

Toni Perkiö

OHJELMISTOTESTAUKSEN HALLINTA JA RAPORTOINTI

OHJELMISTOTESTAUKSEN HALLINTA JA RAPORTOINTI

Toni Perkiö
Opinnäytetyö
Kevät 2017
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, sulautetut järjestelmät

Tekijä: Toni Perkiö
Opinnäytetyön nimi: Ohjelmistotestauksen hallinta ja raportointi
Työn ohjaajat: Timo Vainio, Janne Koivuluoma
Työn valmistumislukukausi ja -vuosi: Kevät 2017
Sivumäärä: 35

Opinnäytetyön aiheena oli suunnitella rakenne testauksen hallintaan ja raportointiin käytettäväksi Bittium TACOM-projektissa. Tavoitteena oli luoda mahdollisimman hyvä rakenne testauksen hallinnalle ja tätä kautta mahdollistaa myös järkevä raportointi. Työn toteutusosio sisälsi myös työkalun kehittämisen, jolla testauksen hallintaan käytettävän työkalun kautta saatavasta tiedosta saadaan ylemmille raportointitasoille visuaalisesti helposti lähestyttävää tietoa.

Menetelminä tässä työssä hyödynnettiin testauksen hallintaan käytettävää SpiraTest Management -työkalua. SpiraTest Management -työkalusta saatavan datan prosessointiin ja visuaaliseen esitysmuotoon työssä kehitettiin MS Excelillä käytettävä työkalu. Tämän työkalun ohjelmointikielenä käytettiin Visual Basic -ohjelmointikieltä.

Työn aikana ylitettiin tehtävän alkuperäisessä tehtävänannossa määritellyt asiat. Alkuperäinen tehtävänanto ei sisältänyt testauksen raportointia helpottavan työkalun toteutusta. Työkalun jatkokehitys vastaamaan tämän projektityön tarpeita oli kuitenkin luontevaa sisällyttää tähän työhön.

Asiasanat: järjestelmätestaus, testauksen hallinta, raportointi, taktinen kommunikointi

ABSTRACT

Oulu University of Applied Sciences
Information technology

Author: Toni Perkiö

Title of thesis: Software test management and reporting

Supervisors: Timo Vainio, Janne Koivuluoma

Term and year when the thesis was submitted: Spring 2017

Pages: 35

Subject for this thesis was to design structure for test management and reporting to be used in Bittium TACOM project. Target was to create test management structure to enable understandable reporting for the project. This work included design and programming of an application to support test reporting activities. Purpose for the application was to enable more visual test reporting for test project personnel. Visual Basic programming language was used to achieve this goal.

During writing this thesis the initial definitions were exceeded. In initial definition test reporting tool was not included. However, this tool was designed and programmed during the work.

Keywords: system testing, test management, reporting, tactical communications

ALKULAUSE

Kiitän työntajani ja tämän työn toimeksiantajani Bittium Wireless Oy:tä toimeksiannosta tähän tehtävään. Testauksen teknisestä tuesta haluan kiittää Tradenomi Janne Koivuluomaa. Testauksen raportoinnin työkalun toteutukseen liittyen haluan antaa kiitoksen Insinööri Timo Savikoskelle.

Eriytinen kiitos tulee perheelleni ja puolisololleni DI Satu Vähänikkilälle. Perheellisen opintojen suoritus työn ohessa vaatii erityistä pitkäjänteisyyttä ja ajanhallintaa sekä opiskelijalta että lähimmäisiltä.

Oulussa 10.6.2017

Toni Perkiö

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	9
2 TAKTINEN KOMMUNIKOINTI	10
2.1 Taktinen kommunikointijärjestelmä	10
2.2 Bittium Tactical Wireless IP Network (TAC WIN)	10
2.2.1 Bittium TAC WIN Tactical Router	11
2.2.2 Bittium TAC WIN Radio Head I, III and IV	11
2.2.3 Ohjelmistoradiot	11
3 TESTAUKSEN HALLINTA	12
3.1 Testauksen hallinnan työkalut	13
3.1.1 HP Quality Center	13
3.1.2 Inflectra SpiraTest	13
3.2 Testitapaus	13
3.3 Testisetti	13
3.4 Testauksen raportointi	14
4 OHJELMISTOTESTAUKSEN JAOTTELU	15
4.1 Testaustyyppit	15
4.1.1 Toiminnalliset testit	15
4.1.2 Ei-toiminnalliset testit	16
4.2 Testaus tässä työssä	17
5 TÄMÄN TYÖN TAVOITTEET	18
6 TOTEUTUS	19
6.1 Testauksen rakenne	19
6.2 Testauksen raportointi	21
6.2.1 Käyttöliittymän esittely	21

6.2.2 Käyttöliittymän käyttö	22
6.2.3 Sähköpostiominaisuus	28
6.3 Yhteenveto toteutuksesta	31
7 YHTEENVETO	32
LÄHTEET	34

SANASTO

- ISTQB International Software Testing Qualifications Board. Järjestelmä, joka valvoo ja opastaa testauksen laadukkuutta
- MANET Mobile Ad-hoc Network. Langaton tiettyä tarkoitusta varten muodostuva verkko
- SDR Software Defined Radio, Ohjelmistoradio
- TACOM Tactical Communications. Yleisesti käytettävä termi taktisille kommunikointijärjestelmille, joita käytetään esimerkiksi puolustusvoimien tarpeisiin
- TAC WIN Bittium Tactical Wireless IP Network. Bittiumin kehittämä järjestelmä taktisen kommunikaation tarpeisiin

1 JOHDANTO

Ohjelmistojen kehittämisessä testaus on merkittävässä asemassa varmistamassa ohjelmiston haluttu toiminnallisuus. Yleisesti ottaen täysin virheetöntä ohjelmistoa on varsin mahdotonta tuottaa. Tästä syystä eri tasoilla tapahtuva testaus laitteistolle on olennaista.

Tuotekehitysprojektissa testauksen hallinta ja raportointi ovat itse testauksen lisäksi tärkeitä. Hyvin toteutetulla raportoinnilla projektin eri osapuolet pysyvät tietoisina ohjelmiston kypsyydestä sekä kullakin hetkellä olevista vioista ja ongelmakohtista helposti. Tässä työssä keskitytään ohjelmistotestauksen hallintaan ja raportointiin. Koska työ on tehty Puolustusvoimien tarpeisiin kehitetyn laitteiston testauksen hallintaan, työ esittelee myös sotilaallisessa käytössä olevien taktisten kommunikointijärjestelmien peruseriaatteita.

Projektin lähtökohtana on tilanne, jossa systemaattinen testauksen hallinta ja raportointi puuttuvat lähestulkoon kokonaan. Näin ollen projektin sisällä saatikka sitten asiakkaalle näkyvyys testatuista asioista ja testauksen tuloksista esitettävässä muodossa puuttuu. Raportoinnin tärkeys on korostumassa jatkossa tuotteen viennin myötä kansainvälisille markkinoille.

Raportointiin ja testauksen hallintaan on olemassa monia eri työkaluja. Tässä työssä keskitytään rakentamaan testauksen hallinta käyttäen hyväksi Inflectran tarjoamaa SpiraTest Management -työkalua. Tämän lisäksi vaatimusten ja vikojen hallintaan käytetään Atlassian-yhtiön tarjoamaa JIRA-työkalua.

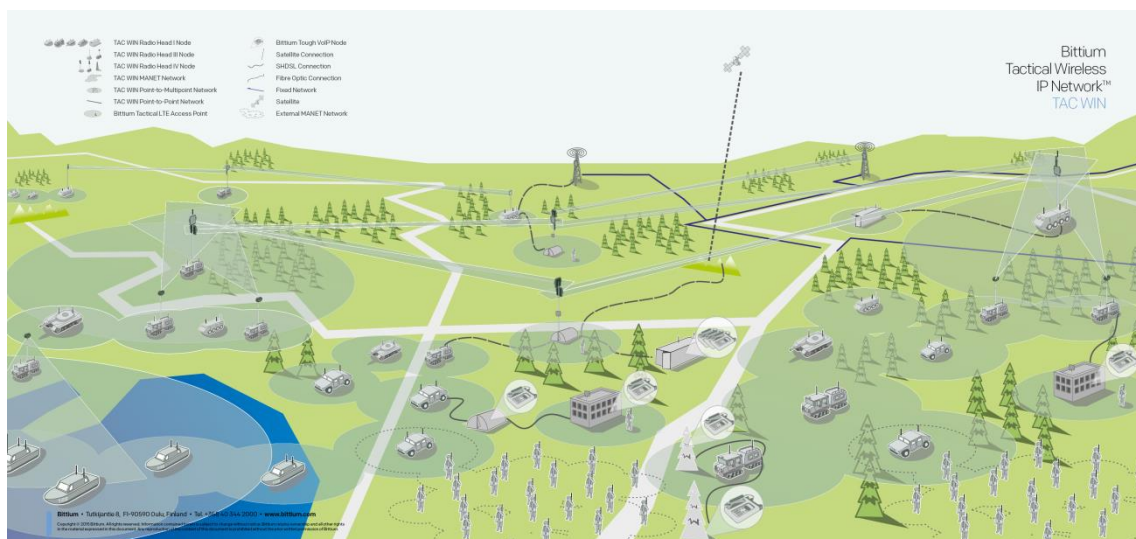
2 TAKTINEN KOMMUNIKOINTI

2.1 Taktinen kommunikointijärjestelmä

Taktinen kommunikointijärjestelmä (engl. Tactical communications system) on kommunikointijärjestelmä, jota käytetään taktisten sotilasjoukkojen toiminnan tukemiseen. Järjestelmä on luotu täyttämään taktisten tilanteiden ja ympäristön muuttuvat vaatimukset. Järjestelmä tarjoaa liikkuville käyttäjille suojatun yhteyden eri liikennetyypeille, kuten ääni, data ja video. Järjestelmän toimeenpanoai-
ka tulee olla erittäin lyhyt.

2.2 Bittium Tactical Wireless IP Network (TAC WIN)

Taistelukentän digitalisointi ja monenlaisia toimintoja sisältävät säätötekniiset sovellukset, anturit ja reaaliaikainen tilannekuva ovat asettaneet kasvavan tarpeen suurille datanopeuksille ja matkaviestinverkoille taktisessa käytössä. Paras liitettävyyden ja suorituskyky saavutetaan täysin IP-yhteensopivilla verkoilla. Ohjelmistoradioon (engl. Software Defined Radio, SDR) perustuvalla laitteistolla taistelujoukot voivat luoda korkean tiedonsiirtokyvyn langattomia IP-verkkoja tukemaan C2-luokan datansiirtoa sotilaallisissa operaatioissa. (1.)



KUVA 1. Bittium Tactical Wireless IP Network (1)

2.2.1 Bittium TAC WIN Tactical Router

Bittium TAC WIN Tactical Router on järjestelmän reititys- ja radiokantataajuus-prosessointiyksikkö, jota käytetään Bittium TAC WIN -järjestelmän aaltomuodossa. Yksikkö tarjoaa luotettavan alustan MANET-verkkojen kymmenien solmujen Point-to-Multipoint- ja radiolinkkiyhteyksille. (1.)

2.2.2 Bittium TAC WIN Radio Head I, III and IV

Bittium TAC WIN Radio Head yksiköitä käytetään muodostamaan MANET, joka on optimaalinen ratkaisu suhteellisen pienellä alueella toimiville joukoille ja joiden vaatimuksena on liikuteltava ja joustava verkko. Bittium TAC WIN MANET tarjoaa luotettavan ja itsenäisesti korjaantuvan alustan ääni ja viesti palveluille. Yksiköitä on eri taajuusalueille ja kaistanleveyksille. Ne tarjoavat korkean suorituskyvyn yhteyksiä johtokeskukseen tai runkoverkkoon. (1.)

2.2.3 Ohjelmistoradiot

Ohjelmistoradio on lähetin-vastaanotinjärjestelmä, jonka ominaisuuksista suurta osaa voidaan määrittellä ohjelmallisesti. Radiot luokitellaan ohjelmitavuuden perusteella. Kansainvälinen ohjelmistoradion kehitykseen keskittynyt SDR Forum -organisaatio jakaa radiot viiteen luokkaan seuraavasti:

1. HW-radio (hardware radio)
2. Ohjelmistolla kontrolloitava radio (software controlled radio)
3. Ohjelmistolla määritetty radio (software defined radio)
4. Ideaalinen ohjelmistoradio (ideal software radio)
5. Lopullinen ohjelmistoradio (ultimate software radio). (2.)

Näistä luokista ohjelmistolla määritetty radio vastaa Bittium TAC WIN -järjestelmää.

3 TESTAUKSEN HALLINTA

Testauksen hallinnalla tarkoitetaan yleisesti ottaen jonkin tuotteen, ohjelmiston tai ohjelmistokokonaisuuden testauksen hallintaa. Yleisesti ajatellen testauksen hallinta sisältää asioita kuten seuranta, korjaustoimenpiteet, eskalointi ja ongelmanratkaisu, laadun arviointi ja viestintä sidosryhmille. Lisäksi voidaan ajatella suunnitteluun sisältyviä asioita kuten tavoitteet ja riskit, aikataulut, lähestymistapa ja resursointi. (4, s. 42.) Testaus itsessään voidaan jakaa aihealueisiin kuten valmistelu, määrittely, suoritus ja lopetus. (4, s. 43–46.)

Testauksen hallintaan on eri ohjelmistontuottaja yrityksillä tarjota omia työkalujaan. Mikäli tuotetta toteuttava yhtiö tarjoaa palvelujaan monille eri yhtiöille, testauksen hallintaan soveltuva työkalu olisi hyvä olla joustava erityyppisten tuotteiden testauksen hallintaan. Tässä yhteydessä hallintaan on yrityksessä päädytty käyttämään Inflectran tarjoamaa SpiraTest Management -työkalua.

Ohjelmistotestauksen hallintaan on käytettävissä monia eri työkaluja. Näistä esimerkkinä mainittakoon Hewlett Packardin Quality Center sekä tässä työssä käytettävä Inflectran SpiraTest. Koska testauksen hallintaan käytettävän työkalun tarkoituksena on mahdollistaa ohjelmistokokonaisuuden laadun valvonta, sen täytyy sisältää eri ominaisuuksia, kuten (5, s. 5)

- testitapausten määrittely, luokittelu, priorisointi ja muu hallinta
- testien suorituksen tilan hallinta
- mahdollinen linkitys vaatimusten hallintaan
- vikojen hallinta
- testitulosten raportointi
- testiautomaation hallinta.

3.1 Testauksen hallinnan työkalut

3.1.1 HP Quality Center

HP:n tarjoama Quality Center on laajalti käytetty ohjelmisto testauksen hallintaan. (6.) Ohjelmistosta on saatavilla eri versioita riippuen tarpeesta:

- HP Quality Center Community Edition
- HP Quality Center Express Edition
- HP Quality Center Enterprise Edition

Näistä Community ja Express Edition on tarkoitettu perustason ohjelmistojen laadun varmistamiseen. (6.) Enterprise Edition on taas ohjelmisto organisaatioille, jotka hallitsevat keski ja suurikokoisia ohjelmisto julkaisuja.

3.1.2 Inflectra SpiraTest

Inflectran tarjoama SpiraTest on HP:n Quality Centeriin verrattuna ominaisuuksiltaan hieman kevyempi. (7.) Esimerkiksi raportoinnin eri mahdollisuudet ovat pelkistetympiä. Tästä syystä se on käyttölisenssimaksuiltaan Quality Centeriin verrattuna myös halvempi. Ohjelmisto soveltuu kuitenkin myös keski – ja suurikokoisten ohjelmistojulkaisujen hallintaan.

3.2 Testitapaus

Testitapausten hallinta on osa ohjelmistokokonaisuuksien hallintaa. Testitapauksella pyritään varmistamaan tuotteen toiminnallisuus halutulla tavalla. Eri-tyyppiset testitapaukset on hyvä jaotella loogisten kokonaisuuksien perusteella ts. esimerkiksi aihealueittain. Tässä työssä keskitytään Bittium TAC WIN -järjestelmän testitapausten hallintaan.

3.3 Testisetti

Ohjelmiston testaus sisältää monesti erityyppistä testausta. Tässä työssä keskitytään tuotteen järjestelmätason testaukseen ja testisetit luodaan vastaamaan tämän vaiheen tarpeita:

- 1) Päivittäin suoritettava testaus. Tämän testauksen tarkoituksena on varmistaa päivittäin tuotavan julkaisun kypsyys ja todeta, ettei julkaisuun ole erinäisistä syistä, kuten johonkin korjaukseen liittyen tai uuden toiminnallisuuden tuonnin yhteydessä ilmaantunut uusia vikoja (ns. regressiotestaus). Tyypillisesti tässä testauksessa suoritetaan jo aiemmin luotuja ns. vanhoja testitapauksia.
- 2) Uusien toiminnallisuuksien testaus. Testauksessa keskitytään jonkin tietyn uuden toiminnallisuuden testaukseen.
- 3) Ohjelmistojulkaisun kokonaisuuden testaus ns. release testaus. Testaus keskittyy ohjelmiston laajempaan testaukseen ennen asiakastoimitusta.

Yleisesti ottaen testisetillä pyritään helpottamaan testauksen hallintaa ja raportointia.

3.4 Testauksen raportointi

Kun ohjelmistojulkaisulle on luotu tarkoituksenmukainen testisetti, sisältäen testitapaukset, joilla ohjelmiston toiminnallisuus ja kypsyys voidaan osoittaa, saadaan testauksen hallintatyökalusta pienellä vaivalla myös tilannetta vastaava raportti. Sisällöltään erilaisia raportteja saadaan tarpeesta riippuen. Jos esimerkiksi halutaan raportoida vain prosentuaalisella tasolla, paljonko testeistä on ajettu ja mikä on hyväksytysti ajettujen taso, voidaan tätä raportoida helposti. Toisaalta testauksen hallintatyökalusta saadaan tarvittaessa tarkempaakin raporttia, johon voidaan sisällyttää testitapausten testistepit, historiatietoa, testiajot ja niin edelleen.

4 OHJELMISTOTESTAUKSEN JAOTTELU

Ohjelmistotestaus voidaan jaotella kahteen eri alueeseen, toiminnalliseen ja ei-toiminnalliseen testaukseen. Toiminnallinen testaus käsittää ohjelmiston tarjoamien toimintojen testaamista yksittäisinä toimintoina, toimintoryhminä sekä yhdistettynä ohjelmiston aineiston kanssa. Ei-toiminnallinen testaus kohdistuu taas nk. ei-toiminnallisiin ominaisuuksiin, jotka vaikuttavat ohjelmistoon liittyvään kokonaislaatukokemukseen, mutta eivät ole liitettävissä suoraan toimintoon tai toimintoryhmään ohjelmistossa, kuten käytettävyys, suorituskyky ja luotettavuus. (8, s. 57.)

Toisaalta testaus voidaan määritellä myös jakamalla se staattiseen ja dynaamiseen testaukseen. Näistä staattisessa testauksessa ei ajeta koodia, kun taas dynaamisessa ajetaan. Staattisella tarkoitetaan tietokoneeseen perustumattoman testaamisen prosessia, kuten katselmoinnit, tarkastukset ja analyysit. Dynaamisella taas tarkoitetaan tietokoneeseen perustuvaa perinteistä testausta, joka perustuu ajatteluun, että ohjelmat on kirjoitettu vain koneella ajamista varten eikä ihmisten luettavaksi. (8, s. 58.)

4.1 Testaustyypit

Ohjelmistotestaus voidaan jaotella myös eri testaustyyppihin. Testaustyyppi on ryhmä testausaktiviteettejä, joilla on yhteisiä ominaisuuksia. Näiden ominaisuuksien perusteella testaustyypit voidaan tunnistaa omana luokkana ja ryhmitellä arvioimaan yhtä tai useampaa toisiinsa liittyvää laatuominaisuutta. (4, s. 27.) Nämä testaustyypit voidaan jakaa aiemmin mainittuihin toiminnallisiin ja ei-toiminnallisiin testeihin.

4.1.1 Toiminnalliset testit

Toiminnallisten testien testaustyyppinä ovat (4, s. 28; 10)

- toiminnallisuustestaus (functionality testing, feature testing)
- yhtäaikaaisuustestaus (concurrency testing)

- asennustestaus (installation testing)
- alustetestaus (platform testing)
- aloitustestaus (build verification testing, smoke testing)
- konfiguraatiotestaus (configuration testing)
- yhteensopivuustestaus (compatibility testing)
- rinnakkaistestaus (end-to-end testing)
- rajapintatestaus (interface testing)
- poikkeustilannetestaus (recovery testing)
- lokalisoititestaus (localization testing)
- dokumentaation testaus (documentation testing)
- aineiston laadun testaus (data quality testing)
- alfatestaus (alpha testing)
- betatestaus (beta testing)
- muuntotestaus (conversion testing)
- tuotantotestaus (production testing, operational testing)
- standardien testaus (standards testing).

4.1.2 Ei-toiminnalliset testit

Ei-toiminnallisten testien testaustyyppejä ovat (4, s. 29; 10)

- luotettavuustestaus (reliability testing)
- suorituskykytestaus (performance testing)
- kuormitustestaus (load testing)
- rasitustestaus (stress testing)
- paljoustestaus (volume testing)
- kestävyystestaus (endurance testing)
- tietoturvatestaus (security testing)
- käyttöturvallisuuden testaus (safety testing)
- käytettävyydestestaus (usability testing)
- esteettömyystestaus (accessibility testing)
- palautettavuustestaus (recoverability testing)

- tuettavuustestaus (supportability testing)
- ylläpidettävyytestaus (maintainability testing)
- siirrettävyytestaus (portability testing)
- koodin laadun testaus (code quality testing).

4.2 Testaus tässä työssä

Tämän työn jakautuessa kahteen osioon testaus käsitti sekä varsinaisen tuotteen testausta että toteutusvaiheessa toteutetun testauksen hallintatyökalun testausta. Tuotteen testaus käsitti sekä toiminnallista että ei-toiminnallista testausta. Tuotteen testauksen jaottelu on esitelty kappaleessa 6.1. Testauksen hallintatyökalun testaus tapahtui olemassa olevassa testausympäristössä, josta testausdata saatiin. Itse työkalun koodin toimivuus oli mahdollista testata Spira-Test Management -työkalusta saatavalla datalla.

5 TÄMÄN TYÖN TAVOITTEET

Tuotteen tai ohjelmistokokonaisuuden testaus on monesti laaja-alainen tehtävä, joka käsittää eri tasoilla tapahtuvaa testausta. Tämän työn tavoitteena on jäsentää järjestelmätasolla testattavat asiat Bittium TAC WIN -tuotteelle sekä luoda raportointijärjestelmä kattamaan sekä tuotekehitys että asiakastoimitusvaiheen testauksen raportointi. Testauksen tulee jäsentyä loogisiin kokonaisuuksiin, jolloin eri osa-alueiden hallinta ja näkyvyys on taattu. Tavoitteena on ensisijaisesti luoda testauksen rakenne, jota voidaan käyttää hyväksi tämän työn kohteena olevalle tuotteelle, mutta jota voidaan hyödyntää myös muille taktisten kommunikointijärjestelmien tuoteperheen tuotteille.

Testauksen raportointi on tärkeä osa tuotteen kypsyden esittämiseen. Järjestelmällisellä raportoinnilla pysytään ajan tasalla olemassa olevista ongelmista ja saadaan näkyvyys tuotekehityksen etenemään. Vaatimuksena raportoinnin toteuttamiselle on, että tuotteelle on olemassa vaatimuksia vastaavat testitapaukset. Ilman olemassa olevia testitapauksia myöskään järkevää raportointia ei voida toteuttaa. Raportoinnin osalta tämän työn tavoitteena on luoda sekä tuotekehitystä että asiakasta palvelevat raportointitavat. Koska tuotekehityksen raportoinnin vaatimukset poikkeavat monesti asiakkaan vaatimuksista, näille on tarpeen luoda toisistaan hieman poikkeavat raportit. Tämä tarkoittaa sitä, että tuotekehitysvaiheessa on tarpeen seurata tarkemmalla tasolla ts. testitapaus tasolla toiminnallisuuksien toimivuuksia, kun taas asiakkaalle toimitettava raportointi voi olla yleisemmällä tasolla.

6 TOTEUTUS

Testauksen raportointiin on markkinoilla olemassa monia eri työkaluja. Asiakas on yhtiötasolla valinnut testauksen hallintaan Inflectra Corporationin tarjoaman SpiraTest Management -työkalun. Työkalun valintaan on vaikuttanut mm. sen sopivuus testitapausten ja testisettien hallintaan. Valintaan on todennäköisesti vaikuttanut myös hinta, joka on halvempi isompien yhtiöiden tarjoamiin työkaluihin verrattuna, kuten esimerkiksi Hewlett Packardin Quality Center.

SpiraTest Management -työkalulla voidaan hallita vaatimuksia ja vaatimukset voidaan linkittää testitapauksiin, jolloin saadaan näkymä testikattavuudesta. Työkalua voidaan käyttää myös vikojen hallintaan. Vikojen sekä vaatimusten hallintaan asiakas on kuitenkin tässä tapauksessa päättänyt käyttämään Atlassian-yhtiön tarjoamaa Jira Softwarea. Tähän syynä on Jiran tarjoama parempi käytettävyys näiden asioiden hallintaan. Jira soveltuu myös ohjelmistotoimitusten hallintaan ja näin ollen onkin järkevää, että vikojen hallinta on samassa työkalussa, jolla hallinnoidaan ohjelmistotoimituksia. Näiden kahden työkalun valintaan on vaikuttanut myös mahdollisuus integroida ne toimimaan yhteen. Tämä tapahtuu Jiraan saatavilla olevalla SpiraTest-liitännäisellä, joka mahdollistaa tietojen kuten vaatimusten ja vikaraporttien synkronoinnin työkalujen välillä.

6.1 Testauksen rakenne

Kuten jo edellä mainittiin, on testaus järkevää jakaa tuotteelle määriteltyihin loogisiin kokonaisuuksiin. Kokonaisuudet voivat olla erilaisia ja poiketa riippuen testattavasta tuotteesta. Lähtökohtana monesti käytetään tuotteen päävaatimuksia ja niiden alla olevia tarkempia vaatimuksia, jolloin testaus voidaan rakentaa suoraan vastaamaan niitä. Toisaalta vaatimuksissa ei aina ole huomioitu kaikkia mahdollisia käyttäjätapauksia ja nämä tulevat esiin vasta testausta suunniteltaessa, joskus myös vasta testausvaiheessa. Joskus myös tuotevaatimukset eivät ole kovin loogisesti järjesteltyjä ja tästä syystä joudutaan miettimään rakenne enemmänkin testausta vaativien asioiden ja asiakasnäkökulman kautta. Yleisesti ottaen testauksen rakenne päätasolla voidaan kuitenkin suun-

nitella jo varsin varhaisessa vaiheessa ts. vaatimusmäärittelyjen loppuvaiheessa. Tässä työssä keskitytään taktisen kommunikointijärjestelmän tuotteen ja sen eri toiminnallisuuden testaukseen. Tästä syystä testialueet määräytyvät pitkälti sen mukaan.

Kuten kappaleessa 4 mainittiin, ohjelmistotestaus voidaan jakaa eri testaus-tyyppeihin. Kaikkia testaus-tyyppejä ei suinkaan ole tarkoitus käyttää vaan näitä voidaan käyttää kuhunkin projektiin ja tuotteeseen soveltuvien osien. Testaus-tyypit voivat olla myös osaksi päällekkäisiä. Tästä syystä tässä työssä päädyttiin osaksi käyttämään määriteltyjä testaus-tyyppejä ja osaksi yhdistelemään niitä suurempiin kokonaisuuksiin. Testaus jaoteltiin seuraavasti viiden eri testaus-tyypin alle:

- Functionality and feature testing
 - Tuotteen eri toimintojen ja toiminnallisuuden testaus
- Configuration and usability testing
 - Tuotteen konfigurointi, käyttöliittymän ja käytettävyyden testaus
 - Eri verkkotopologioiden testaus
 - Tuotteen sisäisen raportoinnin testaus (logitus)
 - Käynnistyminen ja resetointi
 - Ohjelmiston lataus ja päivitys
- Robustness testing
 - Tuotteen kuormitustestaus lyhyellä ja pitkällä aikavälillä (load, stress, endurance)
 - Tuotteen stabiilisuus eri tilanteissa
 - Tuotteen kyky palautua eri tilanteista
- Performance testing
 - Tuotteen suorituskykytestaus käyttäen eri tiedonsiirtoprotokollia
 - Eri radioyksiköillä suoritettavat suorituskykymittaukset
 - Radiorajapinnan mittaukset eri modulaatioilla

- Security testing
 - Tietoturvan ja salauksen testaus radorajapinnassa sekä langallisissa yhteyksissä
 - Käyttäjän autentikointi
 - Sertifikaattien testaus.

6.2 Testauksen raportointi

Aiemmin tässä työssä tehtyjen testaustyyppien ja yleisen testirakenteen perusteella voitiin edetä seuraavaan vaiheeseen eli testauksen raportointiin. Kun rakenne oli selvillä, voitiin testitapaukset jaotella määriteltyjen testaustyyppien alle, ja tätä kautta alkaa suunnitella järkevä näkymä testauksen tilasta.

Raportointidatan generointiin tässä työssä käytettiin edellä esiteltyä SpiraTest Management -työkalua. Koska SpiraTestin raportointiominaisuudet ovat rajalliset eivätkä mahdollista esimerkiksi tietyn testisetin järkevää seuranta, tässä työssä käytettiin hyväksi MS Excel - ja Visual Basic-ohjelmistoja. Tällä mahdollistettiin Spirasta saatavaan dataan perustuvien graafisten näkymien luonti halutuista asioista, kuten projektin eri tuotteista ja tuotteiden eri toiminnallisuuksien ja testauksen eri alueiden testauksen tilanteesta.

6.2.1 Käyttöliittymän esittely

Testauksen raportoinnin käyttöliittymän etusivu sisältää eri painikkeita, joista tärkein on ylälaudassa sijaitseva projektin testi raportin päivittämiseen käytettävä painike. Tämän painikkeen alapuolella on informatiivinen ikkuna, josta selviää raportointiin käytettävään lähteeseen ja tallennuskohteeseen liittyvää tietoa. Työkalu sisältää myös mahdollisuuden lähettää tuotetut diagrammit testitilanteista ennalta määritellyille henkilöille. Näin, että tietyt henkilöt saavat vain heidän tuotteeseen liittyvän testitilanteen sähköpostiinsa. Vaihtoehtona on myös lähettää sähköpostilla kaikkien tuotteiden statukset ennalta määritellyille henkilöille yhdellä painikkeen painalluksella. (Kuva 2.)

Start → Update Project Test report

Path for Spira source files
C:\Data\SpiraExport\

Source file for Test Set data
Project_Report_TestSet.xls

Project Test Set Detailed Report.html

Time stamp for Test Set source file
Spira: 10.02.2017 12:53:11

Test set detailed report: 10.02.2017 12:55:49

Weekly Export saved here
C:\Data\SpiraExport\TestStatus_2017_wk12.xlsx

Last update
Updated 20.03.2017 @ 10:25:54 | Week: 12

Weekly test report

Passed: 7000, Failed: 1422, Not Run: 2960

Status report in email using following buttons

Send PRODUCT A DAILY TEST charts | Send PRODUCT A charts | Send PRODUCT B charts | Send PRODUCT C charts | Send PRODUCT D charts | Send TEST 1 charts | Send TEST 2 charts | Send ALL charts

Feature Specific test reports

Send Feat1 charts | Send Feat2 charts | Send Feat3 charts | Send Feat4 charts

Newest data	
All	(All) Not Run: 107, Pass: 157, Fail: 57, Block: 0 (PR: 73.4% RR: 66.7%)
PRODUCT A DAILY 1	(PRODUCT A Dsily - All) Not Run: 42, Pass: 56, Fail: 53, Block: 0 (PR: 51.4% RR: 72.2%)
PRODUCT A	(Customer X) Not Run: 64, Pass: 50, Fail: 4, Block: 0 (PR: 32.6% RR: 45.8%)
PRODUCT B	(PRODUCT B) Not Run: 1, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
PRODUCT C	(PRODUCT C) Not Run: 0, Pass: 51, Fail: 0, Block: 0 (PR: 100% RR: 100%)
PRODUCT D	(PRODUCT D (All)) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 1	(TEST 1) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 2	(TEST 2 TA) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
Updated 20.03.2017 @ 10:25:54 Week: 12	

All	(All) Not Run: 107, Pass: 157, Fail: 57, Block: 0 (PR: 73.4% RR: 66.7%)
PRODUCT A DAILY 1	(PRODUCT A Dsily - All) Not Run: 42, Pass: 56, Fail: 53, Block: 0 (PR: 51.4% RR: 72.2%)
PRODUCT A	(Customer X) Not Run: 64, Pass: 50, Fail: 4, Block: 0 (PR: 32.6% RR: 45.8%)
PRODUCT B	(PRODUCT B) Not Run: 1, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
PRODUCT C	(PRODUCT C) Not Run: 0, Pass: 51, Fail: 0, Block: 0 (PR: 100% RR: 100%)
PRODUCT D	(PRODUCT D (All)) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 1	(TEST 1) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 2	(TEST 2 TA) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
Updated 10.02.2017 @ 13:37:30 Week: 06	

All	(All) Not Run: 107, Pass: 157, Fail: 57, Block: 0 (PR: 73.4% RR: 66.7%)
PRODUCT A DAILY 1	(PRODUCT A Dsily - All) Not Run: 42, Pass: 56, Fail: 53, Block: 0 (PR: 51.4% RR: 72.2%)
PRODUCT A	(Customer X) Not Run: 64, Pass: 50, Fail: 4, Block: 0 (PR: 32.6% RR: 45.8%)
PRODUCT B	(PRODUCT B) Not Run: 1, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
PRODUCT C	(PRODUCT C) Not Run: 0, Pass: 51, Fail: 0, Block: 0 (PR: 100% RR: 100%)
PRODUCT D	(PRODUCT D (All)) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 1	(TEST 1) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 2	(TEST 2 TA) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
Updated 10.02.2017 @ 13:36:03 Week: 06	

All	(All) Not Run: 107, Pass: 157, Fail: 57, Block: 0 (PR: 73.4% RR: 66.7%)
PRODUCT A DAILY 1	(PRODUCT A Dsily - All) Not Run: 42, Pass: 56, Fail: 53, Block: 0 (PR: 51.4% RR: 72.2%)
PRODUCT A	(Customer X) Not Run: 64, Pass: 50, Fail: 4, Block: 0 (PR: 32.6% RR: 45.8%)
PRODUCT B	(PRODUCT B) Not Run: 1, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
PRODUCT C	(PRODUCT C) Not Run: 0, Pass: 51, Fail: 0, Block: 0 (PR: 100% RR: 100%)
PRODUCT D	(PRODUCT D (All)) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 1	(TEST 1) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 2	(TEST 2 TA) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
Updated 10.02.2017 @ 13:16:14 Week: 06	

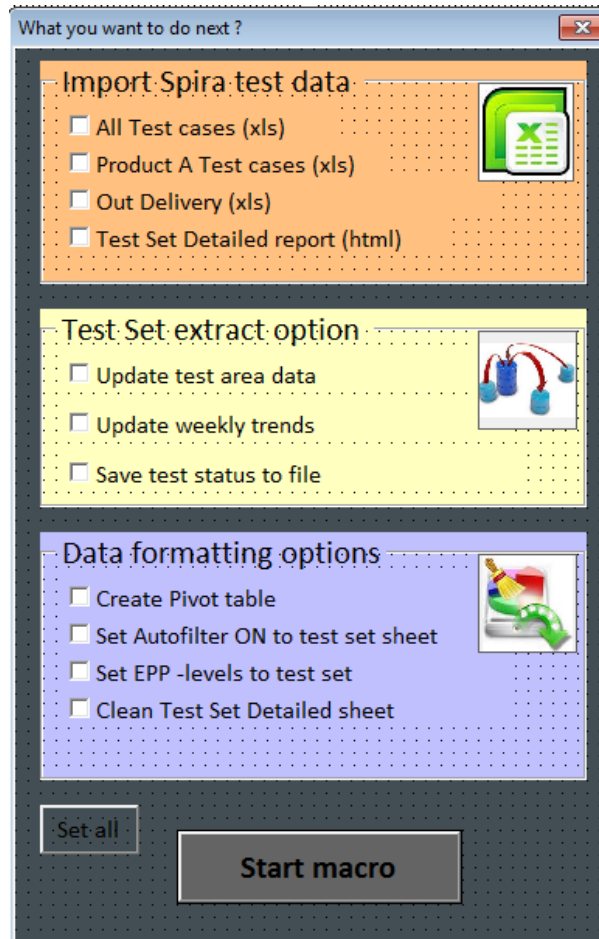
All	(All) Not Run: 107, Pass: 157, Fail: 57, Block: 0 (PR: 73.4% RR: 66.7%)
PRODUCT A DAILY 1	(PRODUCT A Dsily - All) Not Run: 42, Pass: 56, Fail: 53, Block: 0 (PR: 51.4% RR: 72.2%)
PRODUCT A	(Customer X) Not Run: 64, Pass: 50, Fail: 4, Block: 0 (PR: 32.6% RR: 45.8%)
PRODUCT B	(PRODUCT B) Not Run: 1, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
PRODUCT C	(PRODUCT C) Not Run: 0, Pass: 51, Fail: 0, Block: 0 (PR: 100% RR: 100%)
PRODUCT D	(PRODUCT D (All)) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 1	(TEST 1) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
TEST 2	(TEST 2 TA) Not Run: 0, Pass: 0, Fail: 0, Block: 0 (Tot: 0 <-> Pass %: 0)
Updated 10.02.2017 @ 13:09:06 Week: 06	

KUVA 2. Testiraportointi työkalun käyttöliittymän pääsivu

6.2.2 Käyttöliittymän käyttö

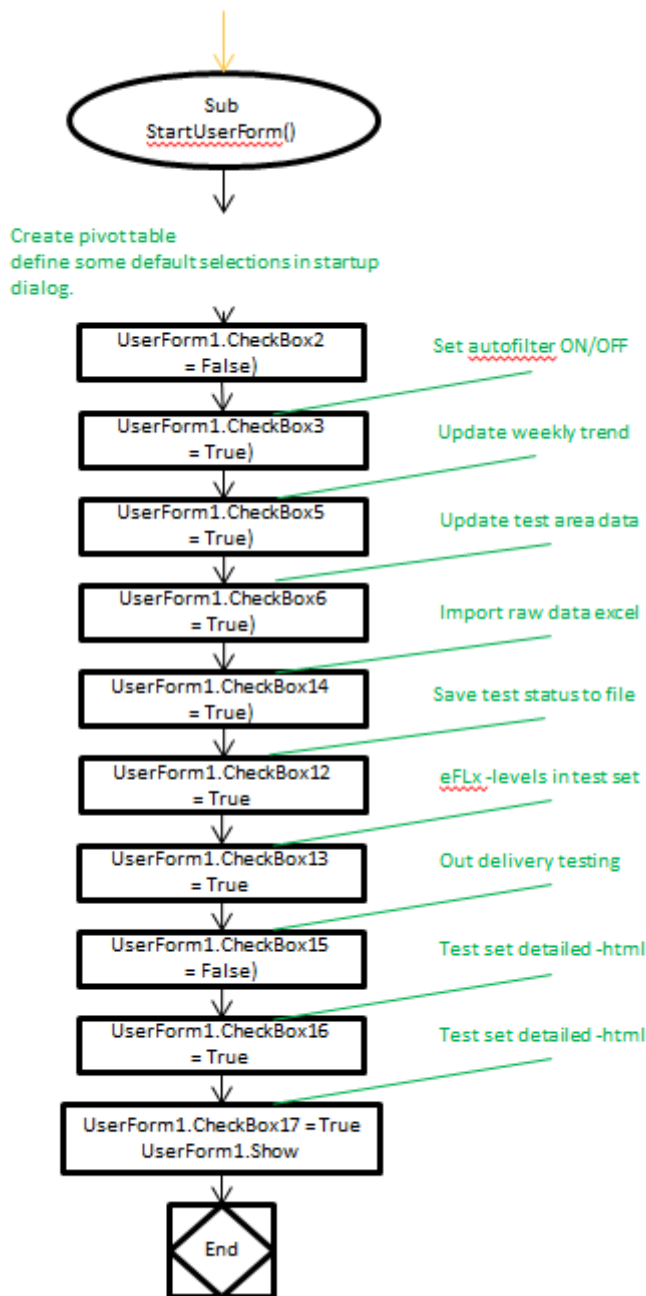
Excel-työkalulla luotavan graafisen testitieteen mahdollistamiseksi SpiraTest Management -työkalusta on ensin haettava kaksi Excel-tiedostoa, Test Set export ja Test Case export. Tämän jälkeen voidaan Excelille toteutettua käyttöliittymää hyväksi käyttäen luoda haluttujen testialueiden tilasta graafiset näkymät. Tässä työssä toteutettiin TACOM-projektille luotava testiraportti. Työkalua muokkaamalla saadaan kuitenkin myös muille projekteille luotua vastaavat raportit suhteellisen helposti, mikäli testauksen hallintaan käytetään SpiraTest Management -työkalua.

Raportin generointi aloitetaan käyttöliittymän vasemmasta yläkulmasta löytyvässä Start -> Update Project Test report -painikkeella. Tämän jälkeen esiin nousee uusi ikkuna, josta voidaan valita testiraporttiin halutut asiat. (Kuva 3.)



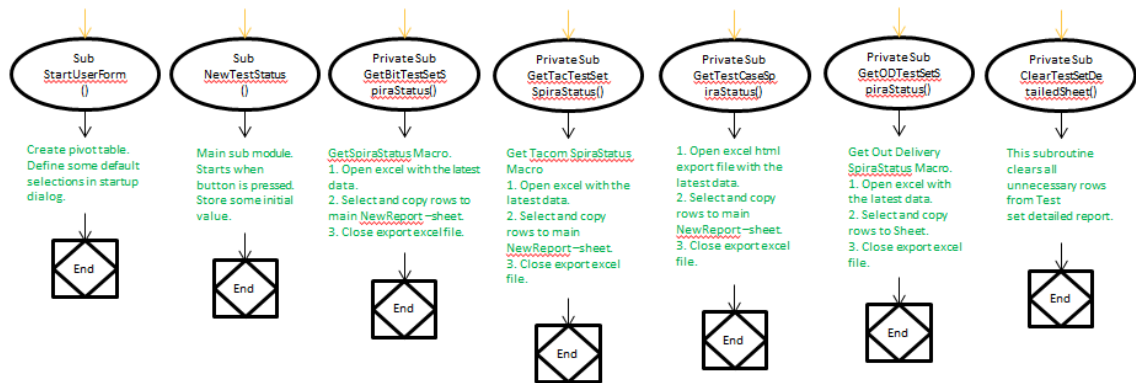
KUVA 3. Testiraportointi työkalun käyttöliittymän valintaikkuna

Työkalun valintaikkunan vuokaavio on esitetty kuvassa 4.

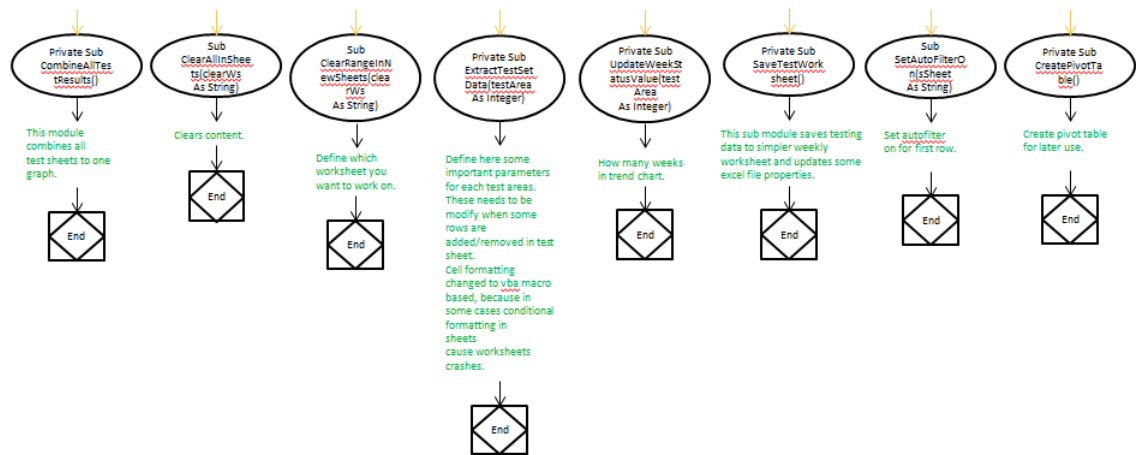


KUVA 4. Testiraportointi työkalun valintaikkunan vuokaavio

Testiraportointi-työkalu tuli sisältämään 15 moduulia. Moduulien sisältö pääpiirteittäin. (Kuva 5, Kuva 6).

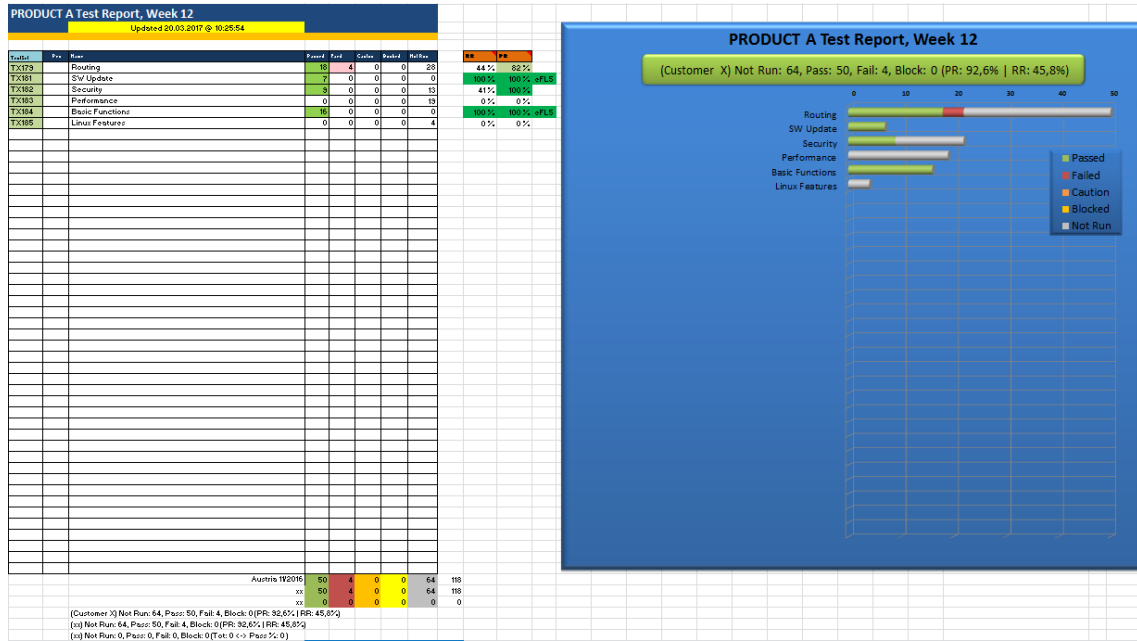


KUVA 5. Testiraportointi työkalun vuokaavio, osa 1



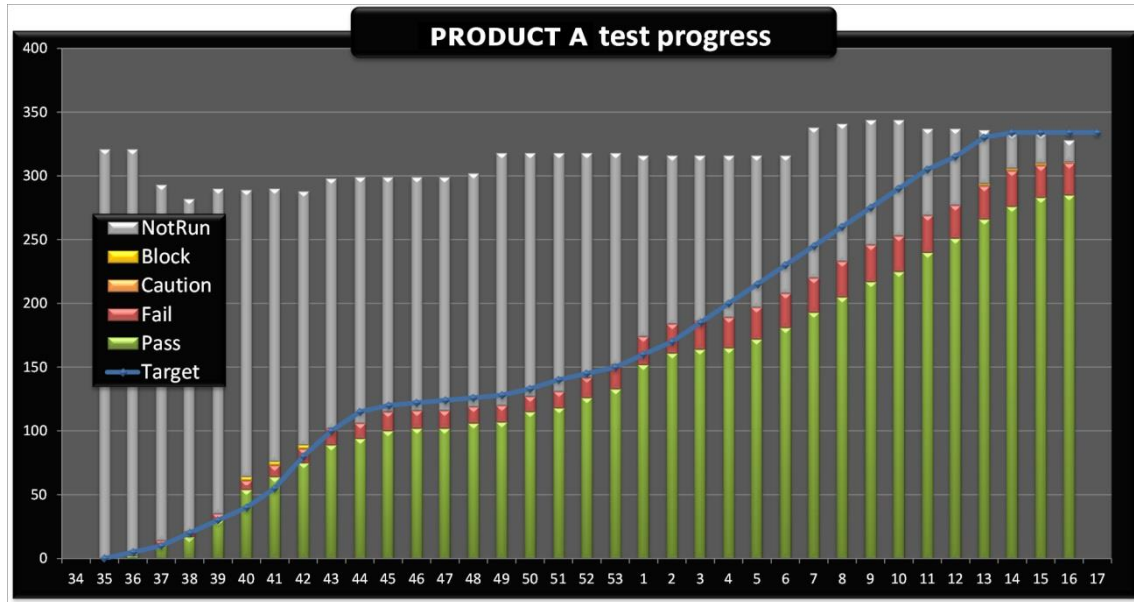
KUVA 6. Testiraportointi työkalun vuokaavio, osa 2

Testauksen tilanteesta saadaan viikoittainen raportti tuotteittain. Raportin sisältö on riippuvainen testauksenhallinta työkalussa (SpiraTest) määritellyistä testiseteistä. Mikäli testaus on jaettu eri testisetteihin, saadaan myös testiraporttiin tarkempi tila eri alueiden testauksen tilanteesta. Esimerkissä testit on jaettu kuuden eri testisetin alle. (Kuva 7.)



KUVA 7. Esimerkki testiraportointi työkalusta saatavasta testauksen tilanteesta tuotteittain

Työkalu generoi lisäksi pylväsdiagrammin, josta nähdään koko projektin ajalta viikoittainen testi etenemä historiatietoineen. Testi etenemälle voidaan asettaa myös viikoittainen tavoitekäyrä, josta voidaan seurata ollaanko projektissa testauksen osalta aikataulussa vai jäljessä siitä. (Kuva 8.)



KUVA 8. Esimerkki testiraportointi työkalusta saatavasta viikoittaisesta testi etenemästä

Edellä mainittujen lisäksi työkalulla voidaan summata kaikkien projektissa olevien tuotteiden tilanne yhdelle näkymälle. (Kuva 9.)

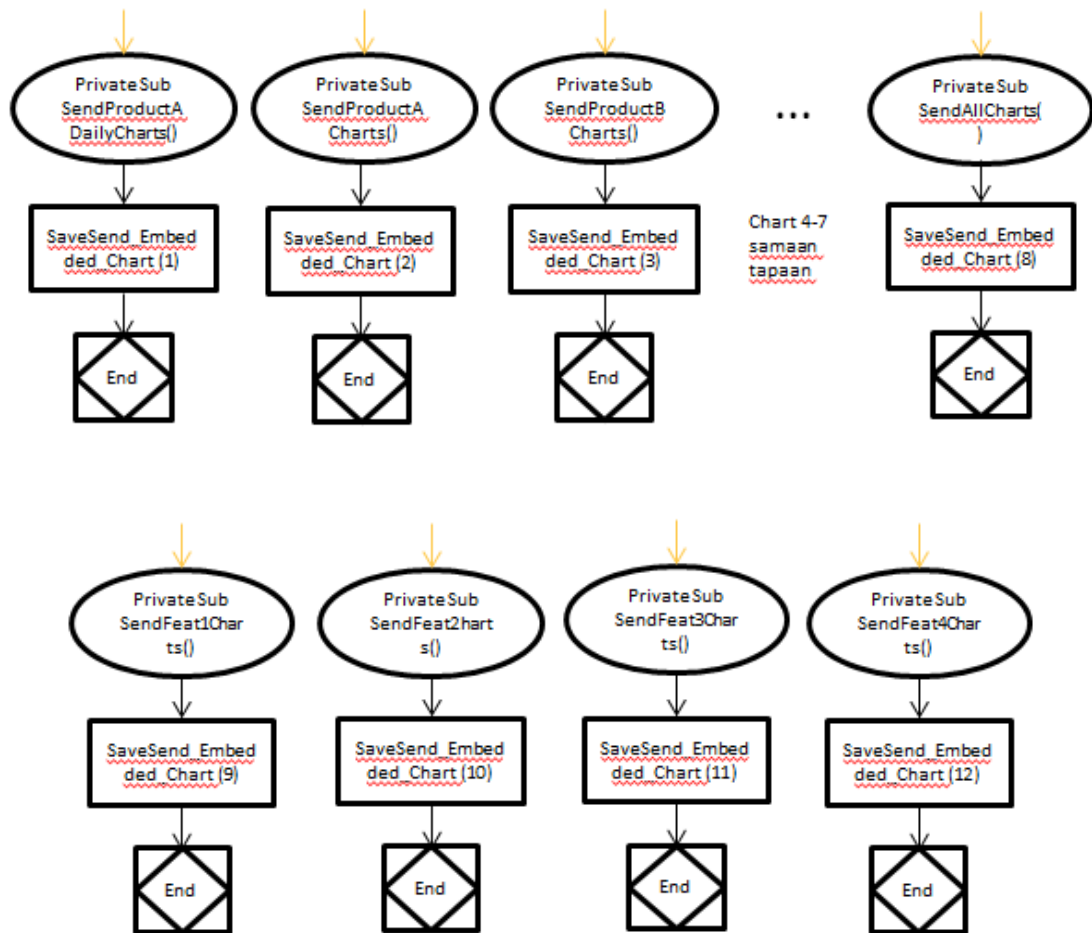


KUVA 9. Esimerkki testiraportointi työkalusta saatavasta kokonaistilanteesta kaikille tuotteille

6.2.3 Sähköpostiominaisuus

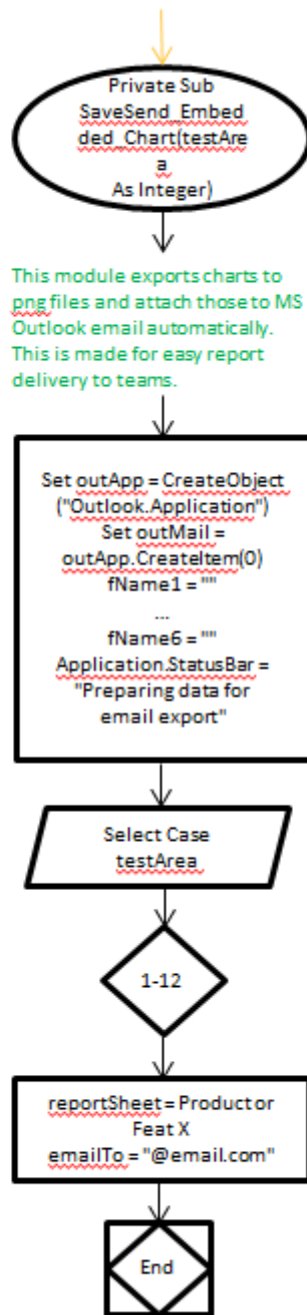
Testiraportointityökalun lisäominaisuutena on lähettää diagrammit testauksen tilanteesta yhdellä näppäimen painalluksella halutuille henkilöille. Ennalta määrätty henkilöt voidaan lisätä työkaluun valmiiksi, jolloin haluttu raportti liitetään Outlook sähköpostin liitteeksi automaattisesti. Viestin jakelulista sisältää tällöin automaattisesti myös henkilöt, joille raportti lähetetään. Tämä toiminto helpottaa raportointia tietyille kohderyhmälle huomattavasti.

Lähetettävän diagrammin valinta tapahtuu painamalla haluttua painiketta käyttöliittymässä. Visual Basicilla toteutettu valintalogiikka noudattaa käyttöliittymällä olevaa näkymää. (Kuva 10.)



KUVA 10. Lähetettävän diagrammin valinta

Diagrammin valinta ja sähköpostiin lähetytys tapahtuu seuraavasti. (Kuva 11.)



This module exports charts to png files and attach those to MS Outlook email automatically. This is made for easy report delivery to teams.

KUVA 11. Diagrammin lähetytys sähköpostiin

6.3 Yhteenveto toteutuksesta

Työssä toteutetulla työkalulla voidaan paikata SpiraTest Management -työkalusta puuttuvia raportointi ominaisuuksia. Työkalulla saadaan ajantasainen tieto tuotteen, tuotteen tietyn ominaisuuden tai vaikkapa tietyn testitiimin testauksen tilanteesta. Työkalu ei ole automaattinen ja vaatii tiettyjen projektikohtaisten asetusten tekemisen kullekin projektille. Asetusten tekeminen vaatii myös perusymmärrystä Visual Basic -ohjelmointikielestä ja ymmärrystä ohjelman rakenteesta. Asetusten jälkeen testauksen raportointi visuaalisessa ja helposti ymmärrettävässä muodossa on kuitenkin helppoa. Työkalulla voidaan generoida yksi Excel-tiedosto sisältäen koko projektin testauksen tilanne tai vaihtoehtoisesti vain jonkin tietyn tuotteen tai alueen status. Tämä tieto voidaan myös jakaa helposti sähköpostin välityksellä.

7 YHTEENVETO

Tämän työn tarkoitus oli määritellä Bittium TACOM-tuoteperheelle testauksessa käytettävä rakenne, jota voidaan käyttää SpiraTest Management -työkalussa testauksen hallintaan. Tämä rakenne tarkoittaa miten testauksen hallintaan käytettävässä työkalussa jaotellaan testitapaukset ja testisetit. Tavoitteena oli löytää yhtenäinen rakenne, jota voidaan käyttää kaikille TACOM-tuoteperheen tuotteille. Lisäksi työssä toteutettiin raportointiin käytettävä työkalu helpottamaan testauksen tilanteen esittämistä helposti ymmärrettävässä muodossa. Raportointiin tehty työkalu toteutettiin Visual Basic -ohjelmointikielellä MS Excelille.

Työ lähdettiin tekemään lähtökohdasta, että mitään yllä mainituista asioista ei projektista löytynyt valmiina, lukuun ottamatta muutamia testitapauksia. Näin asiaan oli melko vapaat kädet ja mm. työkaluun luoduille testitapauksille ja testiseteille sai rakentaa omiin kokemuksiin perustuen parhaaksi katsomansa rakenteen.

Työn lopussa testitapauksia oli testauksen hallinnan työkaluun kirjoitettu jo useampi sata. Työ eteni aikataulussa eikä isompia vastoinkäymisiä projektin aikana esiintynyt. Normaaleja projektityöhön liittyviä asioita toki tuli vastaan, kuten resurssien vähyys ja muiden työtehtävien kuormittavuus vieden aikaa tältä tehtävältä.

Työlle asetettu tavoite saavutettiin työn aikana ja testaukselle saatiin jo varsin kattavat testit rakennettua testauksenhallinta työkaluun. Tällä saatiin testauksen systemaattisuutta mikä aiemmin puuttui projektista. Myös päivittäisen testaustyön raportoinnin saattaminen kuntoon toteutui projektin edetessä. Tätä edesauttoi projektin ohessa tapahtunut testiautomaation kehitystyö. Testiautomaatiosta saatava päivittäinen raportointi saatiin myös projektin aikana integroitua käytettyyn testauksenhallintatyökaluun.

Testaustyö on jatkuvaa muutosta ja tämän työn jälkeen myös testauksen kehitystyö tulee luonnollisesti jatkumaan. Tätä kautta myös testauksen raportointia tullaan jatkossa kehittämään automaattisemmaksi.

LÄHTEET

1. Bittium Tactical Wireless IP Network. 2017. Bittium Corporation. Saatavissa:
https://www.bittium.com/products_services/defense/bittium_tactical_wireless_ip_network#bittium_tactical_wireless_ip_network_trade_tac_win.
Hakupäivä 28.10.2016.
2. Ohjelmistoradio. 2016. Wikipedia, vapaa tietosanakirja. Saatavissa:
<https://fi.wikipedia.org/wiki/Ohjelmistoradio>. Hakupäivä 11.11.2016.
3. Ohjelmiston testaaminen. 2016. Wikipedia, vapaa tietosanakirja. Saatavissa:
https://fi.wikipedia.org/wiki/Ohjelmiston_testaaminen. Hakupäivä 4.11.2016.
4. Pyhäjärvi, Maaret & Pöyhönen, Erkki: Testaus ohjelmistokehityksen osana. Saatavissa:
http://users.jyu.fi/~sakkinen/testaus-2008/aineisto/2_TestausOhjelmistokehityksenOsana_v1.ppt. Hakupäivä 14.3.2017.
5. Tuovinen, Antti-Pekka. Testaustyökalut. Saatavissa:
https://www.cs.helsinki.fi/u/aptuovin/testaus/Ohj_testaus_2013_11.pdf.
Hakupäivä 21.11.2016.
6. Hewlett-Packard, Quality Center Enterprise (QC). Saatavissa:
<http://www8.hp.com/fi/fi/software-solutions/quality-center-quality-management/>. Hakupäivä 4.11.2016.
7. Inflectra, SpiraTest. Saatavissa:
<http://www.inflectra.com/SpiraTest/>. Hakupäivä 4.11.2016.
8. Pyhäjärvi, Maaret & Pöyhönen, Erkki: Tehokas ohjelmistotestaus. Saatavissa:
http://users.jyu.fi/~sakkinen/testaus-2005/kalvot/1_TehokasOhjelmistotestaus_v1.ppt. Hakupäivä 14.3.2017.

9. Mobile ad-hoc network. 2017. Wikipedia, vapaa tietosanakirja. Saatavissa: https://fi.wikipedia.org/wiki/Mobile_ad-hoc_network. Hakupäivä 12.12.2016.

10. van Veenendaal, Erik: Standard glossary of terms used in Software Testing. Version 2.2 (dd. October 19th, 2012). Produced by the 'Glossary Working Party'. International Software Testing Qualifications Board. Saatavissa: http://science.df.lu.lv/kaab13/istqb_glossary_of_testing_terms_2.2.pdf. Hakupäivä 13.3.2017.