



Lauri Jyrkinen

Potilasmonitorin järjestelmätason vaatimusten testiautomaation kattavuusanalyysi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

22.2.2024

Tiivistelmä

Tekijä:	Lauri Jyrkinen
Otsikko:	Potilasmonitorin järjestelmätason vaatimusten testiautomaation kattavuusanalyysi
Sivumäärä:	31 sivua
Aika:	22.2.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaajat:	Matti Välikylä Lehtori, Joonas Vainio Lead Automation Engineer

Tämä insinöörityö keskittyy potilasmonitorin järjestelmätason vaatimusten analysointiin testiautomaation näkökulmasta GE HealthCaressa. Työn tarkoituksena on tutkia, miten testiautomaation kattavuutta voidaan tehostaa ja tunnistaa esteet, jotka rajoittavat fysiologisten parametrien automaatiota.

Järjestelmätason vaatimusten analyysissä keskitytään erityisesti aikaa vievimpien parametrien vaatimuksiin ja niiden testimenetelmiin, kun pyritään löytämään yhtäläisyyksiä ja autetaan yritystä löytämään mahdollisia ratkaisuja testiautomaation parantamiseksi. Analyysi kohdistuu potilasmonitorien vaatimusten testaamiseen, kun käytetään Robot Frameworkia automaation työkaluna. Sen tarkoituksena oli palvella potilasmonitorin tuotekehityksestä vastaavia tiimejä, mikä auttoi heitä keskittymään tulevaisuuden ohjelmistojulkaisuissa testiautomaation haasteisiin.

Työn lopputuotteena tuotettiin erillinen Excel-tiedosto, jossa oli listattu järjestelmätason vaatimukset eri parametreille jaoteltuna eri luokkiin. Tuloksissa selvisi, että eri parametrien vaatimusten automaation esteinä oli samankaltaisuuksia.

Avainsanat: Testiautomaatio, ohjelmistotestaus, laatu, vaatimukset

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Lauri Jyrkinen
Title: Patient Monitoring System Level Requirement Test Automation Coverage Analysis
Number of Pages: 31 Pages
Date: 22 February 2024

Degree: Bachelor of Engineering
Degree Programme: Electrical and Automation Engineering
Professional Major: Automation Engineering
Supervisors: Matti Välikylä, Senior Lecturer
Joonas Vainio, Lead Test Automation Engineer

This engineering thesis focuses on analyzing system-level requirements of patient monitors from the perspective of test automation at GE HealthCare. The purpose was to explore how to enhance the coverage of test automation and identify obstacles limiting the automation of physiological parameters.

The analysis of system-level requirements specifically targets the requirements and testing methods of the most time-consuming parameters, aiming to identify similarities and assist in finding possible solutions to improve test automation. The analysis focuses on testing requirements using Robot Framework as the automation tool. The purpose is to serve teams responsible for patient monitor product development by helping them focus on test automation challenges in future software releases.

As a final product, a separate Excel file was produced, listing the system-level requirements for different parameters, categorized into different classes. The results revealed similarities in automation barriers across different parameters' requirements.

Keywords: Test Automation, Software Testing, Quality, Requirements

Sisällys

Lyhenteet

1	Johdanto	1
2	Lääkinnällinen laite	2
3	Potilasmonitori	3
4	Ohjelmistotestaus	5
4.1	Ohjelmistotestaus yleisesti	5
4.2	Ohjelmistotestauksen tavat	6
4.2.1	Manuaalitestaus	6
4.2.2	Testiautomaatio	9
4.2.3	Kooditarkastelu	11
5	Robot Framework	12
5.1	Avainsanaohjattu testaaminen	12
5.2	Dataohjattu testaaminen	14
5.3	Testiraportti	15
6	Laatuvaatimukset	17
6.1	Standardit	17
6.2	ISO	17
6.3	ISO 9001	18
6.4	ISO 13485	19
6.5	FDA	19
7	Työn vaiheet ja tulokset	20
7.1	Projektin alustus	20
7.2	Työn tavoitteet	21
7.3	Vaatimusten keräys	22
7.4	Vaatimusten luokittelu	23
7.5	Vaatimusten analysointi parametreittain	24
7.6	Analyysin luotettavuus	28
7.7	Tulokset	28

7.8	Jatkokehitysmahdollisuudet	30
8	Yhteenveto	31
	Lähteet	1

Lyhenteet

API	Application programming interface. Ohjelmointirajapinta.
CE	Conformité Européenne. Valmistajan tai maahantuojan vakuus, että tuote täyttää sitä koskevan EU-lainsäädännön vaatimukset.
EN	European Standard. Eurooppalainen standardi.
EKG	Elektrokardiogrammi. Sydänfilmi.
FDA	Food and Drug Administration. Yhdysvaltain elintarvike- ja lääkeviranomainen.
DOORS	Rational Dynamic Object Oriented Requirements System. Vaatimusten hallintatyökalu.
GitLab	Verkkopalvelu git-versionhallinta tehtävienhallintatoiminnoille.
HTML	Hypertext Markup Language. Avoimesti standardoitu merkintäkieli, jolla voidaan kuvata hyperlinkkejä.
IVD	In vitro device. Lääkinnällinen laite, jota käytetään ihmisestä otettavan näytteen käsittelyyn tai sen analysointiin.
MD	Medical Device. Lääkinnällinen laite.
SpO2	Oxygen saturation. Happisaturaatio.

1 Johdanto

Työn tilaaja on GE HealthCare, joka on yksi maailman johtavan terveysteknologia-alan yritys. Yritys työllistää maailmanlaajuisesti 50 000 työntekijää, joista noin 700 työskentelee Suomessa. Suomen päätoimipisteessä Helsingin Vallilassa tuotetaan potilasmonitoreiden ja potilasmonitoreiden ohjelmistojen lisäksi anestesiaan liittyviä tuotteita sairaaloiden käyttöön. [1.]

Lääketieteen kehittymisen ja uusien tuotteiden syntymiseen vaikuttavat ihmisten ikärakenteiden muutokset. Lääkinnällisiä laitteita kehitetään ja valmistetaan jatkuvasti vastaamaan ihmisten, sairaaloiden sekä hoitoalan ammattilaisten muuttuviin tarpeisiin. Jokaisen lääkinällisen laitteen on täytettävä sille asettamat viranomaismääräykset ja vaatimukset. Näin varmistetaan, että jokainen laite toimii luotettavasti eikä vaaranna potilasturvallisuutta.

Potilasmonitorien ohjelmistoihin tehdään säännöllisiä päivityksiä, jossa lisätään ominaisuuksia sekä parannetaan niiden tietoturvaa. Potilasmonitorin ohjelmistolle suoritetaan aina ohjelmiston validaatio ennen sen julkaisua, missä verifioidaan järjestelmän kriittisimmät toiminnot. Verifiointi tarkoittaa tässä yhteydessä testausta, jossa dokumentoidaan objektiivisin todistein ohjelmiston vaatimusten mukainen toiminta.

Tämän insinööriyön tarkoituksena oli tehdä GE HealthCaren potilasmonitorin järjestelmätason vaatimuksille analyysi, jolla tutkitaan, millä tavoin testiautomaation kattavuutta GE HealthCaren tuotekehityksessä voidaan tehostaa. Analyysissä selvitetään, mitkä ovat tämänhetkiset syyt ja pullonkaulat, miksi tiettyjen parametrien osa-alueita ei olla vielä automatisoitu. Vaatimukset luokitellaan eri kategorioihin, jotka helpottavat analyysin luettavuutta ja auttavat tunnistamaan testaamisen kehityskohteita.

Tässä insinööriyössä käsitellään ainoastaan aikaa vievimpien parametrien järjestelmätason vaatimuksia, vaatimusten testausmenetelmiä ja laatua.

2 Lääkinnällinen laite

Lääkinnällisellä laitteella tarkoitetaan laitteistoa, instrumenttia, ohjelmistoa tai välinettä, jonka tuotteen valmistaja on tarkoittanut käytettäväksi ihmisillä yksinään tai yhdistelmänä. Lääkinnällistä laitetta voidaan käyttää esimerkiksi sairauden diagnosointiin, ehkäisyyn, ennakointiin, hoitoon ja tarkkailuun. [2.]

Lääkinnällisissä laitteissa, pakkauksessa tai myyntipakkauksessa on oltava selvästi merkittynä laitteen nimi sekä laitteen käyttötarkoitus. MD- ja IVD-asetusten mukaisissa tuotteissa on nähtävä MD- tai IVD merkintä tai muu selvä ilmaisu siitä, että kyse on lääkitinnällisestä tai in vitro -diagnostiikkaan tarkoitettu laite. MD lyhenne tulee englanninkielisistä sanoista ”Medical Device”. Lisäksi lääkitinnällisten laitteiden mukana on toimitettava valmistajan tunnistetiedot sekä laitteen turvallisuutta ja suorituskykyä koskevat käyttäjälle tai muille henkilöille merkitykselliset tiedot. [2.]



Kuva 1. Potilasmonitorin tunnistetiedot.

Euroopan unionissa ja Suomessa saa markkinoille tuoda ja ottaa käyttöön ainoastaan vaatimukset täyttäviä lääkitinnällisiä laitteita. Tuote suunnitellaan tuotteen valmistajan määrittelemien vaatimusten ja riskien mukaan. CE-

merkintä toimii osoituksena siitä, että laite täyttää sitä koskevat vaatimukset ja että tuote on tehty lääkinällisten laitteiden standardien mukaisesti. Suomessa lääkinällisten laitteiden vaatimusten mukaisuutta ja alan toimijoita valvoo Fimea. [2.]

3 Potilasmonitori

Potilasmonitori on laite, joka on suunniteltu ja tarkoitettu mittaamaan sekä seuraamaan potilaan elintoimintoja ja fysiologista tilaa. Yleisin käyttöympäristö potilasmonitorille on sairaala ja sitä voidaan käyttää lyhytaikaisessa hoidossa leikkaussalissa tai pitkäaikaishoidossa teho- ja valvontaosastolla.

Potilasmonitorin tärkein tehtävä on antaa luotettavaa tietoa potilaan vitaalielintoimintojen muutoksista, joka ohjaa hoitohenkilökuntaa tekemään varhaisessa vaiheessa oikeat sekä vaadittavat toimenpiteet potilaan tilan turvaamiseksi. [3, s. 947.]

Potilasmonitorit ovat yksi eniten käytetyistä laitteista sairaaloissa. Niitä voidaan löytää sairaalan eri osastoilta, kuten teho-osastolta, heräämöstä, leikkaussalista sekä päivystyksestä. [4.] GE HealthCaren potilasmonitorilla on erilaisia ohjelmistopaketteja, jotka on räätälöity eri yksiköiden käyttötarkoitusta tai käyttöympäristöä varten. Osa potilasmonitorin toiminnoista on lisensoitu, mikä tarkoittaa sitä, että niiden käyttöä varten vaaditaan erillinen lisenssi.



Kuva 2. Potilasmonitori [5].

Yleisimmin seurattavat parametrit, joita potilaalla seurataan, ovat pulssioksimetria (SpO₂), sydänsähkökäyrä (EKG), verenpaine (IP), hengityskaasut ja lämpötila. Valvottavien fysiologisten parametrien määrä vaihtelee osastoittain. Esimerkiksi teho- tai leikkausosastolla potilasmonitorin on pystyttävä seuraamaan ja näyttämään kymmeniä eri parametreja samanaikaisesti, kun taas päivystyksessä saatetaan selvittää vain osalla näistä. Parametrit yleensä esitetään aaltomuotoina ja numeerisina arvoina potilasmonitorin näytöllä. [6.]

Yleensä potilasmonitorit ovat modulaarisia eli niissä ei ole sisäänrakennettuja sensoreita, vaan ne ovat riippuvaisia ulkoisista laitteista ja monitoriin kiinnitetyistä antureista. Potilasmonitoriin kiinnitetyt laitteet toimivat tiedonkeruutyökaluina, jotka muuntavat potilaalta mitatut biologiset signaalit sähköisiksi signaaleiksi ja nämä lähetetään valvontalaitteeseen. [7.]

Potilasmonitorissa hyödynnetään algoritmeja parametrien analysointiin, joka antaa käyttäjälle jatkuvaa tietoa potilaan sen hetkisestä tilasta. Potilasmonitorilla

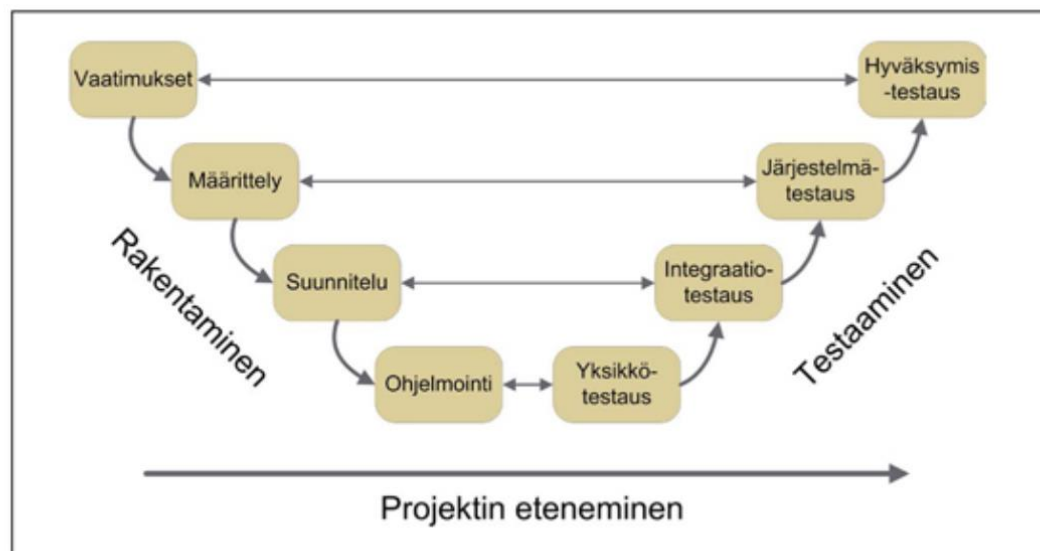
mitattuja arvoja tulkitsemalla voidaan havaita potilaan elintoimintojen muutokset välittömästi, mikä auttaa hoitohenkilökuntaa määrittämään hoidon tarvetta sekä suuntaa.

4 Ohjelmistotestaus

4.1 Ohjelmistotestaus yleisesti

Yleisesti testaaminen nähdään siten, että se on vain yksi työvaihe perinteisessä ohjelmistotuotannon vesiputousmallissa. Vesiputousmallin ideana on se, että projekti etenee vaiheittain ohjelmiston vaatimusten määrittelyllä, jossa kerätään saatavilla oleva tieto asiakastarpeista markkina-analyysiin saakka.

Taustatutkimuksen jälkeen aineistosta kerätään lista vaatimuksista, jotka ohjelman on täytettävä. Vaatimuksia hyödynnetään muodostamaan esivalmistelu, jossa määritellään tarkemmin suunnitelma, jossa määritetään rakennettavan ohjelmiston tai laitteen rakenne, toiminnalliset vaatimukset sekä projektisuunnitelma kehitysprojektin toteutusta varten. [8.]



Kuva 3. V-malli [8, s. 14].

V-malli on lähestymistapa, jossa testaus aloitetaan projektin toteutusvaiheessa ja jatkuu sen loppuun saakka. Tämän aikana laadunvalvonta ja tarkastukset varmistavat, että tuote toimii oikein. Jokaiseen suunnitteluvaiheeseen liittyy oma testausmenetelmä, joka varmentaa tuotteen laatua. [8.]

Ohjelmistotestaus on prosessi, jolla varmistetaan, että ohjelmisto tai tuote vastaa sille asetettuja vaatimuksia ja odotuksia. Ohjelmistotestauksella pyritään myös löytämään vikoja ja osoittamaan että ohjelmisto on käyttötarkoitukseensa sopiva. Käytettävissä on lukuisia testausmenetelmiä, tyyppejä ja tekniikoita toiminnallisten ja ei-toiminnallisten vaatimusten validoimiseksi. [8]

4.2 Ohjelmistotestauksen tavat

Manuaalitestauksen, testiautomaation ja kooditarkastelun kappaleissa käsitellään potilasmonitorin järjestelmätason ohjelmiston testaustapoja yleisellä tasolla.

Potilasmonitorien ohjelmistotestausta tehdään kolmella tavalla: manuaalisesti, automatisoidusti ja kooditarkastelulla. Vaikka testausmenetelmät, joita näissä kappaleissa käydään, on jaettu kolmeen eri kategoriaan, jotkin järjestelmätason vaatimukset voivat edellyttää testaamista usealla testausmenetelmällä, jotta vaatimuksen ehdot täyttyvät.

4.2.1 Manuaalitestaus

Manuaalitestauksella tarkoitetaan ohjelmiston toiminnan varmistamista ja virheiden etsimistä manuaalisesti testaajan tai muun henkilön tekemänä. Manuaalitestaaaja voi imitoida loppukäyttäjää ja yrittää käyttää ohjelmistoa tavanomaisesti tai tarkoituksellisesti väärin. Testaaminen voidaan jakaa kahteen kategoriaan positiiviseen ja negatiiviseen testaamiseen. Positiivisella testaamisella tarkoitetaan sitä, että käytetään laillisia syötteitä, jolloin varmistetaan, että toiminto tuottaa odotettuja tuloksia. Negatiivisessa testaamisessa puolestaan annetaan virheellisiä syötteitä, jota käytetään

poikkeuksellisten tai epänormaalien ja odottamattomien tulosten tarkistamiseen. [9.]

Potilasmonitorien testaamisessa käytetään apuna erilaisia simulaattoreita ja moduuleja, joiden avulla saadaan esimerkiksi näkyviin fysiologisia parametreja ja niiden aaltomuotoja. Yksinkertaisuudessaan manuaalitestaaja kiinnittää kuvan 4 mukaisesti ensin kaapelin monitoriin ja tämän jälkeen simulaattoriin, jolla voidaan simuloida esimerkiksi haluttua aaltoa tai hälytystä potilasmonitoriin.



Kuva 4. Potilasmonitori simulaattorin ja moduulin kanssa.

Testin suorittajan tulee hallita testaustoimintaa ja ymmärtää testattavasta järjestelmästä, mitä pitää testata. [9.] Potilasmonitorin järjestelmätason testaamisessa tulee ymmärtää, mitä vaatimus edellyttää, mitkä ovat testin edeltävyys ehdot, mitä ja miten testataan sekä mikä on toiminnasta poikkeavaa. Järjestelmätason vaatimuksille on tehty omat manuaalitestit, jotka imitoivat loppukäyttäjän tapaa käyttää monitoria. Testeihin kirjataan tarkasti testattava tuote, ohjelmisto sekä laitteisto, mikä on monitorissa kiinni testaushetkellä. Testeissä edellytetään testaajaa kirjaamaan, mitä eri toiminnallisuuksia testin eri hetkillä on muokattu ja mitä arvoja on simulaattoreilla simuloitu. Lopuksi testit merkitään joko "PASS"- tai "FAIL" -tuloksilla.

Testit ja tulokset katselmoidaan lopuksi toisen osapuolen toimesta. Tämän vuoksi dokumentaatio testaamisessa on ensiarvoisen tärkeää, sillä se helpottaa katselmoijaa näkemään, onko testaaminen suoritettu oikein ja lisäksi auttaa mahdollisten virheiden löytämisessä sekä toistamisessa.

Testaaminen edellyttää vahvaa potilasmonitorin vaatimusten, testattavan parametrin sekä simulaattorin käyttöominaisuuksien tietämystä. Mikäli testaaja ei huomioi edellä mainittuja asioita voi syntyä laatuongelmia. GE HealthCare varmistaa, että testaaja on käynyt riittävän perehdytyksen ja vaadittavat koulutukset ennen varsinaista testaamista. Testaajan on hyvä ymmärtää, kuinka merkityksellisessä roolissa toiminta- ja työskentelytavat ovat parhaan mahdollisen laadun saavuttamiseksi. Testaajan sitoutuminen laadun jatkuvaan toteuttamiseen luo asiakkaille ja potilaille luottamusta siitä, että potilasmonitorit ovat turvallisia siinä käyttötarkoituksessa, mihin ne on suunniteltu.

Manuaalinen testaaminen on erittäin sopiva tutkivassa testauksessa, jossa testaajan ammattitaito ja luovuus ovat keskiössä uusien testitapausten kehittämisessä testausprosessin aikana. Tämä lähestymistapa on tehokas, helppo käynnistää ja erityisen soveltuva pienille ohjelmistoprojekteille, joissa se voi toimia jopa ensisijaisena testausmenetelmänä. Manuaalitestaus on erityisen hyödyllinen käytettävyydestestauksessa, joka on haastava automatisoida. [9.]

Manuaalitestauksen merkittävänä haittapuolena on sen suuri resurssitarve. Erityisesti tietojen oikeellisuuden tarkastaminen edellyttää testaajalta huolellista tarkkuutta sekä aikaa. Tämä korostuu regressiotestauksessa, jossa toistuvasti testataan samoja toimintoja ja jokaisessa testauksessa vaaditaan samanlaista tarkkuutta arvojen tarkistamisessa. [9.]

4.2.2 Testiautomaatio

Testiautomaatiolla tarkoitetaan testauksen suorittamista ja raportointia, missä hyödynnetään ulkopuolisia ohjelmia ja työkaluja. Testiautomaatiota voidaan tehdä testaamisen eri tasoilla kuten, yksikkö-, integraatio- ja järjestelmätasoilla. [9.]

Testiautomaatiota käytetään usein ratkaisuna testausajan lyhentämiseen ja manuaalisen testauksen korvaamiseen. On kuitenkin tärkeää arvioida testiautomaation vahvuuksia ja heikkouksia projekteissa. Testiautomaatiolla ei voida korvata manuaalisia testaajia, eikä se ole ratkaisu kaikkiin ongelmiin. Testiautomaatio ei myöskään vähennä testaamista, mutta se on pitkäaikainen sijoitus, jolla tehdään säästöä pitkässä ajanjuoksussa. [9.]

Testiautomaation tavoitteena on testauksen parempi laatu sekä kattavuus. Testiautomaatiolla voidaan laajentaa testien kattavuutta ja tarkkuutta suorittamalla suuri määrä testejä eliminoimalla inhimilliset virheet. Testiautomaation etuina onkin se, että sillä voidaan toistaa sama testi eri arvoilla useaan kertaan.

Testiautomaatio suoritetaan aina oikealla monitorilla sen käyttöliittymää hyödyntäen. Ennen varsinaista testin ohjelmointia testiautomaation kehittäjä perehtyy vaatimukseen, joka pitää sisällään tarkan määrittelyn, testattavan tuotteen sekä ohjelmistoversion. Testiautomaatiota tehdessä tulee huomioida, että testillä saavutetaan riittävä kattavuus. "QMS" eli laadunhallintajärjestelmä asettaa omat ehdot testeille sekä yleisesti testiautomaatiolle. GE HealthCare käyttää Robot Frameworkia, joka on automatisointiin kehitetty viitekehys.

```

*** Settings ***
Documentation      Test Suite for HR alarm annunciation
Suite Setup        Connect To Monitor And Start Patient Simulator
Suite Teardown     Stop Simulator And Disconnect From Monitor
Test Setup         Configure HR High Alarm Limit
Test Teardown      Restore All Alarm Limits
Test Template      Test HR Alarm Activation With High Priority
Resource           patient_monitor.resource
Library            Collections

*** Variables ***
${user_name} =     Monitor
${password} =      PassWord123

*** Test Cases ***
Test HR High Alarm Annunciation
  [Tags]           Requirement_1  MonitorX  Patient_simulator
  # Alarm Limit
  High 180

Test HR Low Alarm Annunciation
  [Tags]           Requirement_2  MonitorX  Patient_simulator
  # Alarm Limit
  Low 40

*** Keywords ***
Test HR Alarm Activation With High Priority
  [Documentation]  This keyword tests HR alarm activation
  [Arguments]     ${hr_high_low}  ${alarm_limit}
  Cause HR Alarm  ${alarm_type}
  Verify Alarm Activation
  Reset HR Alarm

```

Esimerkkikoodi 1. Robot Framework kuvitteellinen esimerkkitapaus

Oletetaan, että simulaattorit on kytketty potilasmonitoriin ja se on valmis testattavaksi. Esimerkkikoodi 1 kuvastaa hyvin, miten robotilla suoritettu yksinkertainen testi edellyttää monia vaiheita. Testiautomaatiota tehdessä on huomioitava:

- testikokonaisuuden alustus ja lopetus (Suite Setup, Suite Teardown)
- testin alustus ja lopetus (Test Setup, Test Teardown)
- tvainsanat (Keywords)
- kirjastot ja resurssit (Resource, Library)

- dokumentaatio (Documentation).

Testitiedoston valmistuttua testiautomaation kehittäjä vie muutokset GitLab versiohallintatyökaluun. GitLab:ssa tehdään koodin tarkastelu, missä annetaan kommentteja ja ehdotuksia muutoksille. Kun muutokset on korjattu, yhdistetään testitiedosto päähaaraan, joka sisältää kaikki julkaistut muutokset ja on tämän jälkeen kaikkien yrityksessä toimivien testiautomaatiokehittäjien saatavilla.

Testiautomaatio siis tekee ohjelmistotestauksesta helpompaa, nopeampaa ja luotettavampaa, jos se toteutetaan oikein. Voitaisiin sanoa, että se on välttämätöntä nykypäivän nopeasti muuttuvassa ohjelmistojen toimitusympäristössä. Yleensä testiautomaatio nähdään vaihtoehtona aikaa vievälle ja työläälle manuaaliselle testaukselle. Testiautomaatio käyttää ohjelmistotyökaluja ja kehyksiä useiden testien suorittamiseen toistuvasti varmistaakseen, että ohjelmisto toimii, kuten se on suunniteltu, kun muutoksia tehdään.

4.2.3 Kooditarkastelu

Kaikkien vaatimusten testaaminen manuaalisesti tai testiautomaatiolla ei aina ole mahdollista. Tämän vuoksi ne testataan tarkastelemalla ja testaamalla ohjelmiston koodia sekä sen käyttäytymistä. Tällaiset vaatimukset voivat olla sellaisia, joita joko manuaalitestaaaja tai robotti eivät kykene näkemään tai generoimaan. Vaatimus voi edellyttää, että potilasmonitori tukee parametrin arvoja tiettyyn raja-arvoon saakka, mutta simulaattori ei kykene simuloimaan stabiilia arvoa potilasmonitorille. Tässä tapauksessa ohjelmistokehittäjä voi testata potilasmonitorin ohjelmistoa syöttämällä haluttua arvoa ja tarkastelemalla, kuinka ohjelmisto ja potilasmonitorin käyttöliittymä käyttäytyvät

Ohjelmistotestauksen kooditarkastelu menetelmää pyritään kuitenkin välttämään, sillä se ei ole kustannustehokasta ja nykypäivän potilassimulaattorit pystyvät generoimaan vaatimuksen täyttymistä edellyttäviä arvoja.

5 Robot Framework

Robot Framework on saanu alkunsa Pekka Klärckin 2006 vuonna tehdystä diplomityöstä “Data-Driven and Keyword-Driven Test Automation Framework”, josta on tullut suosittu työkalu ohjelmistojen automaattiseen testaukseen. [10.]

Robot Framework on avoimen lähdekoodin testiautomaatiokehys, joka tunnetaan avainsanaohjatusta testaustavastaan ja helppokäyttöisyydestään. Se on python-pohjainen ja se antaa käyttäjälle mahdollisuuden luoda korkean tason testitapauksia, jotka voidaan helposti kääntää koneellisesti suoritettaviksi automaatio skripteiksi. Viitekehyksessä on runsaasti kirjastoja ja työkaluja, joiden avulla sitä voidaan käyttää erityyppisiin testauksiin mukaan lukien hyväksymistestaus ja yksikkötestaus. [11.]

Robot Frameworkin yksi tärkeimmistä ominaisuuksista on sen avainsanaohjattu testaustapa, joka mahdollistaa korkean tason testitapausten luomisen taulukkosyntaksissa, jolloin teknisten sekä ei-teknisten käyttäjien on helppoa kirjoittaa ja ylläpitää testejä sekä tehdä yhteistyötä testitapausten luomisessa. Robot Frameworkissa on runsaasti sisäänrakennettuja kirjastoja käyttöliittymien testaamiseen, mutta sen modulaarisen arkkitehtuurin vuoksi se on laajennettavissa lisäkirjastoilla ja työkaluilla, joita voidaan hyödyntää automatisoimaan erilaisia testitapauksia. [11.]

Sen joustavuus ja tuki erilaisille käyttöjärjestelmille ja työkaluille tekevät siitä monipuolisen valinnan erilaisiin ohjelmistokehitys- ja testaustarpeisiin. Kehys on erityisen hyödyllinen sen käyttäjäystävällisen syntaksin ja mukautuvuuden vuoksi, mikä yksinkertaistaa testiautomaatioprosessia ja parantaa testausskenaarioiden tehokkuutta. [11.]

5.1 Avainsanaohjattu testaaminen

Avainsanat ovat Robot Framework -testitapausten rakennuspalikoita. Ne edustavat yksittäisiä toimenpiteitä tai vaihetta, jonka testitapaus sillä hetkellä

suorittaa. Avainsanat voivat olla joko käyttäjän määrittämiä tai sisäänrakennettuja. Käyttäjän määrittämät avainsanat ovat testaajan luomia mukautettuja toimintoja, kun taas sisäänrakennetut avainsanat ovat ennalta määritettyjä, jotka sisältyvät Robot Frameworkin omaan kirjastoon. [12.]

Robot Frameworkilla kirjoitetut testitiedostot koostuvat yleensä neljästä osiosta, jotka ovat Settings (asetukset), Variables (muuttujat), Test Cases (testitapaukset) ja Keywords (avainsanat). [12.]

Asetukset eli "Settings" -osiossa määritellään, mitä resurssi-, kirjasto- ja muuttujatiedostoa käytetään testitiedostossa. Kirjastoista voidaan tuoda uudelleen käytettävät avainsanat testitiedoston testitapausten käyttöön. Tässä voidaan määritellä muuttujat sekä alustaa testitapaukset. [12.]

"Suite Setup" eli testisarjan valmistelu suoritetaan kerran aina testitiedoston alussa. Tässä osiossa määritellään esivalmistelut, mitä kaikki testitiedoston testitapaukset tarvitsevat. "Suite Teardown" eli testisarjan purku suoritetaan, kun kaikki testitiedoston testitapaukset on suoritettu. Tässä kohdassa määritellään kaikki toimenpiteet, mitä suoritetaan sen jälkeen, kun kaikki testitapaukset on suoritettu. Tämän tarkoituksena on se, että testeissä tehdyt laitteisto- tai ohjelmistoasetusten muutokset eivät vaikuta seuraaviin testitiedostojen testitapauksiin. "Test Setup" eli testitapauksen valmistelussa, määritetään toimenpiteet, jotka suoritetaan ennen jokaista yksittäistä testitapausta. Testitiedosto voi sisältää usean testitapauksen, ja ne voivat vaatia erilaisia konfiguraatioita niiden välillä. "Test Teardown" eli testitapauksen purku, määritetään toimenpiteet, mitkä suoritetaan testitapauksen jälkeen. Tämän tarkoituksena on se, että ennen seuraavan testin aloittamista saadaan haluttu lähtötilanne ennen seuraavan testin aloitusta.

"Variables" kohdassa määritellään testitiedostossa käytettävät muuttuja, johon voidaan asettaa joko yksittäisiä arvoja tai antaa useampia arvoja listamuodossa.

5.2 Dataohjattu testaaminen

Data-Driven Testing (DDT), eli dataohjattu testaaminen on testausmenetelmä, joka mahdollistaa testitapausten suorittamisen eri syötteillä. Testisarjassa on määritetty oma avainsana, jonka toiminnallisuutta voidaan hyödyntää kaikkien testitapausten ajoissa. Tämän avulla voidaan suorittaa sama testiskripti useilla eri datajoukoilla, mikä vähentää toistuvaa koodia ja tekee testisarjoista helppolukuisempia. [13.]

```

*** Settings ***
Library          Simulator
Test Template    Test Monitor With Various Values

*** Test Cases ***
Test HR Alarm Priorities|
  [Template]     Verify HR High Alarm
  Low    80     "HR High"
  Medium 100    "HR High"
  High   140    "HR High"

*** Keywords ***
Verify HR High Alarm
  [Arguments]   ${alarm_priority}  ${alarm_limit}  ${exp_alarm}
  Set HR Alarm Priority    ${alarm_priority}
  Simulate HR Over Alarm Limit    ${alarm_limit}
  Verify Alarm Appears

```

Esimerkkikoodi 2. Robot Framework esimerkki testitapaus DDT-menetelmästä.

Esimerkkikoodi 2: mukaisesti testitapaukselle "Test HR Alarm Priorities" määritetään testipohja "Template". Testipohja ottaa vastaan testitapausten argumentit syötedatana, jotka on siihen määritetty, ja käyttää näitä testin suorittamiseen. Jokainen datarivi käsitellään erillisenä testitapauksena käyttäen siihen määriteltyä testipohjaa.

5.3 Testiraportti

Testisarjan tai tapausten ajojen jälkeen saadaan komentoriville lyhyt yhteenveto testien suorituksesta, missä on listattu kaikki suoritettut testit ja niiden tulokset.

Tämän lisäksi Robot Frameworkilla saadaan yksityiskohtaisempi raportti testien suorituksesta kahdessa osassa: *log.html* ja *report.html*. [12.]

REPORT

Total Statistics							Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests							5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Tag							Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
database							1	1	0	0	00:00:00	<div style="width: 100%; height: 10px; background-color: green;"></div>
example							4	4	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
quickstart							5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
smoke							4	4	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
variables							1	1	0	0	00:00:00	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Suite							Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
QuickStart							5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>

Test Execution Log

- SUITE** QuickStart

Full Name: QuickStart

Source: /Users/jth/Code/QuickStartGuide/QuickStart.rst

Start / End / Elapsed: 20220317 16:25:58.137 / 20220317 16:25:59.140 / 00:00:01.003

Status: 5 tests total, 5 passed, 0 failed, 0 skipped

00:00:01.003

 - SETUP** Clear login database

00:00:00.001
 - TEST** User can create an account and log in

Full Name: QuickStart.User can create an account and log in

Tags: example, quickstart, smoke

Start / End / Elapsed: 20220317 16:25:58.181 / 20220317 16:25:58.302 / 00:00:00.121

Status: PASS

 - KEYWORD** Create valid user fred, P4ssw0rd
 - KEYWORD** LoginLibrary.Attempt To Login With Credentials fred, P4ssw0rd
 - KEYWORD** LoginLibrary.Status Should Be Logged In
 - TEARDOWN** Clear login database

00:00:00.121
 - TEST** User cannot log in with bad password

Full Name: QuickStart.User cannot log in with bad password

Tags: example, quickstart, smoke

Start / End / Elapsed: 20220317 16:25:58.302 / 20220317 16:25:58.422 / 00:00:00.120

Status: PASS

 - KEYWORD** Create valid user betty, P4ssw0rd
 - KEYWORD** LoginLibrary.Attempt To Login With Credentials betty, wrong
 - KEYWORD** LoginLibrary.Status Should Be Access Denied
 - TEARDOWN** Clear login database

00:00:00.120
 - TEST** User can change password

00:00:00.241

Kuva 5. Robot Framework log.html esimerkki [12].

Log.html on tiedosto, joka toimii tarkkana tapahtumien selostuksena. Tiedosto sisältää kaikki testeissä suoritettut toiminnot, niiden tilat, suoritusajat ja mahdolliset lokiviestit. Tämä on erityisen hyödyllinen, kun testinkehittäjät haluavat selvittää tarkasti, missä vaiheessa testi on epäonnistunut ja miksi. Tarkka selostus aikajanan tapahtumista auttaa tunnistamaan testien ongelmakohtat.

QuickStart Report

Generated
 20220317 16:25:59 UTC+02:00
 2 minutes 27 seconds ago

LOG

Summary Information

Status: All tests passed

Start Time: 20220317 16:25:58.137

End Time: 20220317 16:25:59.140

Elapsed Time: 00:00:01.003

Log File: [log.html](#)

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
database	1	1	0	0	00:00:00	<div style="width: 100%; height: 10px; background-color: green;"></div>
example	4	4	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
quickstart	5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
smoke	4	4	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>
variables	1	1	0	0	00:00:00	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
QuickStart	5	5	0	0	00:00:01	<div style="width: 100%; height: 10px; background-color: green;"></div>

Test Details

All
Tags
Suites
Search

Suite:

Test:

Include:

Exclude:

Kuva 6. Robot Framework -report.html esimerkki [12].

Report.html:n tarkoituksena on tarjota kattava yhteenveto testituloksista, joka helpottaa testinkehittäjää ymmärtämään testauksen tilaa kokonaisuutena. Tiedosto sisältää yksityiskohtaiset tiedot jokaisesta suoritetusta testitapauksesta, mutta ei yhtä yksityiskohtaisesti kuten *log.html*. Testaajat saavat välittömän käsityksen, kuinka testit ovat ajoissa menneet ja auttavat tunnistamaan mahdolliset ongelmakohdat nopeasti. [12.]

Tiedostot siis yhdessä tukevat toisiaan testauksen tulosten analysoinnissa. Ne ovat kaksi keskeistä komponenttia testitulosten hallinnassa ja analysoinnissa.

6 Laatuvaatimukset

6.1 Standardit

Standardien tarkoituksena on helpottaa viranomaisten, elinkeinoelämän ja kuluttajien elämää. Standardit edistävät prosessien ja tuotteiden yhteensopivuutta, tiedonvaihtoa ja turvallisuutta. Standardien ansioista palvelut, tuotteet ja menetelmät sopivat niihin olosuhteisiin ja siihen käyttöön, joihin ne on tarkoitettu. Terveysthuoltoalalla standardien tarkoituksena on edistää palvelujen laatua, turvallisuutta sekä yhteensopivuutta. [14.]

Standardit tarjoavat yhteisen linjan ja odotukset, jotka helpottavat terveydenhuollon ammattilaisten välistä kommunikointia ja yhteistyötä parantaen potilashoitoa ja -tuloksia. Standardit tukevat terveydenhuollon teknologioiden ja käytäntöjen kehittämistä sekä käyttöönottoa, mikä varmistaa niiden turvallisuuden, tehokkuuden ja yhteentoimivuuden.

Terveysthuoltoalalla standardien noudattaminen on elintärkeää säädösten noudattamisen, riskien minimoinnin ja terveydenhuollon palvelujen kokonaislaadun parantamisen kannalta. [14; 15.]

6.2 ISO

GE HealthCare on valinnut laadunhallintajärjestelmäksi ISO 9001:2015 -laatustandardin sekä terveydenhuollon laitteille ja tarvikkeille määritetyt EN ISO 13485:2016 -sekä ISO 13485:2016 -standardit. ISO 13485:n ja EN ISO 13485:n erot liittyvät standardin soveltamisalaan ja maantieteelliseen tunnukseseen. "EN" -etuliite viittaa eurooppalaiseen standardiin ja ISO 13485 kansainväliseen standardiin.

ISO 9001:2015 ja ISO 13485:2016 ovat maailmanlaajuisesti tunnustettuja standardeja, jotka ovat tärkeässä roolissa terveydenhuollon teknologia-alalla, koska ne keskittyvät laadunhallintajärjestelmiin. Näillä standardeilla on

keskeinen rooli tuotteiden laadun ja turvallisuuden varmistamisessa, mikä on erityisen tärkeää terveydenhuollossa.

ISO 9001 luo kehyksen laadunhallintajärjestelmille, kun taas ISO 13485 keskittyy erityisesti laadunhallintajärjestelmään lääketieteelliselle laitteille. Molemmat standardit korostavat laadunvarmistuksen äärimmäistä tärkeyttä koko tuotteen elinkaaren ajan. Tämä sisältää kaiken suunnittelusta ja kehityksestä valmistukseen, asennukseen ja palveluun. Terveydenhuollossa tuotteiden ja palveluiden laadun ylläpitäminen on ensisijaisen tärkeää potilasturvallisuuden varmistamiseksi. Sekä ISO 9001 että ISO 13485 kannustavat jatkuvan parantamisen kulttuuriin. Terveydenhuollon organisaatioita, mukaan lukien GE HealthCare, kannustetaan arvioimaan säännöllisesti prosessejaan, tuotteitaan ja palveluitaan tunnistamalla alueet, joilla parannuksia voidaan tehdä. Jatkuva parantaminen johtaa parempaan potilasturvallisuuteen, sujuvampiin toimintoihin ja lisääntyneeseen asiakastytyvyyteen.

Lääkinnälliset laitteet voivat vaikuttaa merkittävästi potilasturvallisuuteen. Nämä standardit ohjaavat organisaatiota tehokkaiden riskienhallintaprosessien toteuttamisessa, mikä auttaa tunnistamaan, arvioimaan ja lieventämään riskejä tehokkaasti, mikä vähentää haittatapaturmien todennäköisyyttä.

6.3 ISO 9001

ISO 9001:2015 on osa ISO 9000 -laadunhallintajärjestelmästandardin perhettä. Se tarjoaa kehyksen organisaatioille kaikissa kokoluokissa ja tyypeissä varmistukseen, että ne voivat johdonmukaisesti täyttää asiakkaiden ja muiden soveltuvien lainsäädännöllisten ja sääntelyllisten vaatimukset. Standardi perustuu useisiin laadunhallintaperiaatteisiin, mukaan lukien vahva asiakaskeskeisyys, ylimmän johdon osallistuminen, prosessilähestymistapa ja jatkuva parantaminen. [16; 17.]

Vaikka ISO 9001:2015 ei ole suoranaisesti suunnattu terveydenhuoltoalalle, sen keskittyminen asiakastyytyvyyteen ja jatkuvaan parantamiseen on olennaista terveydenhuollon teknologiayrityksille. Se auttaa varmistamaan, että tarjotut tuotteet ja palvelut täyttävät asiakastarpeet ja noudattavat säännöksiä. Tämän standardin noudattaminen voi parantaa tehokkuutta ja vaikuttavuutta toiminnassa, mikä johtaa parantuneeseen tuotteiden laatuun ja luotettavuuteen. [18.]

Terveydenhuoltoalan organisaatiot, mukaan lukien lääketieteellisten laitteiden valmistukseen osallistuva yritys GE HealthCare toimivat erittäin säännellyssä ympäristössä. ISO-standardien noudattaminen sujuvoittaa säädöksistä saatuja hyväksyntöjä ja minimoi noudattamatta jättämisen riskin.

6.4 ISO 13485

ISO 13485:2016 on erityisesti suunniteltu lääkintälaitteiden valmistajille. Standardi edustaa vaatimuksia kattavalle laadunhallintajärjestelmälle lääkinnällisten laitteiden suunnittelussa ja valmistuksessa. [19.] Tämä standardi on linjassa ISO 9001:n kanssa, mutta sisältää lisävaatimuksia kuten riskienhallinta, suunnittelun hallinta ja sääntelyvaatimukset, jotka ovat erityisiä lääkinnällisten laitteiden alalla.

ISO 13485:2016 on kriittinen lääkinnällisten laitteiden valmistajille, koska se keskittyy turvallisuuteen ja laatuun. Tämän standardin noudattaminen nähdään usein ensimmäisenä askeleena sääntelyvaatimusten noudattamisessa. Standardi varmistaa, että valmistajilla on tehokkaat prosessit jatkuvasti tuottavat lääkinnällisiä laitteita, jotka ovat turvallisia ja soveltuvat niille tarkoitettuun käyttötarkoitukseen. [19.]

6.5 FDA

Yhdysvaltain elintarvike- ja lääkevirasto eli "FDA" säätelee lääketieteellisten laitteiden markkinoita Yhdysvalloissa. FDA säätelee ja valvoo lääkinnällisiä

laitteita varmistaakseen, että ne ovat turvallisia ja tehokkaita siihen tarkoitettuun käytössä. Ennen Yhdysvaltain markkinoille pääsyä tuotteen on läpäistävä FDA:n arviointiprosessi, joka sisältää laitteen turvallisuuden, tehokkuuden ja laadun tarkastelun. [20; 21.]

FDA luokittelee lääkinälliset laitteet kolmeen eri luokkaan riippuen niiden riskitason, sääntelyn valvonnan tarpeen perusteella, joita käytetään tuotteen turvallisuuden ja tehokkuuden takaamiseksi. Potilasmonitorit on luokiteltu luokkaan kaksi, mitkä vaativat FDA:n hyväksynnän 510(k) prosessin kautta. Tällä voidaan osoittaa, että laite vastaa olennaisesti laillisesti markkinoitua laitetta.

7 Työn vaiheet ja tulokset

Tässä luvussa kuvataan insinööriyön eri vaiheita sekä sen tärkeimpiä havaintoja ja tuloksia. Luvussa kuvataan menetelmät yleisellä tasolla, sillä tarkat kuvaukset vaiheiden sisällöstä, ratkaisuista ja tuloksista ovat salassa pidettävää tietoa.

7.1 Projektin alustus

Projektin lähtökohtana oli tutkia tuotekehityksen potilasmonitorin järjestelmätason vaatimuksia tehden niistä analyysi, joka auttaisi GE HealthCare selvittämään, mitkä ovat tämänhetkisiä pullonkauloja potilasmonitorin vaatimusten verifikaation automatisoinnille. Tällä hetkellä potilasmonitorin järjestelmätason verifikaatio toteutetaan manuaali-, testiautomaatio- sekä kooditarkastelutestausmenetelmin.

Järjestelmätason verifikaatio suoritetaan aina ennen virallista ohjelmistojulkaisua. Verifikaatiolla varmistetaan, että koko järjestelmä toimii odotetulla tavalla ja täyttää sille määritetyt vaatimukset. Tällä tavalla tuotteen valmistaja kykenee osoittamaan viranomaisille, että markkinoille tuotu laite toimii sille asetettujen viranomaismääräysten mukaisesti. Järjestelmätason

verifikaatio on osa ohjelmiston validointia, mihin kuuluu verifikaation lisäksi suunnittelu, testaaminen, jäljittäminen kokoonpanon hallinta ja muut ohjelmistotuotannon näkökohdat, jotka tukevat päätelmää, että ohjelmisto voidaan julkaista markkinoille.

Järjestelmätason vaatimusten analyysin tarkoituksena on auttaa yritystä saamaan parempi kuva testiautomaation tämän hetken haasteista ja syistä, miksi osaan vaatimuksista ei ole tehty vielä testiautomaatiota. Työssä yritetään löytää eri vaatimusten tai parametrien välillä samankaltaisuuksia, mihin GE HealthCare voi tarvittaessa kehittää mahdollisia testiautomaation ratkaisuja.

7.2 Työn tavoitteet

Projekti tavoitteena on analysoida aikaa vievimät järjestelmätason vaatimukset, joiden testaaminen toteutetaan tällä hetkellä kokonaan tai osin manuaalisesti. Analyysissa tutkitaan, millä tavalla testiautomaatiota voidaan GE HealthCaren tuotekehityksessä tulevaisuudessa tehostaa.

Tämän työn tarkoituksena on selvittää, mitkä ovat tämänhetkiset syyt, rajoitteet ja pullonkaulat järjestelmätason vaatimusten automatisoinnille.

Järjestelmätason vaatimusten testaaminen on hyvin työlästä sekä aikaa vievää, sillä niiden testaamisessa testataan ohjelmistoa, joka on koottu yhdeksi toiminnalliseksi kokonaisuudeksi. Järjestelmätestauksessa testataan sitä, että järjestelmä toimii kokonaisuutena ja että se toteuttaa sille määritellyt tavoitteet. Testiautomaation mahdollistaminen nopeuttaa raskasta järjestelmätason testausprosessia. Automatisoituja testejä voidaan suorittaa tehostetusti ja toistuvasti yhdenmukaisissa olosuhteissa varmistaen vakaa- ja korkealaatuinen lopputuote. Testiautomaation avulla voidaan tunnistaa ohjelmiston viat ja ongelmat aikaisemmin kehitysjaksossa, mikä vähentää kalliiden käyttöönoton jälkeisten korjausten riskiä. Terveystieteiden sääntelystandardien noudattaminen on ratkaisevan tärkeää. Automaattinen testaus auttaa varmistamaan, että ohjelmisto täyttää jatkuvasti sille asetetut standardit ja vaatimukset.

Tässä työssä luodaan erillinen tiedosto, mikä sisältää analysoidut vaatimukset eri luokissa ja niiden tarkemmat syyt automatisoinnin esteille. Sivutuotteena saadaan lista järjestelmätason vaatimuksista, mitkä ovat mahdollisia automatisoida tämänhetkisillä työkaluilla. Tuotekehitys voi näin ollen investoida aikaa kattamalla nämä vaatimukset testiautomaatiolla, mikä helpottaisi myöhemmin manuaalitestaaajien työtaakkaa.

7.3 Vaatimusten keräys

Terveystieteiden laitteen valmistukseen liittyy tuhansia vaatimuksia, jotka johtuvat pääasiassa laitteen käytön turvallisuuden ja tehokkuuden takaamisesta. Laitteet voivat olla elintärkeitä potilaiden hoidossa, joten niiden on oltava luotettavia ja toimia odotetulla tavalla.

GE HealthCare käyttää Rational DOORS -nimistä työkalua ohjelmistotestauksen vaatimustenhallintasovelluksena. Ohjelmassa voidaan helposti ylläpitää ja päivittää vaatimuksia sopimaan eri ohjelmistoversioille sekä tuotteille.

Projektin lähtökohtana oli kerätä potilasmonitorin järjestelmätason vaatimukset, joiden pohjalta tehdään ohjelmiston varmentaminen eli verifikaatio. Verifikaatiolla tarkoitetaan testaamista, missä dokumentoidaan objektiivisin todistein ohjelmiston vaatimustenmukainen toiminta.

Järjestelmätason vaatimukset ovat ylempään tason vaatimuksia, jotka määrittelevät järjestelmän toiminnan yleisellä tasolla. Järjestelmätason vaatimus voi olla esimerkiksi ”Järjestelmä tukee sydämen iskutilavuuden hälytysrajojen muokkaamista”. Järjestelmätason vaatimusten pohjalta tehdään testit, jotka ovat määritelty tukemaan loppukäyttäjän käyttötappauksia sekä kattamaan kaikki mahdolliset tilanteet. Tällä tavalla voidaan testata myös järjestelmän käyttäytymistä virhetilanteissa.

Osa järjestelmätason vaatimuksista voivat koskea useampaa tuoteryhmää, jotka toiminnallisuudeltaan saattavat poiketa toisistaan. Tämän vuoksi työssä käytettiin poikkeuksellisen runsaasti aikaa vaatimusten, käytössä olevien simulaattorien, manuaalitesteihin ja testiautomaatiotestien perehtymiseen.

7.4 Vaatimusten luokittelu

Vaatimusten ja parametrien runsaan määrän vuoksi tultiin lopputulokseen, että vaatimusten testiautomaation esteet on hyvä jakaa eri pääluokkiin.

Luokittelemalla vaatimusten testiautomaation esteet eri luokkiin saadaan selkeä rakenne analyysiin, mikä helpottaa esteiden tunnistamista ja ymmärtämistä eri parametrien välillä. Vaatimusten analyysissä päädyttiin seuraaviin pääluokkiin:

- audiovisuaaliset esteet
- rajapinnan puuttuminen
- ohjelmiston haasteet
- työkalut
- mahdollista mutta ei ole vielä testiautomaatiota.

Yllä listattuja pääluokkia voidaan pitää yleisimpinä syinä testiautomaation haasteille ja esteille. Analyysi näistä luokista auttaa tunnistamaan keskeiset kehityskohteet ja mahdollistaa tarkemman suunnitelman tekemisen.

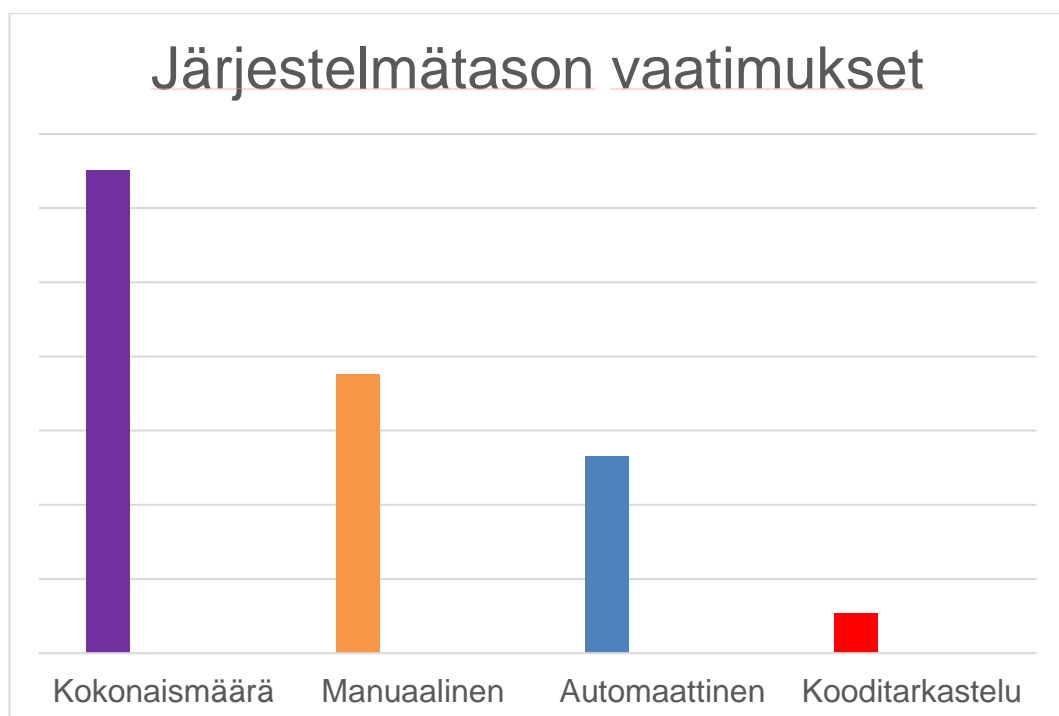
Kukin luokka keskittyy erityyppisiin haasteisiin, mikä mahdollistaa tarkemman ja johdonmukaisemman esteiden arvioinnin. Luokittelussa avataan tarkemmin syyt automaation esteille. Tämä auttaa tunnistamaan erityiset alueet, joissa testiautomaatio kohtaa ongelmia ja tarjoaa mahdollisuuden suunnata resursseja tehokkaammin tai investoimaan aikaa näiden esteiden ratkaisemiseksi. Tämän

lisäksi luokittelu auttaa viestimään löydöksiä selkeästi eri sidosryhmille edistään strategisten päätösten tekemistä testiautomaation kehittämisessä.

7.5 Vaatimusten analysointi parametreittain

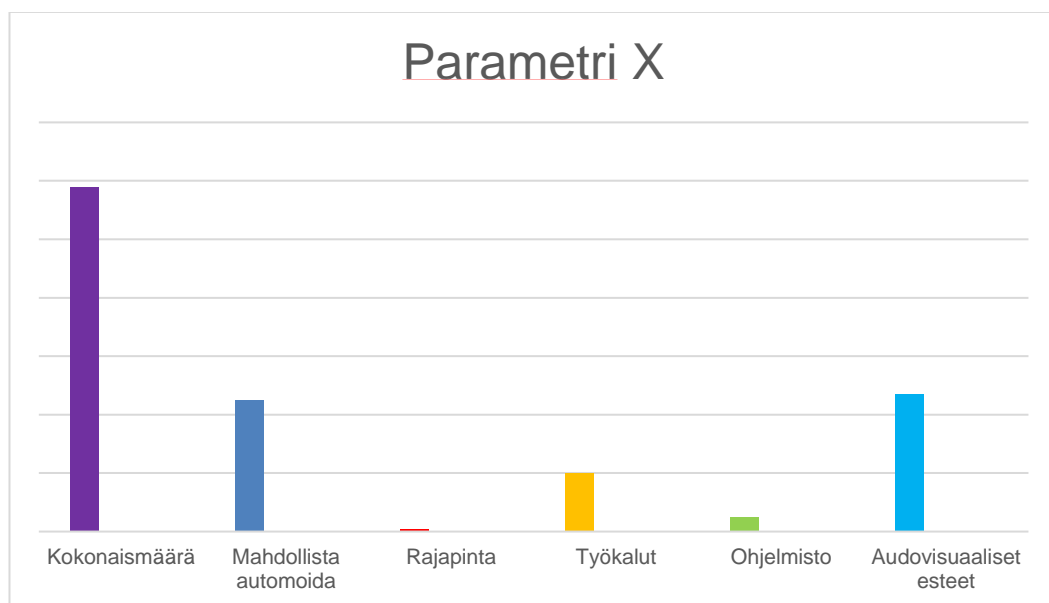
Tässä luvussa käydään läpi yleisellä tasolla, mitkä ovat olleet esteitä ja haasteita eri järjestelmätason vaatimusten testiautomaatille. Kappale ei sisällä parametrien vaatimusten tarkkoja lukumääriä eikä tässä käsitellä testiautomaation esteiden tarkkoja syitä, jotka ovat erillisessä tiedostossa tuotu tuotekehityksen parissa työskentelevien testiautomaation kehittäjille esiin.

Tämän insinööriyön lähtökohtaisena ajatuksena oli keskittyä järjestelmätason vaatimukseen, joiden manuaalitestaaminen on työläintä. Tämä lähestymistapa ei kuitenkaan ollut kaikista edullisin, sillä vaatimuksien ja niiden manuaalitestitapauksien läpikäymiseen olisi kulunut runsaasti aikaa. Tästä syystä projektissa päädyttiin analysoimaan ensiksi parametrit, jotka sisälsivät eniten vaatimuksia.



Kuva 7. Järjestelmätason vaatimuksien testiautomaation nykytilanne

Kuva 7 näyttää nykytilanteen, missä suhteessa eri testaustavat ovat järjestelmätason vaatimusten kokonaismäärään. Manuaalisen, automaattisen ja kooditarkastelun pylväiden yhteiskoko ylittää kokonaismäärän, sillä osa vaatimuksista verifioidaan käyttämällä useaa verifiointitapaa. Pylväsdiagrammin nykytilan testimenetelmien katselmoinnissa pitää huomioida se, että pylväiden koko vaihtelee kuukausittain. Uusien ominaisuuksien kehittäminen potilasmonitoreihin kasvattaa entisestään vaatimusten suurta kokonaismäärää.

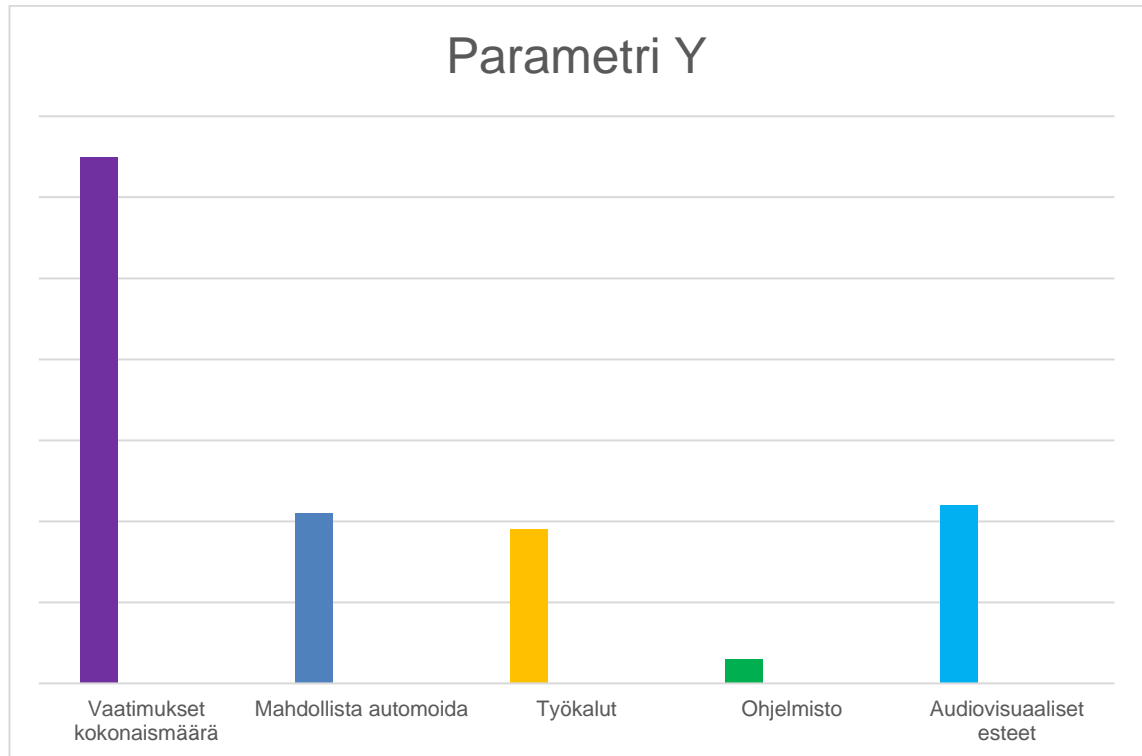


Kuva 8. Parametri X

Kuva 8 kuvastaa parametri X:n järjestelmätason vaatimuksien tuloksia analyysin jälkeen. Vaikka tilanne näyttää testiautomaation mahdollisuuksien puolesta hyvälle, pylväsdiagrammin tuloksissa on vääristynyt kuva, sillä testit edellyttävät erillisen laitteen kiinnittämistä monitoriin lokaalisti.

Parametrissa X audiovisuaaliset esteet estävät testiautomaation kehittämisen parametrille, sillä robotilla ei voida tulkita käyttöliittymältä tarkasteltavia ominaisuuksia ja aaltokäyriä eikä niiden piirtonopeutta.

Simulaattorin tuomat esteet puolestaan liittyvät parametrin kalibrointiin, ääriarvojen testaamiseen tai teknisiin hälytyksiin, jotka ovat tällä hetkellä mahdollista testata ainoastaan manuaalisesti.

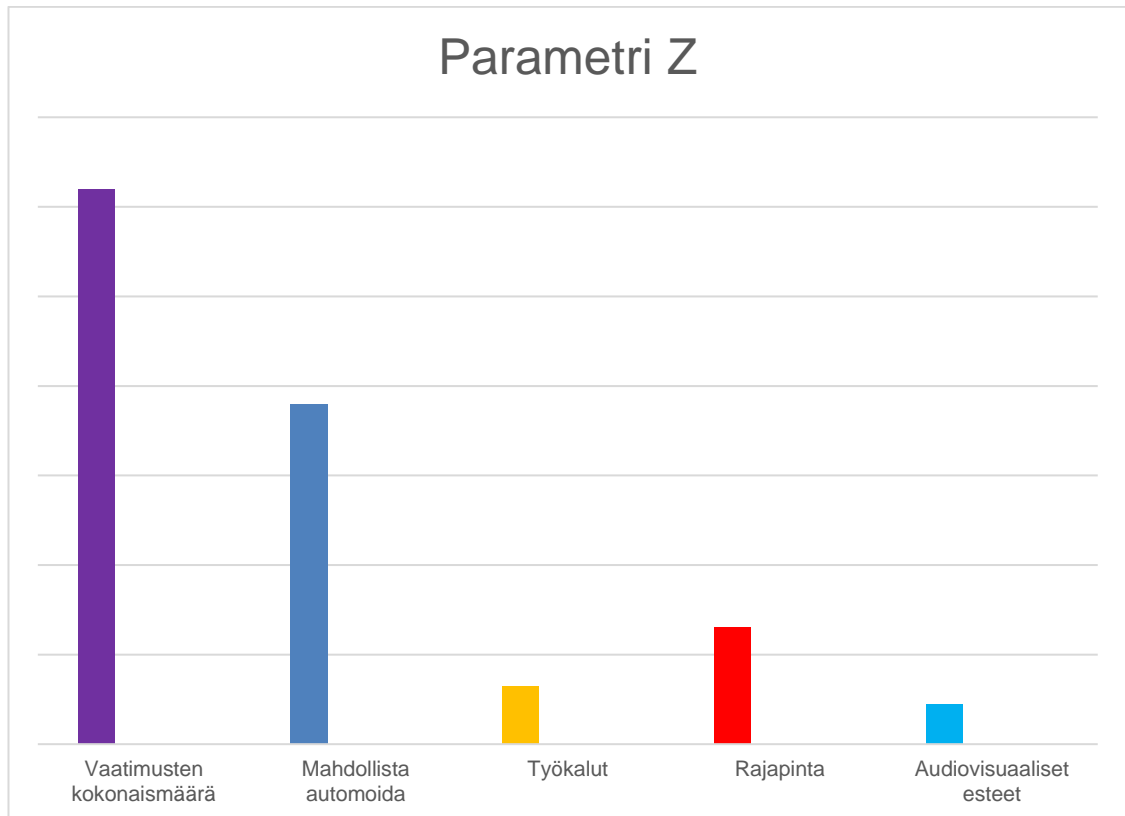


Kuva 9. Parametri Y

Kuva 9 näyttää, että suurin este vaatimusten testauksen automatisoinnille on audiovisuaaliset esteet ja käytettävissä olevat työkalut. Audiovisuaalisilla esteillä tarkoitetaan sitä, että vaatimus edellyttää tarkistamaan tiettyä näkymää tai näkymän tarkastelua potilasmonitorilta. Tätä ei tällä hetkellä voida robotilla automatisoida, aaltokäyrien dynaamisen luonteen vuoksi.

Työkaluilla puolestaan tarkoitetaan simulaattorin rajoitteita. Simulaattori ei kykene simuloimaan riittävän useita parametrien aaltokäyriä ja tekee näin ollen testin suorittamisesta jopa manuaalitestaajalle haastavan. Tämän lisäksi parametrin vaatimuksissa testataan teknisiä hälytyksiä ja ääriarvoja. Osa teknisistä hälytyksistä edellyttävät manuaalitestausta simulaattorin rajoitteiden vuoksi. Ääriarvojen testaaminen puolestaan haastaa simulaattoria siten, että

korkeiden tai matalien arvojen simulointi tasaisina lukuarvoina on hankalaa. Ohjelmiston tuomia esteitä ei voida tässä diagrammissa käsitellä.



Kuva 10. Parametri Z

Kuva 10 antaa vaikutelman, että parametrin Z vaatimuksille olisi pääluokittelun mukaan mahdollista kehittää testiautomaatiota. Tarkemmassa selvityksessä kuitenkin paljastui, että tähän pääluokkaan asetettuihin vaatimuksiin on vaikea kehittää testiautomaatiota tai ne eivät tuo lisäarvoa. Järjestelmätason vaatimuksia testataan parametrien kappaleissa askelmaisesti, ja osa vaatimuksista edellyttää tiettyä etenemistä niiden varmentamiseksi. Tämä tarkoittaa sitä, että vaikka vaatimuksille saataisiin testiautomaatiokattavuus, niin niitä ei voida manuaalitesteistä kuitenkaan poistaa.

Toinen haaste parametrin Z testiautomaation kattavuuden lisäämisessä on rajapinnan puuttuminen. Vaatimukset edellyttävät parametrin toiminnallisuutta ja

sen ominaisuuksia testataan ohjelmistolla, mistä puuttuu rajapinta, joka mahdollistaa tarvittavan vuorovaikutuksen järjestelmän ja robotin välillä.

Audiovisuaaliset esteet puolestaan taas liittyvät siihen, että robotilla ei voida tulkita käyttöliittymältä tarkasteltavia ominaisuuksia ja aaltokäyrien dynaamisuutta.

7.6 Analyysin luotettavuus

Projektissa haluttiin varmistaa potilasmonitorin vaatimusten oikeinluokittelu. Vaatimukset on määritetty eri parametreille, joista vastaavat eri tiimit. Tiimien välillä voi olla eroja, mitä simulaattoreita tai ohjelmistoja eri parametrit käyttävät. Analyysin tekeminen edellytti hyvää parametrien, vaatimusten, ohjelmistojen sekä simulaattoreiden tuntemista. Tämän vuoksi ennen vaatimusten varsinaista luokittelua koettiin edeltävyysehtona perehtyä analysoitavan parametrin toiminnallisuuksiin, käytettäviin simulaattoreihin, ohjelmistoihin, vaatimuksiin sekä niiden manuaalitestausten menetelmiin.

Epäselvät ja vaikeammat järjestelmätason vaatimukset kerättiin erilleen, jotka analysoitiin niistä vastaavien tiimien automaatio- ja verifikaatioinsinöörien kanssa.

Järjestelmätason vaatimusten luokittelussa huomioitiin hyvä dokumentointi, joka oli tärkeässä roolissa, mikä mahdollisti analyysin läpinäkyvyyden sekä sen tarkastettavuuden. Dokumentoinnin avulla pystytään tallentamaan vaatimuksen testiautomaation mahdollisuuden kannalta tärkeää tietoa myöhempää käyttöä varten.

7.7 Tulokset

Lopullisessa järjestelmätason vaatimusten analyysissä oli havaittavissa parametrien sekä vaatimusten välillä samankaltaisuuksia testiautomaation esteille. Vaatimukset oli jaettu viiteen eri pääluokkaan, joita ovat

audiovisuaaliset esteet, rajapinnan puuttuminen, ohjelmisto, työkalut ja ei estettä.

Audiovisuaaliset esteet testiautomaatiolle potilasmonitorien aaltokäyriin liittyvät haasteisiin, joita ilmenee visuaalisen tiedon, kuten aaltomuotojen ja muiden graafisten esitysten automaattisessa tunnistuksessa ja analysoinnissa. Haasteet johtuvat visuaalisen datan monimutkaisuudesta sekä niiden dynamisesta luonteesta, mikä tekee vaikeaksi luoda tarkkaa ja luotettavaa testiautomaatiota vaatimusten täyttämiseksi.

Rajapinnan puuttuminen testiautomaatiossa tarkoittaa, että ohjelmistolla ei ole suoraa pääsyä potilassimulaattorien tai testissä käytettävän ohjelmiston tuottamaan dataan tai ohjattavuuteen. Haaste ilmenee, kun potilasmonitorit ja käytettävät simulaattorit tai ohjelmistot eivät tarjoa ohjelmallisia rajapintoja ”API”, joiden kautta testiautomaatio voisi suoraan olla vuorovaikutuksessa laitteiden tai ohjelmiston kanssa. Seurauksena on se, että automatisoidun testauksen kehittäminen ja toteuttaminen voi muuttua monimutkaiseksi, koska testiskriptien on esimerkiksi vaikeampi simuloida käyttäjän toimia tai lukea laitteiden antamaa palautetta ilman standardoitua rajapintaa.

Ohjelmistosta johtuvat haasteet testiautomaatiolle voivat sisältää erilaisia tekijöitä. Tällaisia haasteita voivat olla esimerkiksi ohjelmiston monimutkaisuus, joka vaikeuttaa automaattisten testien luomista ja ylläpitoa, sekä ohjelmiston jatkuvat päivittyminen, mikä voi vaatia jatkuvia muutoksia testauskoodissa. Tämän lisäksi potilasmonitorien ominaisuudet kuten reaaliaikainen data ja erilaisten lääketieteellisten standardien noudattaminen asettavat ylimääräisiä vaatimuksia testiautomaatiolle.

Potilassimulaattorien käyttö testiautomaatiossa tuo mukanaan omia haasteita erityisesti ääriarvojen testaamisessa. Simulaattorit mahdollistavat keinotekoisen ympäristön, joka matkii potilaan fysiologisia toimintoja, mutta niiden tuottama data voi poiketa todellisesta kliinisestä datasta. Simulaattorit voivat rajoittaa testien kattavuutta, sillä ne eivät välttämättä pysty simuloimaan kaikkia

mahdollisia potilastilanteita tai monitorin käyttöolosuhteita. Tämä voi johtaa puutteelliseen testaukseen ja potentiaaliseen riskiin potilasturvallisuudessa ja sen vuoksi vaatimuksen testausmenetelmä edellyttää, että testi suoritetaan manuaalisesti.

Uusien potilassimulaattorien, ohjelmistojen sekä päivittyneiden työkalujen ansiosta analyysissä löydettiin myös vaatimuksia, jotka ovat mahdollista tai on jo osin katettu testiautomaatiolla. Vaikka osa vaatimuksista olisikin mahdollista osin tai kokonaan kattaa testiautomaatiolla, ei välttämättä tarkoita sitä, että niihin kannattaa testiautomaatiota kehittää. Järjestelmätason manuaalitesteissä testataan potilasmonitorin toiminnallisuuksia, jotka saattavat edellyttää tiettyä askelmaista etenemistä vaatimusten täyttämiseksi. Tämän vuoksi analysoidut järjestelmätason vaatimukset annettiin potilasmonitorin tuotekehityksessä oleville testiautomaatiosta vastaaville henkilöille, jotka voivat hyödyntää tuloksia nyt tai tulevaisuudessa.

7.8 Jatkokehitysmahdollisuudet

Pohdittaessa analyysin jatkokehitysmahdollisuuksia löytyi muutamia ajatuksia, jotka voidaan tuoda tässä työssä esiin.

Järjestelmätason vaatimusten tulee tukea selkeästi niiden testejä. Järjestelmän suunnittelussa ja vaatimusten määrittelyssä olisi hyvä huomioida asioita testaamisen näkökulmasta varmistamalla, että jokainen vaatimus on testattavissa käytettävissä olevilla työkaluilla. Järjestelmätason vaatimusten pohjalta luodaan monimutkaisia testejä, jotka koostuvat monista eri osista ja saattavat edellyttää monien simulaattorien käyttöä. Raskasta testausprosessia voitaisiin virtaviivaistaa siten, että se olisi helpommin ymmärrettävissä ja nopeammin suoritettavissa. Yksinkertaistamisella tarkoitetaan, että testeille tehtäisiin säännöllisin väliajoin katselmointi, missä arvioidaan ovatko jotkin testauksen vaiheet tarpeettomia. Tällä ei ainoastaan nopeuteta testausprosessia, vaan vähennetään testausvirheiden mahdollisuutta.

Toisena asiana projektissa huomattiin, että osa vaatimusten automatisoinnin esteenä oli potilassimulaattorin kykenemättömyys ylläpitää riittävän stabiileja arvoja. Mikäli GE HealthCare kehittäisi oman potilassimulaattorin, jolla voitaisiin jäljitellä potilaan fysiologisia toimintoja eri skenaarioissa mahdollistaisi monipuolisemman ja kattavan automaattisen testaamisen.

8 Yhteenveto

Insinööriyön tavoitteena oli tehdä analyysi potilasmonitorin järjestelmätason vaatimuksista. Suurimmat haasteet liittyivät, kuinka vaatimukset voidaan analysoida riittävän tehokkaasti käyttämättä niihin liikaa aikaa.

Työn lopputuotteena tuotettiin erillinen Excel-pohjainen tiedosto, jossa oli listattu vaatimukset eri parametreille, jaoteltuna selkeästi eri luokkiin. Tuloksissa selvisi, että eri parametrien järjestelmätason vaatimusten automaation esteinä oli samankaltaisuuksia. Tähän olisi hyvä kiinnittää tulevaisuudessa huomiota, jos testiautomaation kattavuutta halutaan tehokkailla tavoilla nostaa. Analyysin tarkoitus on palvella potilasmonitorin tuotekehityksestä vastaavia tiimejä, mikä auttaa heitä keskittymään tulevilla ohjelmistojulkaisuissa testiautomaation haasteisiin. Projekti edellytti työskentelyä eri sidosryhmien välillä, mikä oli sulavaa ja tehokasta.

Kokonaisuudessaan projekti onnistui, ja tämä lähestymistapa mahdollisti systemaattisen ja kohdennetun tavan käsitellä vaatimuksia, mikä tehosti tuotekehitysprosessia ja auttoi tunnistamaan mahdollisia riskejä ja kehityskohteita varhaisessa vaiheessa.

Lähteet

- 1 Antton Lounasheimo. 2022. Langattomat potilasmonitorit parantavat turvallisuutta – GE HealthCare innovoi ja investoi Suomessa. Verkkoaineisto. Teknologiateollisuus <<https://teknologiateollisuus.fi/fi/ajankohtaista/yritystarina/langattomat-potilasmonitorit-parantavat-turvallisuutta-ge-healthcare>>. Päivitetty 11.5.2022. Luettu 6.5.2023.
- 2 Lääkinnällisiin laitteisiin liittyvä lainsäädäntö. Verkkoaineisto. Fimea. <https://fimea.fi/laakinnalliset_laitteet/laakinnallisiin-laitteisiin-liittyva-lainsaadanto>. Luettu 12.12.2023.
- 3 Rüdiger, Kramme; Klaus-Peter, Hoffmann; Robert S, Pozos. 2011. Springer Handbook of Medical Technology. E-Kirja. Springer-Verlag Berlin Heidelberg.
- 4 Cañizares M. et al. 2015 Design and Implementation of the Software for Multi-parameter Patient's Monitor. In: Jaffray D. (eds) World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada. IFMBE Proceedings, vol 51. Springer, Cham. <https://doi.org/10.1007/978-3-319-19387-8_328>. Luettu 14.12.2023.
- 5 CARESCAPE Canvas monitoring platform. Verkkoaineisto. GE HealthCare. <<https://www.gehealthcare.com/products/patient-monitoring/patient-monitors/carescape-canvas>>. Luettu 14.12.2023.
- 6 Poncette AS, Spies C, Mosch L, et al. 2019. Clinical Requirements of Future Patient Monitoring in the Intensive Care Unit: Qualitative Study. JMIR Med Inform. 2019. Päivitetty 30.4.2019. Luettu 14.12.2023.
- 7 CARESCAPE Modular Monitors. Verkkoaineisto. GE HealthCare <<https://www.gehealthcare.co.uk/-/jssmedia/global/covid-19/lcs-pdf/qrg-carescape-modular-monitors-pdf.pdf?rev=-1>>. Luettu 7.12.2023.
- 8 Kasurinen, Jussi Pekka. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docendo.
- 9 Jose, B. 2021. Test automation: a manager's guide. BCS Learning and Development Ltd.
- 10 Laukkanen, Pekka. 2006. Data-Driven and Keyword-Driven Test Automation Frameworks. Diplomityö. Helsinki University of Technology, Science and Engineering. <<http://eliga.fi/Thesis-Pekka-Laukkanen.pdf>>. Luettu 6.12.2023.

- 11 Kumar, Pawan. 2023. Introduction to Robot Framework. Verkkoaineisto. BrowserStack. <<https://www.browserstack.com/guide/robot-framework-guide>>. Päivitetty 9.6.2023. Luettu 5.12.2023.
- 12 Robot Framework User Guide Version 7.0. Verkkoaineisto. Robot Framework. <<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>>. Luettu 29.11.2023.
- 13 DataDriven Tests. 2024. Verkkoaineisto. docs.robotframework.org. <https://docs.robotframework.org/docs/testcase_styles/datadriven>. Luettu 29.11.2023.
- 14 Mitä standardi tarkoittaa?. Verkkoaineisto. SFS. <<https://sfs.fi/standardeista/mika-on-standardi/>>. Luettu 16.12.2023.
- 15 Sectors, Health. 2024. Verkkoaineisto. ISO. <<https://www.iso.org/sectors/health>> Luettu 16.12.2023.
- 16 ISO 9001:2015, Quality management systems Requirements. 2015. Verkkoaineisto. ISO. <<https://www.iso.org/standard/62085.html>>. Luettu 16.12.2023.
- 17 ISO 9001:2015(en), Quality management systems – Requirements. Verkkoaineisto. ISO. <<https://www.iso.org/obp/ui/en/#iso:std:iso:9001:ed-5:v1:en>>. Luettu 16.12.2023.
- 18 SFS-EN ISO 9001. Laadunhallintajärjestelmät. Vaatimukset. 2015. Helsinki: Standardisoimisliitto.
- 19 SFS-EN ISO 13485. Terveysthuollon laitteet ja tarvikkeet. Laadunhallintajärjestelmät. Vaatimukset viranomaismääräyksiä varten. 2016. Helsinki: Suomen Standardisoimisliitto.
- 20 What we do. 2023. Verkkoaineisto. Food and Drug Administration. <<https://www.fda.gov/about-fda/what-we-do>>. Luettu: 17.12.2023.
- 21 Is it really 'FDA Approved'?. 2022. Verkkoaineisto. Food and Drug Administration. <<https://www.fda.gov/consumers/consumer-updates/it-really-fda-approved>>. Luettu: 17.12.2023.