey. http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf [WWW] (02.05.2015)

27. Adzic, G. (2011). Specification by Example. Manning

28. Hendrickson, E. Driving Development with Tests: ATDD and TDD. [WWW] http://testobsessed.com/wp-content/uploads/2011/04/atddexample.pdf (02.05.2015)

29. Crispin, L. Gregory, J. (2009) Agile Testing. Addison Wesley.

30. Crispin, L. An Overview of Agile Testing. http://lisacrispin.com/downloads/AgileTestingOverview.pdf [WWW] (03.05.2015)

31. Hunter, M. You are not done yet checklist. http://www.thebraidytester.com/downloads/YouAreNotDoneYet.pdf [WWW] (03.05.2015)

# Lisa 1

TPI *Next* võtmealade kirjeldus.

**Testimise organisatsioon.** Antud võtmeala kirjeldab tervet ettevõtet ning kuidas see töötab, kuidas juhitakse testimisprotsesse, hallatakse vastutusi ja kohustusi. Üks testimise organisatsiooni eesmärke on koguda, korrastada ja jagada kogemusi ning oskuseid. See võtmeala hindab olemasolevate teadmiste korrastatust ja selle jagamist, mis on oluline aspekt struktureeritud protsessi saavutamisel. Teised võtmealad on tihedalt seotud just selle võtmealaga.

**Suhtlemine.** Selleks, et meeskond saaks efektiivsel töötada, peab liikmete vahel olema selge ja üheselt mõistetav kommunikatsioon. Seega on tarvilik, et oleks loodud eeldused sujuvaks suhtlemiseks, näiteks valitud sobivad kanalid (otsesuhtlus, Skype jne). Samuti on oluline, et näiteks koosolekud on efektiivsed ja hästi juhitud. Vähene või puudulik suhtlemine tekitab agiilses keskkonnas kiiresti probleeme.

**Raporteerimine.** Efektiivne raporteerimine toetab klienti riskide hindamisel, mis kaasnevad arendustöödega. Head raportid annavad infot kas testimise eesmärgid said täidetud ning millised riskid eksisteerivad. Samuti annavad need aimdust kui palju ressurssi on vaja testimise eesmärkide täitmiseks ning milliseid ümberkorraldusi on vajalik teha. Agiilses keskkonnas võivad testiraportid olla kerge-kaalulised, näiteks mõtte-kaardid.

**Testimise protsessi haldus.** Sama oluline kui testimise eesmärkide kokkuleppimine ja planeerimine, on testimise protsessi määratlemine, mis kirjeldab kuidas nende eesmärkideni jõutakse. Protsessi kirjeldamine võimaldab varakult mõista, millised on vajalikud osapooled, keda peab testimisse kaasama ning millises mahus ja skoobis seda läbi viiakse.

**Hindamine ja planeerimine.** Võtmeala vaatleb milliseid tehnikaid ja viise kasutatakse arendusprojektis testimistegevuste planeerimiseks ja hindamiseks. See võtmeala on tihedalt seotud testimise protsessi haldamisega, kuna loob aluse testimistegevuste mõõtmiseks. Usaldusväärne hindamine ja planeerimine võimaldab kasutada efektiivselt olemasolevat ressurssi ning vajadusel kaasata vajalikke inimesi juurde.

**Meetrika.** Meetrika loob aluse kvantitatiivsele andmestikule, mis on vajalik testimise hindamiseks. Seda kastutatakse testimise haldamiseks ja vajalike valikute tegemiseks, näiteks lisatestimise vajaduse hindamiseks või testimisprotsessi parendamiseks. Meetrika kogumine võib osutuda keeruliseks ülesandeks, seega on vajalik eesmärke silmas pidada ning leida lihtsaim ja kiireim viis selle saavutamiseks. Samuti peaksid tulemid olema lihtsalt loetavad ning pidevalt kasutatavad ja uuendatavad.

**Vigade haldus.** Agiilses arenduskeskkonnas peab meeskond leidma ühiselt vastuvõetava viisi, kuidas on kõige efektiivsem ja kiirem vigasid hallata. Kuna vead tekivad arendusprotsessis erinevatel etappidel ning lisaks omavad erinevaid olulisustasemeid, siis on vajalik, et vigu hallatakse efektiivselt ja struktureeritult, et tagada vigade parandamist.

**Metoodika.** Testimisprotsessi luues tuleb kirja panna testimismeetodika, mis kirjeldab kuidas testimist läbi viiakse ning jõutakse seatud eesmärgini teatud kvaliteediga. Lisaks peab metoodika kirjeldama juhiseid, malle, näiteid ja vahendite soovitusi, mis omakorda kiirendavad testimist.

Ainuüksi metoodika kirjeldamine ei ole aga piisav, seda tuleb projektides kasutada ning vajadusel täiendada. Levinud on olemasolevate metoodikate ja parimate praktikate kombineerimine ettevõtte vajadustele vastavaks protsessiks. Metoodika peaks sisaldama testiplaani malli ja ka tehnikaid (nt testidisaini tehnikad).

**Testidisain.** Hea testidisain võimaldab testlugusid ja –ideid taaskasutada, mis loovad nendest olulise väärtuse. Halvasti kirjeldatud testlugusid kasutatakse suure tõenäosusega vaid mõned korrad ning tekitavad selle kaudu ülemäärast kulu. Taaskasutatavate testide loomise põhialuseks ja eelduseks on head testidisaini tehnikate tundmise oskused.

**Testimisvahendid.** Testimisvahendite tundmine ja professionaalne kasutamine võimaldavad oluliselt testimist kiirendada ning vahel on spetsiifiliste vahendite kasutamine ainuvõimalik. Kasutada on võimalik palju erinevaid testimisvahendeid – planeerimiseks, haldamiseks, disainimiseks, automatiseerimiseks jne.

**Testkeskkond.** Stabiilne testkeskkond on vajalik selleks, et võimaldada pidevad robustse, stabiilse ja töötava tarkvara tarned. Kuna rakenduse töötamist ja selle ärilist väärtust tuleb testida väga lühikese aja jooksul ning probleemid testkeskkonnaga tekitavad protsessis kiiresti suuri probleeme ja takistusi.

# Lisa 2

Järgnevalt on kirjeldatud, kuidas TPI *Next* mudeli võtmealad on seotud agiilsete põhimõtetega. Iga põhimõtte kohta on kirjeldatud sellega seotud võtmealad ning kirjeldatud seose vajalikkust testimise vaatenurgast.

**Tabel 9. Agiilsete põhimõtete ja võtmealade seosed**

| Agiilne põhimõte | Võtmeala | Seose selgitus |
|---|---|---|
| Kõige olulisem on tagada kliendi rahulolu, tarnides talle vajalikku tarkvara võimalikult kiiresti ja tihti. | Kliendi pühendumine;<br><br>Testimise kaasatus protsessis;<br><br>Suhtlemine;<br><br>Testimise strateegia. | Kvaliteetse lõpptulemi saavutamiseks on vajalik kohe alustada kliendiga suhtlemist ning kaasata teda protsessi pidevalt, efektiivsemat kliendisuhtlust toetab samaaegne testimise kaasatus. Antud meetme kasutamisega on võimalik oluliselt vähendada neid vigu, mida võidakse teha arendusprotsessi alguses. Samuti on kliendi kaasatus üks põhilisi Scrum protsessi alustalasid, ilma milleta protsess ei saa efektiivselt toimida. |
| Mõistame muutuvaid olusid, isegi kui need ilmnevad arenduse lõppjärgus. Agiilsed meetodid pööravad sellised muutused meie kliendi konkurentsieeliseks. | Testimise strateegia;<br><br>Testimisvahendite haldus;<br><br>Testimise kaasatus protsessis. | Hea testistrateegia ja efektiivne testimisvahendite haldus võimaldab vähendada muutuvatest nõuetest tulenevaid riske ning minimeerida muutustest tulenevaid mõjusid testimisprotsessile. |

| | | |
|---|---|---|
| Tarnime tarkvara nii tihti kui võimalik, soovitavalt iga paari nädala kuni paari kuu tagant. | Testimise protsessi haldus;<br><br>Testimise strateegia;<br><br>Testkeskkond;<br><br>Testimisvahendid. | Pideva tarnimise tagamiseks on vaja pidevat testimist. Seega on tarvilik, et testimisprotsess ja –eesmärk oleks hästi hallatud. Stabiilne testkeskkond ja sobivate vahendite kasutamine võimaldab kvaliteedispetsialistidel teha efektiivsemat tööd. |
| Valdkonna spetsialistid ja tarkvaraarendajad peavad töötama igapäevaselt koos kogu projekti vältel. | Kvaliteedispetsialistide oskused;<br><br>Testimise kaasatus protsessis. | Selleks, et meeskond saaks hästi toimida on tarvilik, et sinna kuuluksid erinevate teadmiste- ja oskustepagasiga liikmed (ärianalüüs, arendus, testimine), kuna kõiki teadmisi on vajalik hakata rakendama juba arendusprotsessi alguses. |
| Projekti edukuse aluseks on motiveeritud inimesed. Loo neile meeldiv ja toetav töökeskkond ning nad saavad iseseisvalt tööga hakkama. | Kvaliteedispetsialistide oskused;<br><br>Kliendi pühendumine;<br><br>Testimise organisatsioon. | Motivatsioon on üks võtmefaktoreid eduka kvaliteedispetsialisti juures. Kliendi pühendumine ja protsessi toetamine loob eelduse, et klient tajub testimise väärtust ning hindab selle tulemeid. Kvaliteedispetsialisti tööd muudab optimaalsemaks ka läbimõeldud protsessid ja nende sobitumine projekti läbi viiva organisatsiooniga. |
| Kõige tõhusam ja tulemuslikum viis info jagamiseks arendusmeeskonnas on näost näkku vestlus. | Suhtlemine;<br><br>Kvaliteedispetsialisti oskused. | Antud võtmealad rõhutavad efektiivse suhtlemise olulisust, kuna see tagab, et kõik osapooled mõistavad teineteist |

| | | |
|---|---|---|
| | | ning luuakse ühised eesmärgid, mille poole koos püüeldakse. Kvaliteedispetsialisti oskused ja professionaalsus on sellele aga eelduseks. |
| Edu peamiseks mõõdupuuks on töötav tarkvara. | Vigade haldus;<br><br>Meetrika. | Hea viis toote kvaliteedi tõestamiseks on näidata leitud vigu, nende arvu ning selle alusel eeldada nende puudumist toodangusse lastavas versioonis. Hea vigade haldamise protsess võimaldab luua head ülevaadet. |
| Agiilse tarkvaraarenduse protsessid soodustavad jätkusuutlikku arendust. See tähendab, et projektiga saab samas tempos jätkata määramata aja jooksul. | Kvaliteedispetsialisti oskused;<br><br>Hindamine ja planeerimine;<br><br>Testimise protsessi haldus. | Selle põhimõtte täitmiseks on vajalikud nii kvaliteedispetsialisti professionaalsed teadmised kui ka suhtlemisoskused. Hindamise ja planeerimise kompetents on tarvilik, et toetada projekti ühtlast tempot ja jätkusuutlikust. |
| Tehnilist täiuslikkust ja head disaini pideva tähelepanu all hoides tagatakse tarkvaraarenduse kiirus ja paindlikkus. | Kvaliteedispetsialisti oskused;<br><br>Testimise kaasatus protsessis;<br><br>Testidisain. | Kõik seotud võtmealad on vajalikud selleks, et kvaliteedispetsialist suudaks mõista ja toetada nii tehnilisi kui ka disaini küsimusi juba arenduse algusest. |
| Lihtsus - ebavajaliku töö tegematajätmise kunst - on väga oluline. | Testimise strateegia. | Antud eesmärgi täitmiseks on vajalik optimaalse ja efektiivse testimise strateegia loomine, mis võimaldaks suurt testide katvust minimaalsete kuludega. Testistrateegia loomise üks |

| | | |
|---|---|---|
| | | põhimõtteid peab kindlasti olema „Pole riski, pole testi!“ |
| Parimad arhitektuurilised lahendused, nõuded ja disain<br><br>tekivad iseorganiseeruvates meeskondades. | Kvaliteedispetsialisti oskused;<br><br>Testimise organisatsioon. | Üks põhielementidest ise-organiseeruva meeskonna tekkimiseks on selgus meeskonnaliikmete rollides, ülesannetes ja vastutustes. Väljatoodud võtmealad loovad selgusele aluse. |
| Meeskond otsib regulaarselt võimalusi saamaks veelgi tõhusamaks ja muudab end vastavalt vajadusele. | Kvaliteedispetsialisti oskused;<br><br>Testimise protsessi haldus;<br><br>Suhtlemine;<br><br>Meetrika. | TPI *Next* mudel julgustab igati pidevat arengut ja annab erinevaid soovitusi, kuidas areneda nii personaalselt kui ka meeskonna tasandil. Meetrika võimaldab omaltpoolt identifitseerida neid kohti, mida on kõige rohkem tarvilik parendada. |

# Lisa 3

Rakenduse X sprindi testiplaan.

1. **Skoop:**
    a. Sprinti võetud tööde testimine kasutajaloo (*story*) ja ülesannete (*task*) tasemel.
    b. Rakenduse üldine kirjeldus (sh arenduses kaasatud komponentide ja nende omavaheliste seoste kirjeldus);
    c. Funktsionaalsuse üldine kirjeldus, riskide ja regressiooni nimekiri.
2. **Testimismeetod:**
    a. Vastuvõtukriteeriumite ja nende alusel loodud testilugude testimine;
    b. Uuriv testimine (testiideed salvestatakse ja esitatakse raportina);
    c. Manuaalne ja automatiseeritud integratsioon;
    d. Ühiktestid (Katvus vähemalt 80%) ja pidev integratsioon;
    e. Kasutusmugavuse testimine – enne sprindi algustöid täpsustatakse kliendi vajadusi ning selleks kasutatakse näited, prototüüpe jms;
    f. Ühised läbivaatused (sh koodi ja sprindi läbivaatused).
3. **Riskid:**
    a. Kasutajaliidese pidevad muutused;
    b. Konfliktid testkeskkonnas teiste arendusmeeskondadega;
    c. Sõltuvused teiste süsteemidega.
4. **Regressioon:**
    a. Projekti regressiooniteste jooksutatakse pidevalt sprindi jooksul öösiti;

b. Regressioonitestide hulka lisatakse positiivsed stsenaariumid ja kõrge riskiga vea olukorrad;

c. Regressioonitestide hulka lisatakse võimalikke riski- ja kokkupuute kohti teiste projektidega.

d. Regressioonitestide raames peab viima läbi koormustestid, et mõõta, ega lisatud funktsionaalsus ei ole mõjutanud olemasoleva süsteemi võimekust.

**5. Keskkonnad:**

a. Testimist viiakse läbi eraldi stabiilses ja täisfunktsionaalses keskkonnas;

b. Testkeskkonda on paigaldatud tarne jaoks loodud tarkvara versioon;

c. Rakendus peab töötama korrektselt veebibrauserites (Internet Explorer 11, Firefox 36 ja Chrome 42)

**6. Testitavad funktsionaalsused:**

a. Uue funktsionaalsuse põhiosad (süsteemitestimine);

b. Integratsioon teiste süsteemidega;

**7. Mitte testitavad funktsionaalsused:**

a. Turvatestimine. (Rõhk piiratud kasutajatega rakenduse arendamisel, seega ei ole kriitiline vajadus);

b. Erinevates keskkondades kasutajaliidese testimine;

**8. Andmete valideerimine:**

a. Testimiseks kasutatakse ja luuakse testandmeid käsitsi;

b. Testimiseks kasutatakse hägustatud toodanguandmeid.

# Lisa 4

1. What are testers responsibilities? How do you define a perfect Quality assurance role?
2. What does „automated testing“ mean to you?
3. Does your your team have written automated unit tests?
4. Is your team using unit test framework?
5. Does the team have Continuous Integration environment up and running?
6. Does your team have coverage criteria? (Method, Line, Branch, Path)
    a. How often do you measure those?
7. Does your team have testcases/ideas?
8. Which testing techniques does your team use?
9. Who analyzes test results and logs and checks wheter feature is done?
10. How do you handle defect management? Do you have a systems or some other process?
11. Do you use any other quality assurance measures? Automated code analysis, code reviews etc?
12. Do you have automated regression suite? Would you need one and why?
13. Does your team have automated integration tests?
14. Does your team have stable test environment? Who manages your test env?
15. Do you as a QA have a holistic overview of the system you are testing?
16. How do you know what has been tested?
17. Is testing described in the Definiton of Done?
18. Does your project need non-functional testing? Could you explain a bit.(UX, security, load)?
19. Does you team have agreements/plan what to test before release?
20. Does your team differentiate unit, integration and system tests?
21. Does your project have all of them covered
22. When are system tests performed during sprint?
23. Have you automated system tests?
24. How do you provide traceability of tests?
25. Are you keeping yourself up-to-date with things happening in QA world? How?
26. How do you report bugs? Describe a defect solving lifecycle in you team?

# Lisa 5

**Tabel 10. Kasutajaloo testide mall**

| **Kasutajalugu** | | **Sprint** | x |
|---|---|---|---|
| (id ja nimetus) | | | |
| **Vastuvõtutestid** | | | |
| Test - 1 | | | |
| Test - 2 | | | |
| **Eeldused** | | | |
| Eeldus – 1 | | | |
| Eeldus – 2 | | | |
| **Variatsioonid** | **Oodatav väljund** | **Kommentaarid** | **Staatus** |
| | | | |
| | | | |
| **Märkmed/kommentaarid/küsimused** | | | |
| | | | |

# Lisa 6

Tabelis 10. on kirjeldatud TPI *Next* mudelis püstitatud kontrollpunktide nimekiri võtmealade taseme hindamiseks.

**Tabel 11. Võtmealade kontrollpunktid**

| | Võtmeala/tase | Tase | Kontrollpunktid | Seosed |
|---|---|---|---|---|
| 1 | Kliendi pühendumine | | | |
| | Baas | + | Klient tunneb huvi testimise olemasolu kohta protsessis. | - |
| | Kontrollitud | 1 | Peamised huvitatud osapooled on määratletud ja kvaliteedispetsialistile teada (ehk tootel on omanik). | Projektijuhtimise metoodikad toetavad tooteomaniku pühendumist läbi protsesside. |
| | | 2 | Projekti eelarves on arvestatud testimisega. | |
| | | 3 | Tooteomanik toetab testimist. | |
| | | 4 | Tooteomanik annab sisendit prioriteetide ja riskide kohta (vajalik testistrateegia jaoks). | |
| | Efektiivne | 1 | Kõik võimalikud toote huvipooled on kvaliteedispetsialistile teada. | Muudatuste haldamise süsteem võimaldab tooteomanikul ja teistel huvitatud osapooltel toetada testimisprotsessi. |
| | | 2 | Tooteomanik huvitub aktiivselt testiprotsessist ja testitava tarkvara kvaliteedist. | |
| | | 3 | Tooteomanik ennetab tegevusi, mis mõjutavad tugevalt testimisprotsessi. | |
| | Optimeeritud | 1 | Juhid toetavad testimisprotsessi parendamist ning selle jaoks antakse vajadusel lisa aega. | Küpse arendusprotsessi juurutamine katab ka piisaval tasemel testimisprotsessi. |
| | | 2 | Projekti osapooled on nõus kohanema testimisprotsessiga. | |
| | | 3 | Tooteomanik ja meeskond töötavad pidevalt koos testiprotsessi parendamise eesmärgil | |
| 2 | Testimise kaasatus protsessis | | | |
| | Baas | + | Testimine kaasatakse sprintide lõppfaasis. | - |
| | Kontrollitud | 1 | Testimise eesmärk, skoop ja lähenemine lepitakse kokku varakult ja ühena esimestest testimistegevustest. | Kirjeldatud rollid ja vastutused projekti juhtimise metoodika tasandil. |
| | | 2 | Testimistegevusi teostatakse juba varakult ning enne testide jooksutamise etappi. (Testimine ei ole protsessis „pudelikael“). | |
| | | 3 | Kvaliteedispetsialist on kaasatud planeeringutel ja testimise ning teiste protsessi sõltuvusi arvestatakse. | |
| | | 4 | Kvaliteedispetsialist on kaasatud riskide kaardistamise ja maandamise analüüsi. | |
| | Efektiivne | 1 | Kvaliteedispetsialist osaleb riski ja muutuste mõju analüüsidel. | Struktureeritud vigade haldamise protsess.<br><br>Efektiivsed tööde |
| | | 2 | Kvaliteedispetsialist teostab leitud vigade mõju analüüsi. | |

| | | | | |
|---|---|---|---|---|
| | | 3 | Kvaliteedispetsialist optimeerib ja haldab olemas olevat testime alus dokumente ja vahetulemeid. | planeerimise koosolekud. |
| | Optimeeritud | 1 | Meeskond on kaasatud projekti edukuse hindamisse. Kogemusest õpitakse ja seda kasutatakse järgmistes projektides. | Pideva õppimise protsessi toetamine ettevõtte tasandil. |
| | | 2 | Kvaliteedispetsialistil on kindel roll arendustegevustest, teda aktsepteeritakse ja hinnatakse. | |
| **3** | Testimise strateegia | | | |
| | Baas | + | Teststrateegia põhineb kvaliteedispetsialisti teadmistel ja süsteemi tundmisel. Protsessi kontrollitakse aja ja ressurssi kulu jälgides. Kasutatakse kõige lihtsamaid tehnikaid ja funktsionaalsuste testimist. Puudub kaardistus riskide ja testide kaetuse vahel. | - |
| | Kontrollitud | 1 | Klient nõustub loodud teststrateegiaga. | Riskide haldus protsess, mis loob tugeva aluse heale teststrateegiale. |
| | | 2 | Teststrateegia põhineb toote riskide analüüsil. | |
| | | 3 | Teststrateegia eristab testi tasemeid, kaetust ning lähtub riskide analüüsil. | |
| | | 4 | Kordus testimiseks ja regressiooniks on loodud lihtne strateegia. | |
| | Efektiivne | 1 | Kõik kaasatud osapooled on nõus kokkulepitud ja dokumenteeritud teststrateegiaga. | - |
| | | 2 | Ülekatted ja puudujäägid erinevate testitasemete vahel on läbimõeldud. | |
| | | | Teststrateegia kirjeldab testi disaini tehnikaid. | |
| | Optimeeritud | 1 | Teststrateegia loomist hinnatakse sageli ja vajadusel kohandatakse. | Mõõdikute rakendamine ja efektiivne kasutamine. |
| | | 2 | Teststrateegia edukust hinnatakse kogutud meetrika alusel (nt toodangus esinevad vead). | |
| **4** | Testimise organisatsioon | | | |
| | Baas | + | Testimise eest vastutab põhiliselt äripool. Vastutused ei ole konkreetselt jagatud. Kvaliteedispetsialistidel ei ole ühist nägemust testimisest. | - |
| | Kontrollitud | 1 | Meeskonda on kaasatud kvaliteedispetsialist. | Efektiivne teadmuse jagamise ja pideva õppimise süsteem. |
| | | 2 | Testimisorganisatsiooni töö on juhitud ja struktureeritud. | |
| | | 3 | Kvaliteedispetsialisti tegevused on defineeritud ning ta võtab vastutuse nende eest. | |
| | | 4 | Testimise organisatsiooni tegevus on läbipaistev ja selge kvaliteedispetsialistidele. | |
| | Efektiivne | 1 | Meeskonnad teevad omavahel koostööd ja parendavad testimisprotsessi. | - |
| | | 2 | Testimise organisatsioon arendab kokku lepitud plaani alusel kvaliteedispetsialiste. | |
| | | 3 | Testimiseorganisatsioon toetab meeskondi vastavalt vajadusele. (nt lisatakse | |

| | | | | |
|---|---|---|---|---|
| | | | kvaliteedispetsialiste vajadusel) | |
| | | 4 | Testimismetoodikat jälgitakse. | |
| | Optimeeritud | 1 | Ettevõtte testimisteenused pidevalt arenevad ja neid saab süstemaatiliselt hinnata. | - |
| | | 2 | Testimisorganisatsiooni hinnatakse projektide testimise edukuse alusel. | |
| | | 3 | Testimisorganisatsiooni tulemeid hinnatakse pidevalt teiste sarnastega. | |
| **5** | Suhtlemine | | | |
| | Baas | + | Suhtlus toimub ilma organiseerimatult, kuid info liigub osapoolte vahel. | - |
| | Kontrollitud | 1 | Iga meeskonna liige on teadlik otsuste tegemisest ja sisemisest protsessist. | Koosolekute planeerimine ja memode jäädvustamine. |
| | | 2 | Kvaliteedispetsialist kogub infot kõikidel seotud osapooltelt. | |
| | | 3 | Tagasiulatuvalt on võimalik jälgida tehtud otsuseid. | |
| | | 4 | Kvaliteedispetsialist arutab tooteomanikuga testimise progressist ja toote kvaliteedist. | |
| | Efektiivne | 1 | Kvaliteedispetsialist analüüsib ja jagab vajalikku infot tooteomanikuga. | Huvitatud sihtrühmade ja vajaliku info jagamise allikate analüüs loob aluse edukaks suhtluseks. |
| | | 2 | Kvaliteedispetsialist osaleb olulistel koosolekutel koos tooteomaniku ja teiste huvipooltega. | |
| | | 3 | Kvaliteedispetsialist suhtleb osapooltega professionaalselt. (vajadusel formaalne/mitteformaalne) | |
| | Optimeeritud | 1 | Suhtlemisel kasutatakse parimaid praktikaid ja õpitakse pidevalt. | - |
| | | 2 | Defineeritud on info jagamise strateegia. | |
| **6** | Raporteerimine | | | |
| | Baas | + | Raporteid koostatakse lähtudes kvaliteedispetsialist parimast kogemusest. Puudub kindel struktuur. | - |
| | Kontrollitud | 1 | Raport kajastab testimise aega (kulu), riske ja tulemeid. | Meeskondade omavaheline koostöö ja raportite arendamine ühiselt. |
| | | 2 | Raporteeritakse sageli ning piisava detailsusega, et tooteomanik saab langetada selle alusel otsuseid. | |
| | | 3 | Raporteid esitatakse taasesitataval kujul (mõttekaardid, tabelid vms) | |
| | Efektiivne | 1 | Osapoolte raporteerimise vajadused on täidetud ning nende alusel võetakse vastu otsuseid. | - |
| | | 2 | Raport sisaldab ülevaadet progressist ja võimalikest projekti riskidest. | |
| | | 3 | Raport sisaldab ülevaadet eesmärkidest ja võimalikest toote riskidest. | |
| | Optimeeritud | 1 | Raportid sisaldavad andmestikku ja mõõdikuid, mida on võimalik edaspidi testimisprotsessi parendamiseks kasutada. | Kogu tarkvaraarendus protsessi parendamine. |
| | | 2 | Raporteeritud andmeid ja mõõdikuid kasutatakse arendusprotsessi parendamiseks. | |
| **7** | Testimise protsessi haldus | | | |

| | | | | |
|---|---|---|---|---|
| | Baas | + | Testimisprotsessil on põhiliselt ainult üks etapp – Testide jooksutamine. | - |
| | Kontrollitud | 1 | Projekti plaan on loodud, mis kirjeldab skoobi, rollid j vastutused. | Projekti juhtimise praktikate jälgimine.<br><br>Kvaliteeditagamise protsessi |
| | | 2 | Testiplaan on kooskõlastatud tooteomanikuga. | |
| | | 3 | Testimistegevusi jälgitakse, mõõdetakse ja vajadusel korrigeeritakse. | |
| | | 4 | Testiplaan on kooskõlastatud kõikide huvitatud osapooltega. | |
| | Efektiivne | 1 | Kõrvalekalded testiplaanist arutatakse läbi tooteomaniku ja teiste vajalike osapooltega. | - |
| | | 2 | Muudatused testiplaanis salvestatakse taasesitataval kujul. | |
| | | 3 | Kvaliteedispetsialist võib vajadusel kasutada eraldi ressurssi. (nt arendajate abi) | |
| | Optimeeritud | 1 | Testiprotsessi juhtimist vaadatakse pidevalt üle. | Projekti vahe ja lõpu kokkuvõtted ning analüüsid. |
| | | 2 | Õpitud kogemusi kasutatakse testimisprotsessi parendamiseks. | |
| **8** | Hindamine ja planeerimine | | | |
| | Baas | + | Testimise hindamisel puudub struktureeritud meetod. | |
| | Kontrollitud | 1 | Hindamisel kasutatakse lihtsaid tehnikaid (nt suhtarve). | Hea sisend arendatava töö kirjelduse ja mahuga. |
| | | 2 | Igal testimistegevusel on hinnang, kui kaua seda läbi viiakse ning mis on tulem. | |
| | | 3 | Sõltuvused testimisetapis on kirjeldatud testiplaanis. | |
| | | 4 | Testitegevuste hinnangud ja plaanid on kokkulepitud huvitatud osapooltega. | |
| | Efektiivne | 1 | Hindamisel kasutatakse vähemalt kahte tehnikat. (nt tunnid, suhtarvud, prioriteedid jms) | Testimise planeerimine projekti plaanis.<br><br>Hindamistehnikate kasutamine tööde hindamisel. |
| | | 2 | Testimistegevused on hinnatud ja planeeritud. | |
| | | 3 | Meetrikaid kasutatakse testimise tulemite mõõtmiseks. | |
| | | 4 | Testiplaanis on kirjeldatud testitavuse ülevaade ja hinnang. | |
| | Optimeeritud | 1 | Testiplaanis arvestatakse loodud testide taaskasutamisega tulevikus. | Meetrikate analüüs ja jälgimine kogu ettevõttes. |
| | | 2 | Ettevõtte tasemel on kokku lepitud teatud kvaliteeditagamise võtted ja meetodid. | |
| | | 3 | Võtme andmestik testimisemõõtmiseks on kirjeldatud ettevõtte tasemel. | |
| **9** | Meetrika | | | |
| | Baas | + | Kvaliteedispetsialistil puudub ülevaade ja tõestus olemasoleva toote kvaliteedist. | - |
| | Kontrollitud | 1 | Meetrika kasutatakse kvaliteedi mõõtmisel. | Automaatsed meetrika kogumise ja analüüsimise vahendid. |
| | | 2 | Meetrikat salvestatakse struktuurselt ja süstemaatiliselt. | |
| | | 3 | Salvestatav meetrika on täpne. | |
| | Efektiivne | 1 | Analüüsitakse meetrika kogumisele kuluvat | - |

| | | | | |
|---|---|---|---|---|
| | | | aega ja selle kasulikkust. | |
| | | 2 | Meetrika kogumine ei ole konfliktis protsessiga. | |
| | | 3 | Meetrikad on kirjeldatud testimisprotsessis ja neid kasutatakse efektiivsuse mõõtmiseks. | |
| | | 4 | Meetrika analüüsi alusel tehakse järeldusi. | |
| | Optimeeritud | 1 | Meetrika kasulikkust jälgitakse pidevalt. | - |
| | | 2 | Muutuvate vajaduste alusel korrigeeritakse meetrikate kasutust vastavalt. | |
| **10** | Vigade haldus | | | |
| | Baas | + | Vigu ei salvestata ega kirjeldata taasesitataval kujul. Puudub kindel ülevaade toote kvaliteedist. | - |
| | Kontrollitud | 1 | Vigade haldamise elutsükkel on defineeritud ja kasutusel. | Kokkulepitud töövoog vigade lahendamiseks.<br><br>Vigade haldussüsteemi kasutamine. |
| | | 2 | Vigade kohta salvestatakse seotud kasutajalugu, kirjeldus, staatus ja muud meeskonnaga kokku lepitud parameetrid. | |
| | | 3 | Vigade lahendamise vastutused on jagatud. | |
| | | 4 | Kõik osalised vigade lahendamisel kasutavad vigade haldus vahendit. | |
| | Efektiivne | 1 | Vigade haldussüsteem hoiab informatsiooni vigade staatuse muutumise ja muutjate kohta. | - |
| | | 2 | Kõik vigade lahendamise protsessiga seotud osapooled kasutavad edukalt sama vigade haldussüsteemi. | |
| | | 3 | Vigade haldussüsteem võimaldab automaatset raporteerimist. | |
| | | 4 | Vigade haldussüsteemi kasutamine võimaldab anda infot arenduse tendentside kohta. | |
| | Optimeeritud | 1 | Vigade haldamise juhised on kokkulepitud ja neid jälgitakse. | Põhjuste uurimise ja lahendamise protsess. |
| | | 2 | Kvaliteedispetsialist vastutab vigade haldamise protsessi jaoks meetrika kogumise eest. | |
| | | 3 | Vigu analüüsitakse ja kohandatakse protsessi nende vältimiseks. | |
| **11** | Testimisvahendite haldus | | | |
| | Baas | + | Testimisvahendeid ei hallata ega organiseerita. | |
| | Kontrollitud | 1 | Testimise sisendid on identifitseeritavad nime ja versiooni alusel. | Varajane testimise ja selle tulemite planeerimine. |
| | | 2 | Igal testilool ja –ideel on selge seos testitava süsteemi kasutajalooga. | |
| | | 3 | Kvaliteedispetsialistil on ligipääs kõikidele arendatavatele osadele. | |
| | | 4 | Testimisvahendite haldamine on teada kogu meeskonnale. | |
| | Efektiivne | 1 | Testimise sisendid, rakendus ja muud testimiseks kasutatavad vahendid on identifitseeritavad nime ja versiooni alusel. | Konfiguratsiooni haldus.<br><br>Kasutajalugude haldus. |
| | | 2 | Läbipaistvus on tagatud kasutajalugude ja | |

| | | | | |
|---|---|---|---|---|
| | | | testilugude vahel | |
| | | 3 | Testimisvahendeid hallatakse kokkulepitud struktuuris. (nt SVN, testiraportid Excelis jms) | |
| | Optimeeritud | 1 | Projekti alguses arvestatakse testimise väljundite ja vahendite hoiustamisele. | Projekti tulemite hoiustamise protsess. |
| | | 2 | Juhised on vajalike materjalide hoiustamiseks loodud. Testimisvahendite taaskasutamist mõõdetakse. | |
| | | 3 | Projekti lõppedes hoiustatakse testimiseks kasutatud vahendid ning antakse üle hooldust teostavale meeskonnale. | |
| **12** | Metoodika | | | |
| | Baas | + | Testimismetoodika on valitud lähtuvalt kvaliteedispetsialisti kogemusest ja tema ära nägemise järgi. | - |
| | Kontrollitud | 1 | Testimisprotsess järgib metoodikas kirjeldatud protsessi: sammud, tulemid. | Metoodika kirjeldamine (sh arendus ja hooldus). |
| | | 2 | Testimismetoodika sobitub kasutatava arendusmetoodikaga. | |
| | | 3 | Projektides kasutatakse loodud testimismetoodikat ja seda peetakse kasulikuks. | |
| | Efektiivne | 1 | Testimismetoodika kirjeldab kõiki testimise tegevusi: eesmärk, rollid, tehnikad, eeldused. | |
| | | 2 | On loodud täielik kogu mallidest ja näidetest. | |
| | | 3 | Kohustuslikud, valikulised elemendid testimismetoodikas on kirjeldatud | |
| | | 4 | Kohustuslikke elemente kasutatakse projektides. | |
| | Optimeeritud | 1 | Kvaliteedispetsialistid annavad ja koguvad pidevat tagasisidet testimismetoodika kohta. | Organiseeritud ja küps arendusprotsess. |
| | | 2 | Kasutatavat testimismetoodikat parendatakse pidevalt. | |
| **13** | Kvaliteedispetsialisti oskused | | | |
| | Baas | + | Kvaliteedispetsialisti rolli täidab kas äripool või tehnilised isikud, kellel puuduvad testimise alased teadmised. | - |
| | Kontrollitud | 1 | Kvaliteedispetsialistid on erialaselt haritud ning omavad kogemust struktureeritud testimisalal. | Professionaalsete inimeste värbamine. |
| | | 2 | Kvaliteedispetsialistid on tuttavad kehtestatud testimismeetoditega ning rakendavad neid. | |
| | | 3 | Meeskonnale on olemas laia oskustepagasiga spetsialistid (äri-, valdkonna ja tehnilised spetsialistid) | |
| | | 4 | Kvaliteedispetsialistide oskuseid hinnatakse ning jälgitakse nende üldist oskuste taset. | |
| | Efektiivne | 1 | Kvaliteedispetsialistid on koolitatud (nt *ISTQB*) | - |
| | | 2 | Kvaliteedispetsialistid oskavad selgitada rakendatavaid tehnikaid ja miks neid kasutatakse. | |

| | | | | |
|---|---|---|---|---|
| | | 3 | Kvaliteedispetsialistid naudivad oma tööd ning jagavad oma kogemusi. | |
| | | 4 | Kvaliteeditagamise ülesanded on kirjeldatud, protsessi sobitatud ning eesmärkidega kaardistatud. | |
| | Optimeeritud | 1 | Kvaliteedispetsialistid jagavad aktiivselt oma kogemusi ning täiendavad teadmisi pidevalt lugedes valdkonna spetsiifilist kirjandust ja osaledes koolitustel. | Hästi kirjeldatud ja organiseeritud testimisprotsess (ja arendusprotsess). |
| | | 2 | Kvaliteedispetsialistide arenguplaan on osa ettevõtte personali arenguplaanidest. | |
| | | 3 | Kvaliteedispetsialistid pürgivad oma tööga usalduse ja vastutuse saavutamise poole ning parendavad pidevalt oma tööprotsessi. | |
| **14** | Testidisain | | | |
| | Baas | + | Testilugusid luuakse ilma testidisaini tehnikaid teadlikult kasutamata ning need on tugevas sõltuvuses loojast. | - |
| | Kontrollitud | 1 | Testilugusid kirjeldatakse ühtsel loogilisel kujul. | Protsessi disain. |
| | | 2 | Testilugu koosneb kirjeldusest, sammudest ja oodatud tulemist. | |
| | | 3 | Testilood annavad ülevaate, milline osa testitavast süsteemist on kaetud. | |
| | Efektiivne | 1 | Testilood on lihtsalt mõistetavad ja hallatavad erinevate osapoolte poolt. | Protsessi disain, kirjeldab testidisaini tehnikate kasutuse. |
| | | 2 | Testilugude pool kaetud funktsionaalsuse tase on teada. | |
| | | 3 | Kasutatakse olemasolevaid struktureeritud testidisaini tehnikaid. | |
| | | 4 | Staatilise testimise läbiviimiseks kasutatakse kontrollnimekirju. | |
| | Optimeeritud | 1 | Leitud vigu analüüsitakse ning uuritakse välja põhjus, miks testimises seda ei leitud. | Efektiivne vigade haldamise protsess. |
| | | 2 | Testilugusid hinnatakse ja kontrollitakse regulaarselt. | |
| | | 3 | Testidisaini tehnikaid hinnatakse taaskasutamise perspektiivist ning kohandatakse vajadusel. | |
| **15** | Testimisvahendid | | | |
| | Baas | + | Testimisevahendeid kasutatakse juhuslikul viisil. | - |
| | Kontrollitud | 1 | Testimisvahendeid kasutatakse spetsiifiliste tegevuste täitmiseks, mis on kirjeldatud testimise eesmärkides. | - |
| | | 2 | Jagatakse teadmust kasutatavatest testimisvahenditest. | |
| | | 3 | Kõik osalised, kes kasutavad mõnda spetsiifilist testimisvahendit peavad seda kasulikuks. | |
| | Efektiivne | 1 | Kasutatavad testimisevahendid kiirendavad testimisprotsessi ning parendavad hallatavust. | Professionaalsete arendusvahendite kasutamine. |
| | | 2 | Kvaliteedispetsialistil on vaba ligipääs | |

| | | | | |
|---|---|---|---|---|
| | | | testimisvahenditele. | |
| | | 3 | Äri on teadlik iga uue testimisvahendi kasutusele võtust. | |
| | | 4 | Testimisvahendite kasutus on integreeritud testimisprotsessi. | |
| | Optimeeritud | 1 | Testimisvahendeid kasutatakse sihipäraselt ja pidevalt. | - |
| | | 2 | Testimisvahendite kasutamine, parimad praktikad ja vahendite kirjeldus säilitatakse tuleviku projektide jaoks. | |
| | | 3 | Testimisvahendeid hinnatakse, kas nad on vastanud püstitatud eesmärkidele. | |
| **16** | Testkeskkond | | | |
| | Baas | + | Kvaliteedispetsialist kasutab testimiseks keskkonda, mida ei ole eelnevalt disainitud ega vasta nõuetele. Puudub kontroll keskkonna üle. | - |
| | Kontrollitud | 1 | Testkeskkonna nõuded on kirjeldatud. | Tarne haldus.<br><br>Muutuste haldus. |
| | | 2 | Testkeskkonna haldamise vastutused on kirjeldatud ja jagatud. | |
| | | 3 | Testkeskkond on alati kättesaadav. | |
| | | 4 | Kvaliteedispetsialisti teavitatakse testkeskkonna muudatustest õigeaegselt. | |
| | Efektiivne | 1 | Testkeskkond vastab kirjeldatud nõuetele. | Teenuste haldamise raamistike kasutamine. (nt ITIL) |
| | | 2 | Testkeskkond on loodud järgides loogilist disaini ja funktsionaalseid vajadusi. Tagatakse vajalikud süsteemid, liidestused jms. | |
| | | 3 | Kvaliteedispetsialist on teadlik testkeskkonna füüsilisest ja loogilisest disainist ning kinnitab selle korrektsust. | |
| | | 4 | Seotud osapooltega on sõlmitud SLA (*Service Level Agreement*) lepingud. | |
| | Optimeeritud | 1 | Testkeskkonna eest vastutavad eraldi spetsialistid. | - |
| | | 2 | Testkeskkonna kasutus on kirjeldatud lepingutes. | |
| | | 3 | Testkeskkonna poolt kasutatavad teenused on kirjeldatud ja hallatud. | |

# Lisa 7

Tabelis 12 on toodud parendussoovituste rakendamise tulemus. Rohelisel taustal ja „+“ märgiga märgitud staatus tähendab täidetud soovitust, „-“ märgiga täitmata jäänud soovitust.

**Tabel 12. Parendussoovituste rakendamise tulem**

| | Kontrollpunktid | Soovitused | Staatus |
|---|---|---|---|
| **1** | **Testimise kaasatus protsessis** | | |
| | Testimise eesmärk, skoop, riskid ja metoodika arutatakse projekti põhiliste osapooltega läbi juba varakult projekti alguses; | Kasutajalugude kirjeldamise kvaliteedi hindamiseks kasutada peatükis 6.1.2 kirjeldatud *INVEST* kriteeriumit; | + |
| | Testimistegevusi viiakse läbi juba eelnevalt testide jooksutamise tegevusi. | Kirjeldada projekti testimisprotsess ning teha kliendiga ja meeskonnaga vastavad kokkulepped, võtta kasutusele Scrumile kohandatud protsess; | + |
| | | Leppida kliendiga kokku kvaliteedispetsialisti vastutused ja töö tulemid; | + |
| | | Kirjeldada testimise eesmärk, skoop ja riskid nii sprindi kui tarne põhiselt; | + |
| | | Võtta kasutusele sprindi planeerimise koosolekutel ja kasutajalugude täpsustamise koosolekutel näidete loomise ning vastuvõtutestide loomise praktika; | + |
| | | Luua vastuvõtutestide alusel automatiseeritud testid, millest saab alus „elavaks dokumentatsiooniks“. | - |
| **2** | **Testimise strateegia** | | |
| | Projekti põhilised osapooled nõustuvad testistrateegiaga; | Kirjeldada üleüldine testimisstrateegia, mis on leitud ühisosana vaadeldes erinevaid projekte ning kasutades agiilseid testimise kvadrante ja testistrateegia küsimustikku; | + |
| | Testistrateegia luuakse põhinedes toote riskianalüüsil. | Arendusprotsessi alguses kooskõlastada koos kliendi ja meeskonnaga üldine testimisstrateegia; | + |
| | | Testimistehnikate kasutus tuleb kirjeldada testistrateegias; | + |

| 3 | **Testimise protsessi haldus** | | |
|---|---|---|---|
| | Projekti alguses luuakse testiplaan (mis kirjeldab vähemalt kvaliteedispetsialisti rolli ja kohustused); | Vastuvõtu-, integratsiooni- ja ühiktestide edukas läbimine tuleb kirjeldada Valmimise kriteeriumites; | + |
| | Testiplaan lepitakse kokku kliendiga (täpsustatakse, mida kliendile üle antakse); | Kvaliteedispetsialist kirjeldab koos tooteomanikuga tarne testiplaani: | + |
| | Testimistegevusi jälgitakse ning vajadusel muudetakse; | Toote omanik kirjeldab visiooni ja kuupäevad; | + |
| | Testiplaan lepitakse kokku teiste oluliste osapooltega. | Kaardistatakse huvitatud osapooled ja nende vajadused; | + |
| | | Plaani täiendatakse pidevalt sprintide jooksul kuni toote tarneni; | + |
| | | Võetakse vaatluse alla ka standardi ISO-9216 kvaliteediatribuudid; | - |
| | | Valmimise kriteeriumisse (*Definition of Done*) lisatakse testimise läbimine; | + |
| | | Iga sprint kirjeldatakse kasutajaloo tasemel testiideed (lähtudes vastuvõtukriteeriumitest): positiivsed stsenaariumid, veaolukorrad | + |
| | | Iga sprint kaetakse uus kood madala-taseme testidega (~80% ulatuses); | + |
| | | Iga sprint loodud uuendused kirjeldatakse sobival kujul, et hiljem oleks võimalik analüüsida, milliseid funktsionaalsuseid tarnitakse; | + |
| | | Iga sprint täiendatakse funktsionaalsuse põhist regressiooniplaani, hinnatakse riske, kuidas võivad teised arendused mõjutada ning milliseid teste tuleks lisaks läbida; | + |
| | | Iga sprint valideeritakse uut koodi staatiliste koodianalüüsi vahenditega; | + |
| | | Iga sprindi lõpus on kvaliteedispetsialist võimeline andma tooteomanikule ülevaadet arendatud funktsionaalsusest ja selle kvaliteedist; | + |
| | | Enne sprindi algust luuakse prototüüpe ning kinnitatakse need äripoolega | + |

| | | | |
|---|---|---|---|
| **4** | **Testidisain** | | |
| | Testilugusid luuakse ja kirjeldatakse; | Testistrateegias ja –plaanis tuleb kirjeldada sobivad testimistehnikad ning neid jälgida; | + |
| | Testilood koosnevad sobivast kokkulepitud kirjeldusest. | Testilugusid luuakse ühtse malli alusel (nt lisas 5 toodud näide); | + |
| | | Testilugude loomisel kasutatakse peatükis 6.3 kirjeldatud struktureeritud testidisainitehnikaid ja peatükis 6.3.1.1 kirjeldatud heuristikaid; | + |
| | | Uuriva testimise juhtimiseks kasutatakse peatükis 6.3.1 kirjeldatud sessioonipõhist testimist; | + |
| | | Automatiseeritud testide loomisel tuleb lähtuda automatiseerimise püramiidi põhitõdedest ning luua esmalt ühik- ja komponenttestid; | + |
| | | Automaattestid tuleb luua ühik- ja integratsioonitestide disainitehnikaid jälgides; | + |
| | | Loodavad automaattestid peavad vastama automaattestide heale tavale; | + |