



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Jonne Eilimö

# SAPIN TESTAUSPROSESSIN KEHITTÄMINEN

ALMin käyttöönotto testauksen apuvälineeksi ABB:llä

Tekniikka ja liikenne  
2013

## **ALKUSANAT**

Tämä opinnäytetyö on osa Vaasan ammattikorkeakoulun tietotekniikan koulutusohjelmaa. Työn toteutus sijoittuu vuoden 2012 syksylle ja 2013 keväälle. Työn ohjaajana Vaasan ammattikorkeakoulun puolelta oli yliopettaja Pirjo Prosi. Ohjaajana ABB Oy Motors and Generatorsin puolelta toimi Marita Luoma.

Haluan suuresti kiittää molempia ohjaajiani. Työn tekeminen ei olisi onnistunut ilman heidän apua. Erityiskiitokset vielä Nilla Korhoselle henkisestä tuesta työn kirjoituksen aikana.

Vaasassa, 24.5.2013

Jonne Eilimö

VAASAN AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma

## TIIVISTELMÄ

Tekijä	Jonne Eilimö
Opinnäytetyön nimi	SAPin testausprosessin kehittäminen - ALMin käyttöönotto testauksen apuvälineeksi ABB:llä
Vuosi	2013
Kieli	suomi
Sivumäärä	35
Ohjaaja	Pirjo Prosi

---

Tämä työ on tehty ABB Oy Motors and Generators -liiketoimintayksikön laatuosastolle.

Työn tavoite on toteuttaa ratkaisu testauksien seurantaan ja dokumentointiin. Valmiin toteutuksen on määrä korvata edeltävä Excel-toteutus. Työn toteutus tehtiin HP:n Application Lifecycle Management -ohjelmalla. ALMiin rakennettiin ABB Oy Motors and Generators -liiketoimintayksikölle räätälöity testauskokonaisuus. Valmiissa ALM-toteutuksessa testaaja voi raportoida löydetyt ongelmat vähällä vaivalla. Testien etenemisen seuranta helpottuu myös ALM-toteutuksen ansiosta.

ABB:lla käytetään SAP-toiminnanohjausjärjestelmää. Järjestelmään ajetaan vuosittain viisi isompaa ohjelmistopäivitystä jotka sisältävät muutoksia ja korjauksia tuotantoympäristöön. Mahdollisten virheiden varalta nämä muutosten ja korjausten tuomat toimivuudet joudutaan testaamaan ennen tuotannon järjestelmien päivittämistä. Laatuosasto vastaa näiden testauksien tekemisestä. Tarkoitus on, että aikaisempaan Excel-tiedostoon lisättävät tiedot voidaan suoraan lisätä ALMiin.

Valmis toteutus täyttää tilaajan asettamat vaatimukset. Uusi toteutus voi näin ollen kokonaan korvata edellisen Excel-toteutuksen.

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Tietotekniikan koulutusohjelma

## ABSTRACT

Author	Jonne Eilimö
Title	Improvement of SAP Testing Process - Implementing ALM as a Testing Tool at ABB
Year	2013
Language	Finnish
Pages	35
Name of Supervisor	Pirjo Prosi

---

This thesis was done for the ABB Oy Motors and Generators quality unit.

The final solution is a testing tool built inside HP's Application Lifecycle Management application. One of the goals of this project was to replace the previous Excel testing solution. Data that has earlier gone into Excel can now be inserted directly into ALM. A custom ALM implementation was built for ABB Oy Motors and Generators.

ABB uses SAP as their Enterprise Resource Planning tool. The system receives five major software updates every year. To make sure that the updates do not break any functionality, they have to be thoroughly tested before taken into production. The quality unit is tasked to test the updates upon their release. The person conducting the test can now easily report found defects through ALM. This tool also provides an easier way to track the testing process.

The final ALM implementation meets the requirements set by the client. The implementation can replace the earlier Excel solution.

---

Keywords	SAP, ALM, application lifecycle management, software testing
----------	--

**KÄYTETYT LYHENTEET JA MERKIT**

ABB	Asea Brown Boveri, yritys
ALM	Application Lifecycle Management, ohjelmiston elinkaaren hallinta
Alt	Alternate, Alt-näppäin
CR	Change Request, muutospyyntö
Ctrl	Control, Ctrl-näppäin
HP	Hewlett-Packard, yritys
ID	Identifier, tunniste
PDCA	Plan, Do, Check, Act Suunnittele, Toteuta, Tarkasta, Toimi
PDF	Portable Document Format, tiedostomuoto
QC	Quality Center, laatukeskus
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung, yritys
UAT	User Acceptance Testing, hyväksyntätästäus

**KUVIOLUETTELO**

<b>Kuva 1.</b> Demingin PDCA-ympyrä .....	15
<b>Kuva 2.</b> New Test - Uuden testin luominen .....	20
<b>Kuva 3.</b> Design Step Details - Uuden vaiheen syöttäminen testiin.....	21
<b>Kuva 4.</b> New Test Set - Uuden testikokonaisuuden luominen.....	22
<b>Kuva 5.</b> Test Plan Tree - Yksittäisen testin lisääminen testikokonaisuuteen .....	23
<b>Kuva 6.</b> Testin suorittaminen Test Planissa .....	24
<b>Kuva 7.</b> Execution Grid testin aikana.....	25
<b>Kuva 8.</b> Testin ajo Manual Runnerilla .....	26
<b>Kuva 9.</b> Manual Runner: Testin vaiheistus .....	27
<b>Kuva 10.</b> Testin mahdolliset Statukset .....	28
<b>Kuva 11.</b> Uuden Defectin luominen.....	29
<b>Kuva 12.</b> Defect: Assigned To -kenttä .....	30
<b>Kuva 13.</b> Defektit listattuna.....	31

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KÄYTETYT LYHENTEET JA MERKIT

KUVIOLUETTELO

1	JOHDANTO.....	9
	1.1 Tutkimuksen pohjatietoa.....	9
	1.2 Tavoitteet .....	9
	1.3 Haasteet.....	10
	1.4 ABB-yhtymä.....	10
	1.5 ABB Oy Motors and Generators.....	11
2	APPLICATION LIFECYCLE MANAGEMENT .....	12
	2.1 Yleistä .....	12
	2.2 Hewlett-Packard Application Lifecycle Management.....	12
3	SUUNNITTELU .....	14
	3.1 Yleiskuvaus.....	14
	3.2 Demingin PDCA-ympyrä .....	14
	3.2.1 Suunnittele .....	15
	3.2.2 Toteuta.....	16
	3.2.3 Tarkasta .....	16
	3.2.4 Toimi .....	17
4	TOTEUTUS .....	18
	4.1 Yleistä toteutuksesta .....	18
	4.2 Iteroinnin kautta lopulliseen toteutukseen .....	19
	4.3 Lopullinen ALM toteutus .....	19
	4.3.1 Uuden testin luominen .....	19
	4.3.2 Testien vaiheistaminen.....	20
	4.3.3 Testikokonaisuuksien luominen.....	21
	4.4 Testin suorittaminen.....	23
	4.5 Testissä ilmenneiden ongelmien raportointi .....	28
5	YHTEENVETO .....	32

6 JOHTOPÄÄTÖKSET .....	34
LÄHTEET.....	35

# 1 JOHDANTO

## 1.1 Tutkimuksen pohjatietoa

ABB Oy Motors and Generators -liiketoimintayksikössä käytetään SAP-toiminnan ohjausjärjestelmää. Järjestelmään ajetaan vuosittain viisi isompaa ohjelmistopäivitystä jotka sisältävät muutoksia ja korjauksia tuotantoympäristöön. Mahdollisten virheiden varalta nämä muutosten ja korjausten tuomat toimivuudet joudutaan testaamaan ennen tuotannon järjestelmien päivittämistä. ABB:llä näitä kutsutaan UAT-testauksiksi.

Testauksia suorittavat SAPin pääkäyttäjät ja ongelmana on se, että suurin osa testausprosesseista on pääkäyttäjän omassa muistissa, eikä dokumentoituna mihinkään. ABB Oy Motors and Generatorsin SAP-prosessista testataan moottorin ja siihen kuuluvien osien kulku myyntitilauksesta laskutukseen, sisältäen oston, sisäisen logistiikan, laadunhallinnan, talouden ja tuotannon.

Testauksen seuranta on aikaisemmin kirjattu yhteen Excel-tiedostoon. Ongelmana on ollut muun muassa se, että tiedostoa ei ole voitu muokata monen käyttäjän johdosta samanaikaisesti. Tiedosto on usein myös korruptoitunut, joka on pahimmassa tapauksessa johtanut testauksen dokumentoinnin katoamiseen.

## 1.2 Tavoitteet

Työn tarkoituksena on tarjota ABB Oy Motors and Generators -liiketoimintayksikölle toimiva ratkaisu testauksien seurantaan ja dokumentointiin. ABB Suomi tasolla on tällä hetkellä käytössä HP:n ALM-työkalu, jota tullaan käyttämään tähän tarkoitukseen. Tavoitteena on, että vanhempi Excel-toteutus voidaan kokonaan korvata HP:n ALM-työkalulla. Kaikki toiminnollisuus edeltävästä Excel-toteutuksesta tulisi saada toteutettua ALMin puolella.

### 1.3 Haasteet

Opinnäytetyön kannalta haasteena on löytää ratkaisu, joka sopii kaikille testaajille. Testaukset voivat vaihdella hyvinkin paljon eri osastojen ja testaajien välillä. Aikataulu myös asettaa oman haasteensa työn onnistumiselle. Koska ALMin muutokset voidaan testata käyttäjien kesken vasta isojen SAP-järjestelmäpäivityksien yhteydessä, on testaajilta saatava palaute harvassa. Suoraan SAPin testaajilta kerättävää palautetta ALMin toteutuksesta tulee vain kahdesta kevään 2013 aikana tehtävästä testistä. Tämä asettaa omat haasteensa toimivan ALMin toteutuksen tekemiselle.

### 1.4 ABB-yhtymä

ABB muodostettiin tammikuussa 1988, kun ruotsalainen Asea ja sveitsiläisen Brown Boverin sähkötekniiset liiketoiminnot yhdistettiin 50:50-omistusperiaatteella. Yhtiön nimi, ABB, tulee suoraan näiden kahden yhtiön nimestä. ABB:n kasvu perustuu pitkälti sen teknologiseen voimaan ja vahvoihin paikallisiin juuriin, joita Suomessa edustaa Strömberg.

ABB:n ydinliiketoiminta on organisoitu viiteen divisioonaan:

- Discrete Automation and Motion
- Process Automation
- Low Voltage Products
- Power Systems
- Power Products.

Moottoreiden ja generaattoreiden tuotanto sai alkunsa vuonna 1889 ja Vaasassa se alkoi vuonna 1944.

Tänä päivänä ABB on johtava sähkövoima- ja automaatioteknologiayhtymä, jonka tuotteet, järjestelmät ja palvelut parantavat teollisuus- ja energiayhtiöasiakkaiden kilpailukykyä ympäristömyönteisesti. ABB:n palveluksessa työskentelee maailmanlaajuisesti yli 145 000 henkilöä noin 100 maassa. Suomessa tästä työskentelee noin 7 000 henkilöä yli 30 paikkakunnalla. ABB Oy:n tehdaskeskittymät sijaitsevat Helsingissä, Vaasassa ja Porvoossa. /1/ /2/

## **1.5 ABB Oy Motors and Generators**

Suomen ABB Motors and Generators -liiketoimintayksikkö tarjoaa kattavan valikoiman luotettavia, korkean hyötysuhteen moottoreita ja generaattoreita kaikkiin sovelluksiin. Yksikkö panostaa vahvasti korkean hyötysuhteen moottoreiden ja generaattoreiden tutkimukseen ja tuotekehitykseen. Vaasan tehtaalla on lisäksi maailmanlaajuinen vastuu ABB:n valmistamista räjähdysvaarallisten tilojen pienjännitemoottoreista. /3/

## 2 APPLICATION LIFECYCLE MANAGEMENT

### 2.1 Yleistä

Termi Application Lifecycle Management on yksi ohjelmistokehityksen kulmakivistä. Tällä tarkoitetaan ohjelmiston elinkaaren hallintaa, joka on jatkuva liiketoiminnan prosessi, joka yhdistää vaatimusten-, konfiguraation hallinnan ja laadunvarmistuksen prosessit toisiinsa muutostenhallinnan kautta.

Keskeisimmät prosessit ALMin hallinnan alla:

- vaatimukset
- ohjelmistokonfiguraatio
- testaus
- virheet
- toimitukset
- arkkitehtuuri. /4/ /5/

Nämä prosessit parantavat ohjelmistokehityksen tuottavuutta luomalla paremman jäljitettävyyden, läpinäkyvyyden ja laadunvarmistuksen. ALM-työkalulla on tarkoitus kerätä ja kategorisoida ohjelmiston tuotannon aikainen tieto. Tieto voi olla esimerkiksi testaajilta tulevat virheraportit. Nämä raportit sidotaan seurantakoodilla virheeseen. Seurantakoodin avulla ohjelmoijat näkevät helposti missä virhe on tapahtunut. Tällä tiedolla virhe korjataan ja merkitään ALMiin korjatuksi. Korjauksen jälkeen virhe testataan uudestaan ja arvioidaan voidaanko ominaisuus viedä tuotantoon. /4/

Erilaisia ALM-työkaluja on olemassa monia. Koska ABB:lla on jo käytössään suuri määrä HP:n ohjelmistoja, tullaan HP:n Application Lifecycle Management -ohjelmistoa hyödyntämään tämän työn toteutuksessa.

### 2.2 Hewlett-Packard Application Lifecycle Management

ABB:lla HP:n ALM-työkalua tullaan käyttämään SAPin kehityksen apuvälineenä. ABB:lla SAPin muutokset ja kustomoidut SAP-transaktiot ohjelmoidaan pääosin

Intiassa. Suomessa kerätään tieto siitä, mitä halutaan tehdä. Tilauksen yhteydessä tieto syötetään HP:n Quality Center -ohjelmaan, jossa luodaan niin kutsuttu CR-tiketti. CR-tiketti sisältää tiedon siitä, mitä halutaan tehdä, ja kuka tilauksen vastuuhenkilö on. Intiassa CR ilmestyy ohjelmoijien työjonoon, jolloin sen toteutus alkaa. Kun työ on valmis, kuitataan se Intiassa tehdyksi. Suomessa vastuuhenkilön saa kuitauksesta viestin, jolloin työ on tarkastettava. Jos toteutuksessa on vielä puutteita, palautetaan se dokumentoinnin kanssa takaisin ohjelmoijalle. Tätä jatketaan kunnes työ on valmis.

Edellä mainitun testauksen tekee kuitenkin ainoastaan työn tilaaja, joka useimmiten on yksi henkilö. Muutoksia ei vielä tällä kertaa testata kokonaisvaltaisesti. Muutokset ajetaan pääsääntöisesti tuotantoon SAPin järjestelmäpäivityksien yhteydessä, ja tämän takia ne voidaan helpoiten testata päivitysten yhteydessä. Tähän testausvaiheeseen ALM on todella hyvä työkalu. Jos testeissä havaitaan jonkin muutostyön aiheuttaneen ongelmia SAP-transaktioiden kanssa, huomataan se viimeistään näiden testien aikana. Tällöin voidaan vanha CR avata uudelleen. Helpommaksi työn tekee se, että vanha CR voidaan suoraan avata ALMista. Jos virhe tai ongelma on uusi, voidaan siitä luoda uusi CR ALMin tai HP Quality Centerin kautta.

## 3 SUUNNITTELU

### 3.1 Yleiskuvaus

ABB:lla ajetaan vuodessa viisi suurempaa SAP-järjestelmäpäivitystä läpi. Ennen näiden viemistä tuotantoon, on järjestelmän toiminta testattava kokonaisvaltaisesti. Ongelmaksi voi muodostua täydellisten testien vähäinen määrä.

Tämän projektin aikataulun ensimmäiset testit antavat tärkeätä tietoa kuinka tämän hetkinen Excel-toteutus on toiminut. Tarkoituksena olisi kerätä testaaajilta palautetta miten tämän hetkinen toteutus on toiminut ja mitkä ominaisuudet vaativat parannuksen. Työn toteutuksen kannalta tärkeintä olisi tietää mitä testaaajat haluavat uuteen toteutukseen.

Työn aikataulun puitteissa ehditään suorittaa kolmet testit. Ensimmäisellä kerralla kerätään tarvittavaa tietoa, kun toisella kerralla voi jo tuoda ALM-toteutuksen vanhan Excelin rinnalle. Tavoitteena on, että kolmannella kerralla ALM-toteutus on käytössä testeissä, ja että Excel-toteutus on vain varalla jos ALMin kanssa ilmenee jokin suurempi ongelma.

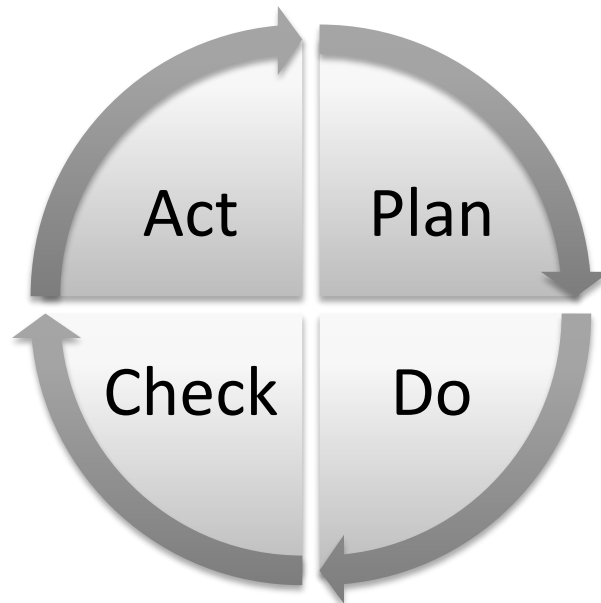
Koska suunnittelu on tärkeässä roolissa työn toteutuksen laatuun, on tähän panostettava tarpeeksi. Suunnittelun onnistumisen kannalta olisi hyvä käyttää jotakin jo aikaisemmin hyväksi todettua työkalua.

### 3.2 Demingin PDCA-ympyrä

Eräs laadunhallinnan ja ongelmanratkaisun peruseriaate on William Edwards Demingin vuonna 1951 tunnetuksi tekemä PDCA-ympyrä. Nimensä ympyrä on saanut englannin kielen sanoista Plan — Do — Check — Act, jotka vapaasti suomennettuna on suunnittele, toteuta, arvioi ja paranna. /6/

SAPin järjestelmäpäivitysten ja projektisuunnitelman syklisyyden ansiosta Demingin PDCA-ympyrä soveltuu hyvin työn toteutuksen apuvälineeksi. Suunnitteluvaiheen jälkeen toteutetaan suunniteltu muutos. Toteutuksesta kerättävän palautteen

perusteella voidaan toteutus arvioida. Palautteen pohjalta voidaan toteutusta yrittää parantaa, jonka jälkeen taas siirrytään takaisin suunnitteluun.



**Kuva 1.** Demingin PDCA-ympyrä /6/

### 3.2.1 Suunnittele

Demingin PDCA-ympyrä jäsentelee prosessin parantamisen neljään eri vaiheeseen, joista ensimmäinen on suunnittelu. Tämä vaihe on prosessin parantamisen kannalta todella tärkeä.

Suunnitteluprosessi aloitetaan ongelmien määrittelyllä. Nämä ongelmat ovat luonnollisesti niitä, joihin halutaan prosessin kehittämisen avulla ratkaisu. Tässä vaiheessa on kerättävä tarpeeksi tietoa siitä, mikä oli edeltävä tilanne, ja miten tätä voisi parantaa. Lisäksi olisi hyvä määritellä jokin mittari, jolla onnistumisen voi arvioida.

Suunnitteluvaihe käsittää SAPin järjestelmäpäivitysten aikaisen tiedon keräämisen. Tämän vaiheen aikana tarkastetaan kuinka edeltävää Excel-toteutusta on hyödynnetty, ja mitä tietoa testien aikana halutaan raportoida. Tarkoitus on, että mitään tietoa tai toiminnallisuutta ei menetetä johtuen testien seurannan ja raportoinnin

siirtämisestä ALMiin. Projektin yksi tavoite on, että ALMin tulisi tarjota vähintään samat ominaisuudet kuin edeltävä Excel-toteutus.

Suunnitteluosuuden aikana tulisi määritellä kaikkien eri testiskenaarioiden vaatimat toimenpiteet. Toimenpiteet tulee kerätä yhdeksi pitkäksi testiputkeksi, joka on helposti vietävissä ALMiin. Putket sisältävät tiedot, joiden pohjalta testaaja suorittaa testin. Tällaisia tietoja ovat esimerkiksi tilausnumero, moottorin kokoonpano, vastuuhenkilö, sekä jokin merkintä siitä, mitä juuri tällä testillä halutaan testata.

Näiden tietojen pohjalta voidaan siirtyä Demingin PDCA-ympyrän seuraavaan vaiheeseen, eli toteutukseen. /6/

### **3.2.2 Toteuta**

Demingin PDCA-ympyrän toisessa vaiheessa on määrä suorittaa suunnitellut muutokset. Projektin kannalta tämä tarkoittaa sitä, että suunnitellut prosessiputket ja niiden kuvaukset rakennetaan ALMiin.

Mahdollisimman hyvän tuloksen saamiseksi on jatkuvasti arvioitava tekemistään. Tämän takia on mahdollista, että suunnitelmaan on tässä vaiheessa jo tullut muutoksia. Voi olla, että jokin suunniteltu ratkaisu ei yksinkertaisesti toimi ALMin toteutuksessa. Tällaisessa tilanteessa on palattava takaisin suunnitteluvaiheeseen, ja mietittävä miten saadaan uusi ratkaisu aikaiseksi, joka myös toimii ALMissä. /6/

### **3.2.3 Tarkasta**

Kolmas vaihe Demingin PDCA-ympyrässä on edellisen vaiheen, eli toteutuksen, tarkastus ja analysointi. Vaihe on erittäin tärkeä, sillä tämän tiedon pohjalta tehdään parannusehdotuksen suunnitelma.

Mahdollisuuksien mukaan, on tässä vaiheessa hyvä käyttää jotakin mittaria millä suorituksen laatua arvioidaan. Tässä työssä tosin ei ole mitään numeerista mittaria, vaan toteutuksen testaajilta kerätään palautetta haastattelun muodossa. Jokaisen testaajan ja vastuuhenkilön kanssa sovitaan tapaaminen, missä pohditaan onnistumisia

ja epäonnistumisia. Kun palaute on kerätty ja analysoitu, voidaan siirtyä PDCA-ympyrän viimeiseen vaiheeseen. /6/

### **3.2.4 Toimi**

Demingin PDCA-ympyrän neljäs, ja viimeinen, vaihe on palautteen pohjalta toimiminen. Kun ratkaisuun lopulta ollaan tyytyväisiä, otetaan muutos käyttöön kaikkien testaajien kesken.

Jos tarjottua ratkaisua ei kuitenkaan käyttöönoteta, pyörähtää Demingin PDCA-ympyrä alkuunsa, ja työtä jatketaan jälleen suunnittelun merkeissä. Ympyrän neljää askelta jatketaan kunnes ratkaisu täyttää projektille asetetut tavoitteet, tai jos päätetään projektin rajaamisesta ajan tai budjetin puitteissa. Jos suunnitteluun joudutaan palaamaan, on erityisesti pohdittava mikä tämän aiheutti.

SAPin päivitysten yhteydessä tehtävien testien syklisyys antaa hyvän mahdollisuuden parantaa toteutusta muutamaa otteeseen. Kaikkien testien jälkeen tulisi kerätä mahdollisimman paljon palautetta testaajilta, ja toimia näiden pohjalta. /6/

## 4 TOTEUTUS

Toteutusosiossa kuvaillaan ALMiin tehtävän tietokannan rakenne. Kuten aikaisemmin on todettu, seuraa työn toteutus Demingin PDCA-ympyrän rakennetta. Edellisessä luvussa perehdyttiin työn suunnitteluun ja sen teoriaan, kun tässä perehdytään tarkemmin toteutukseen. Lopullinen toteutus tehdään HP:n Application Lifecycle Management ohjelman versioon 11.00.

### 4.1 Yleistä toteutuksesta

Kuten aikaisemmin on todettu, on PDCA-ympyrän ensimmäinen vaihe suunnittelu. Tämä työ aloitettiin keräämällä mahdollisimman paljon tietoa yhden täydellisen SAP-järjestelmätestin aikana. Tiedot kerättiin Excel-tiedostoon, jossa näiden ylläpito ja järjestäminen helpottui Excelin työkalujen ansiosta. Käyttäjiltä kerätyn tiedon perusteella luotiin jokaiselle testiskenaariolle täydellinen prosessiputki, eli eräänlainen kaava testauksen läpiviemiseen.

Täydellisen järjestelmätestin aikana käydään läpi lähes 70 erillistä testikokonaisuutta. Suurimmat testikokonaisuudet kattavat lähes 100 vaihetta, kun pienimmät testikokonaisuudet ovat vain muutaman vaiheen mittaiset.

Laajimmat testit käsittävät toimitusketjun aina asiakkaan tilauksen tekemisestä moottorin valmistukseen ja laskutukseen. Tähän väliin mahtuu muun muassa komponenttien valmistus, moottorin kokoonpano, sisäiset testit sekä lähettäminen.

Lähes kaikissa edellä mainituissa vaiheissa käytetään SAPia tuotannon ohjaukseen. Jos SAPista ei löydy valmista transaktiota, tai toimintoa, joka halutun työn tekisi, on se ABB:lla useimmiten toteutettu kustomoituna transaktion. Nämä kustomoidut transaktiot ovat testien kannalta hankalimmat, sillä niiden toiminta riippuu usein monista eri tekijöistä ja tiedoista. Voi olla, että jokin SAPin järjestelmäpäivityksen tuoma muutos rikkoo yhden kustomoidun transaktion, joka puolestaan rikkoo kaksi muuta kustomoitua transaktiota.

## 4.2 Iteroinnin kautta lopulliseen toteutukseen

Suunnittelun jälkeen siirryttiin tekemään itse toteutusta. Suunnitteluun oltiin käytännössä tyytyväisiä, sillä se sisälsi ratkaisut niihin haasteisiin ja ehtoihin, jotka työn alussa asetettiin. Testien aikana kuitenkin huomattiin, että toteutus oli liian monimutkainen.

ALMin puolella jokainen vaihe piti käydä erikseen kuittaamassa tehdyksi. Näin ollen oli mahdollista, että yhdellä testaajalla oli edessään kaksikymmentä testiä, jotka jokainen vaati manuaalisen kuittauksen. Tämä sama toistui jokaisessa testiskenariossa, missä käytettiin samaa pohjaa. Yhdellä testaajalla saattoi olla kokonaisen SAP-järjestelmätestin aikana yhteensä lähes tuhat käsin kuitattavaa testiä.

Kuten alkuperäisessä suunnitelmassa todettiin, oli ensimmäinen testi pilotti. Pilotin jälkeen haluttiin kaikilta testaajilta palautetta toteutuksesta. ALMiin oltiin muuten tyytyväisiä, mutta kuittausten määrä oli aivan liian suuri.

Suunnitteluun oli siis palattava. Suunnittelun tarkoitus tällä kertaa olisi saada samat tiedot jäsenettyä siten, että kuittausten määrä pienenesi moninkertaisesti. Haasteeksi muodostui se, että yksi työn tavoitteista oli tiedon säilyminen. Mitään toiminnollisuutta ei siis voida hylätä, vaan ratkaisun on oltava helpompi, kuitenkin sisältäen saman määrän tietoa.

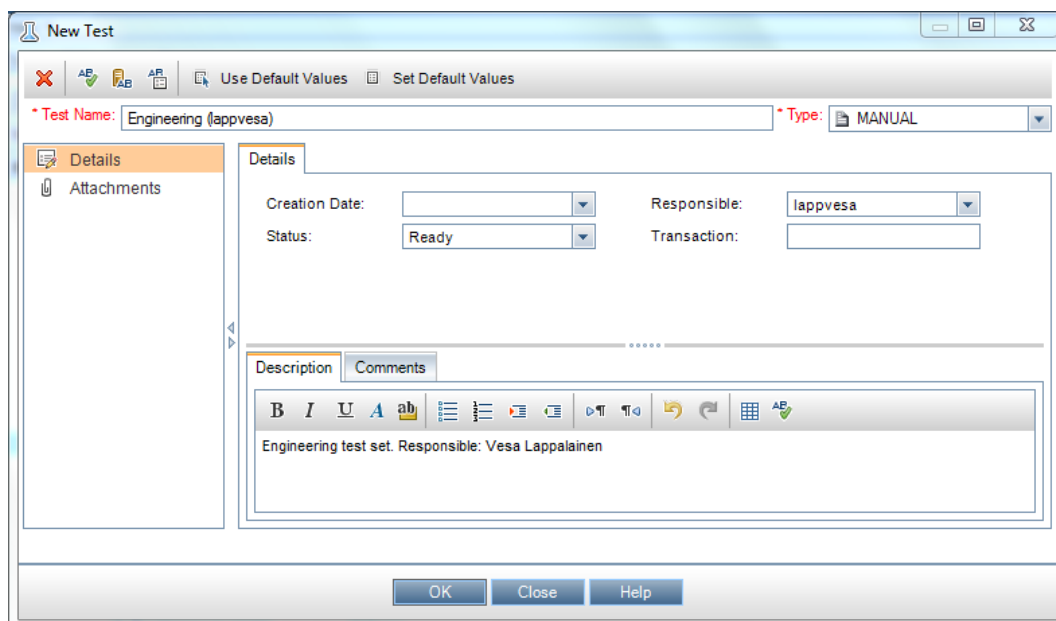
## 4.3 Lopullinen ALM toteutus

Lopullinen ALM-toteutus sisältää käytännössä saman tiedon kuin edellinen toteutus, mutta tällä kertaa yksittäisten käyttäjien tiedot on ryhmitetty valmiiksi pakeiksi. Tämä tekee testauksen helpommaksi niille, joilla on monta peräkkäistä testiä tarkastettavanaan.

### 4.3.1 Uuden testin luominen

Testikokonaisuuden luominen ALMiin aloitetaan ohjelman Test Plan puolella (vasemmasta laidasta Testing » Test Plan). Uuden testin luomiseksi valitaan valikosta

Test » New Test, tai vaihtoehtoisesti näppäinyhdistelmällä Ctrl + N, jolloin uusi ikkuna aukeaa (**Kuva 2.**).



**Kuva 2.** New Test - Uuden testin luominen

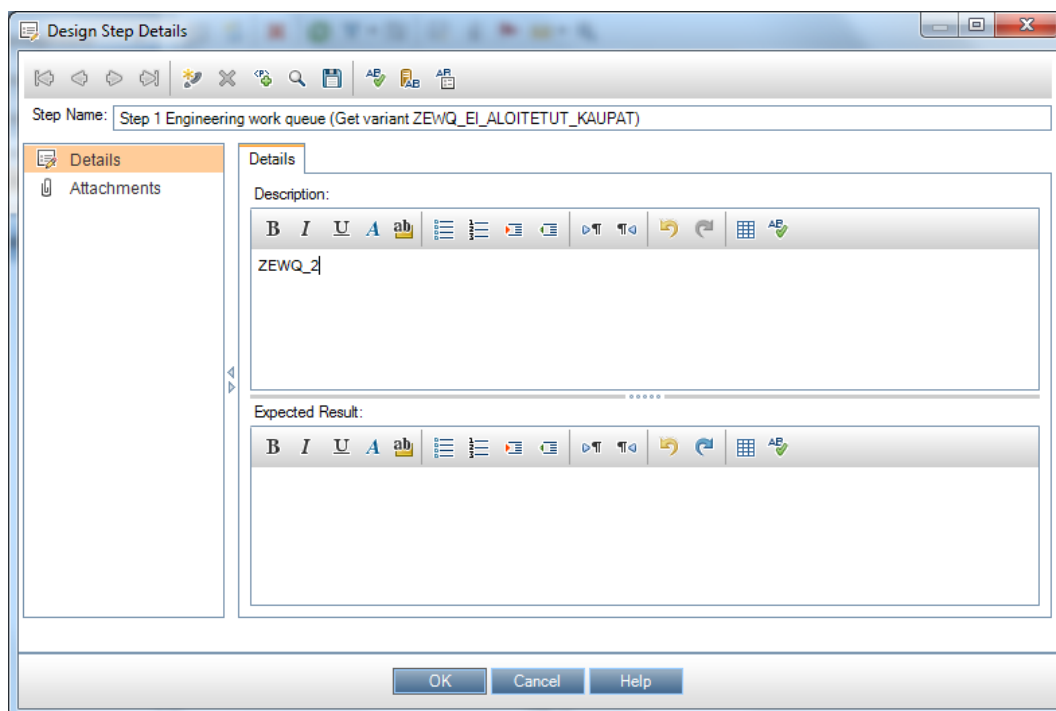
New Test -ikkunaan syötetään yllä näkyvät tiedot. Test Name -ruutuun syötetään kuvaava nimi testille, kuvan 2 esimerkissä tämä on Engineering ja suluissa vastuuhenkilön käyttäjätunnus. Suositeltavaa on, että vastuuhenkilö myös syötetään Responsible-valintalistaan. Tämä ei kuitenkaan onnistu jos vastuuhenkilöllä ei ole ALM-tunnuksia. Vastuuhenkilön syöttäminen helpottaa esimerkiksi testien aikana omien testien seuranta ja valintaa. Kun tiedot on täytetty, siirrytään seuraavaan kohtaan painamalla OK-näppäintä.

### 4.3.2 Testien vaiheistaminen

Kuvassa 3 syötetään uutta vaihetta testiin. Tähän ikkunaan päästään painamalla valikosta New Step, tai näppäinyhdistelmällä Alt + N. Vaiheilla voidaan helposti ryhmittää yhden henkilön peräkkäin suorittamat testit yhteen. Step Name -kenttään syötetään testivaiheen kuvaus, joka yllä olevassa kuvan tapauksessa on Engineering work Queue (Get variant ZEWQ\_EI\_ALOITETUT\_KAUPAT). Description-kenttään syötetään testattavan SAP-transaktion nimi.

Kun tiedot on syötetty, voidaan siirtyä seuraavaan saman henkilön suorittamaan vaiheeseen. Tämä tapahtuu painamalla työkalurivistä Next Entry -näppäintä, tai vaihtoehtoisesti näppäinyhdistelmällä Alt + oikea nuoli.

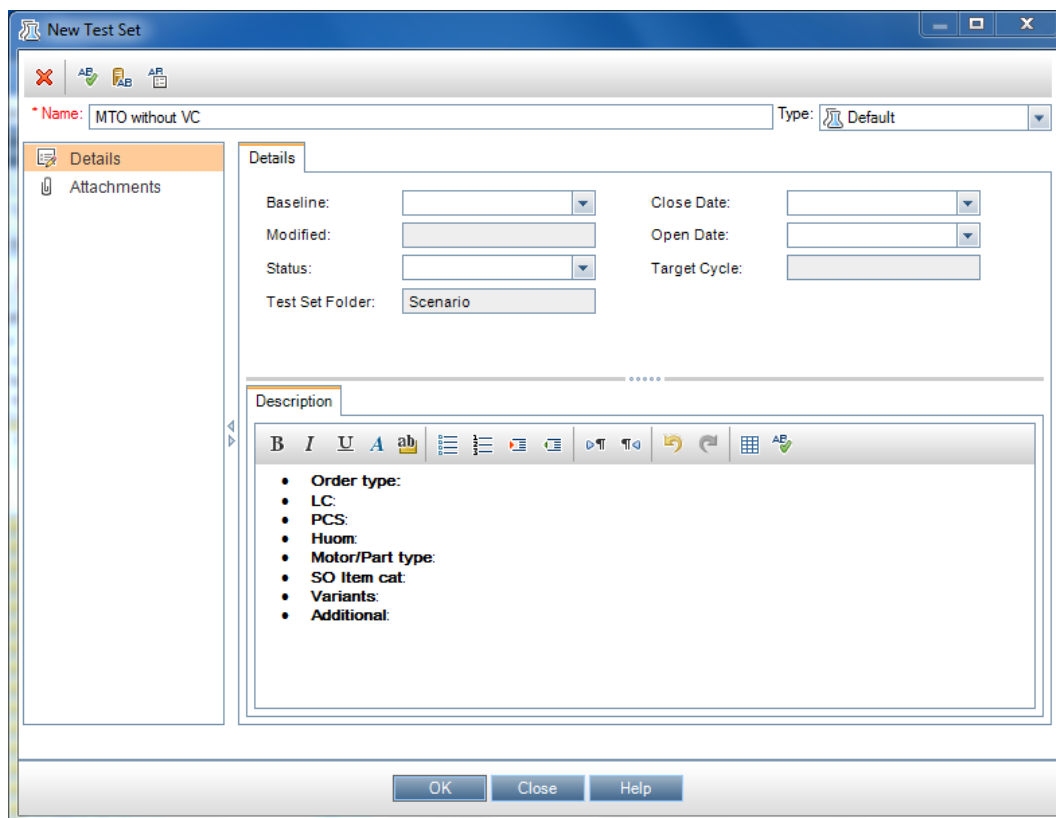
Kun kaikki saman henkilön peräkkäin suorittamat testivaiheet on syötetty, voidaan testin luominen lopettaa painamalla OK-näppäintä.



**Kuva 3.** Design Step Details - Uuden vaiheen syöttäminen testiin

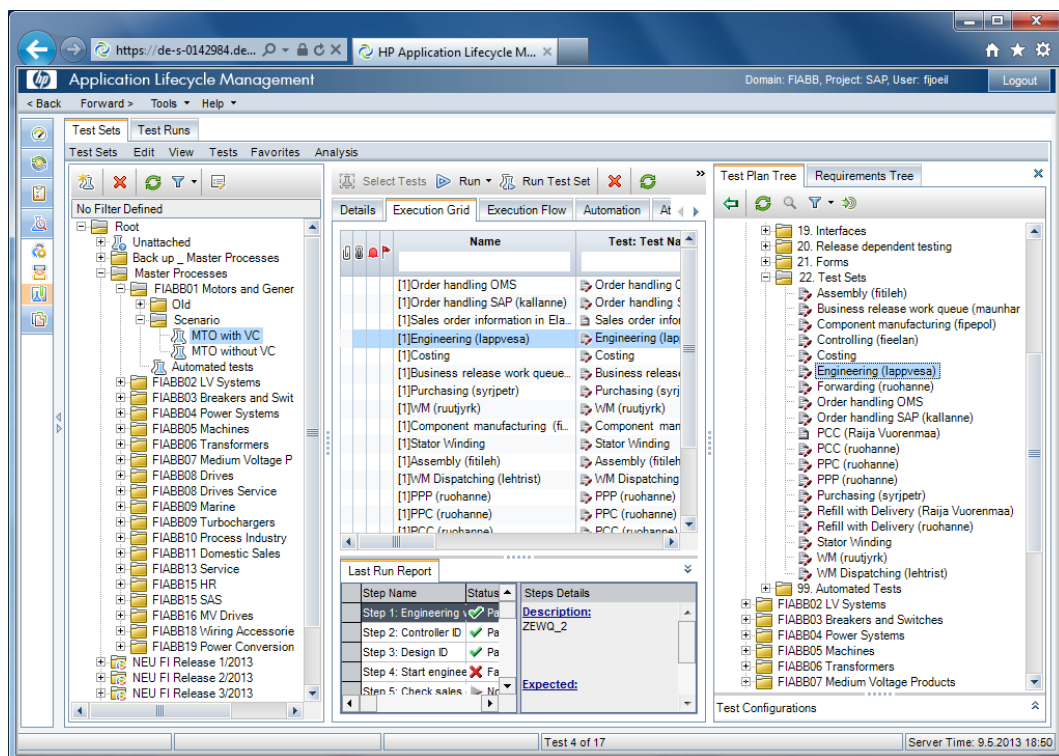
### 4.3.3 Testikokonaisuuksien luominen

Kun kaikki testivaiheet on luotu, on aika luoda testikokonaisuus (Test Set). Tämä onnistuu helpoiten painamalla työkalurivistä New Test Set -näppäintä, tai vaihtoehtoisesti näppäinyhdistelmää Alt + N.



**Kuva 4.** New Test Set - Uuden testikokonaisuuden luominen

Kuvassa 4 näkyy uuden testikokonaisuuden luomisen yhteydessä täytettävät tiedot. Name-kenttään syötetään testiä kuvaava nimi, mikä tässä tapauksessa on ”MTO without VC”, joka kertoo testin olevan ”made to order without variable configurator”. Näitä skenaarioita on yhteensä noin kaksikymmentä. Description-kenttään syötetään kuvassa 4 näkyvät tiedot, jotka kertovat testistä hieman enemmän. Description-kentän tiedot vastaavat edellisen Excel-toteutuksen sarakkeita.

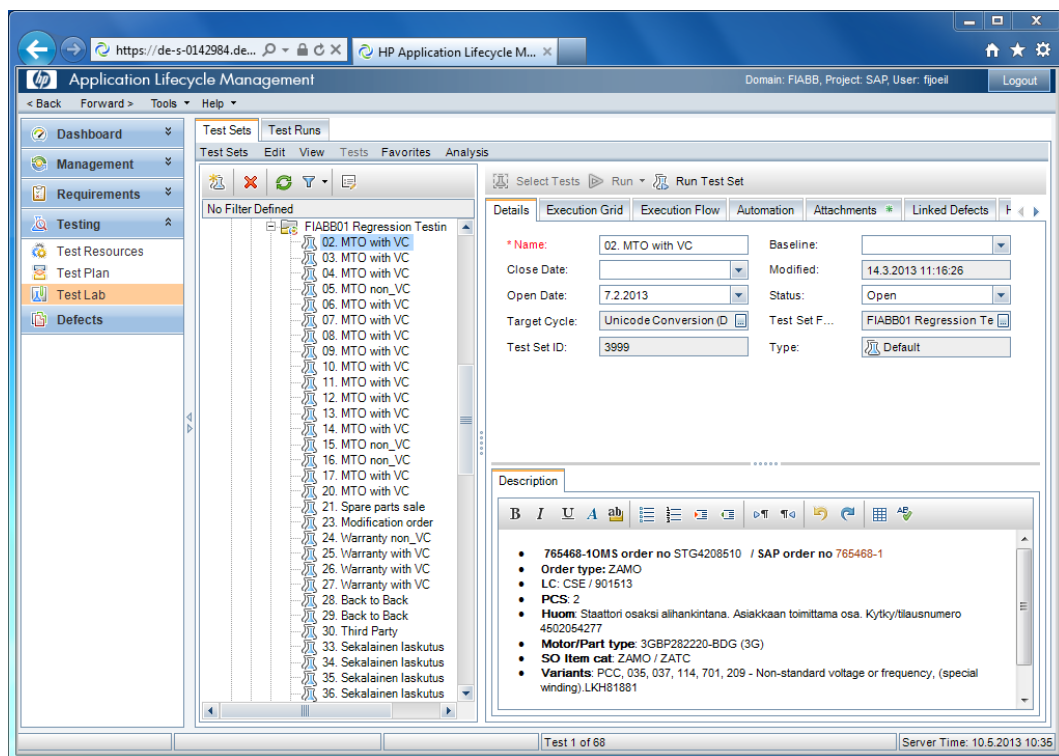


**Kuva 5.** Test Plan Tree - Yksittäisen testin lisääminen testikokonaisuuteen

Kuvassa 5 näkyy Execution Grid sekä Test Plan Tree. Näiden tarkoitus on kuvata testin kulkua. Test Plan Tree -valikko aukeaa kun painaa Select Tests -näppäintä. Test Plan Tree -valikosta siirretään suunnitellut testit testikokonaisuuteen. Kuvan 5 tapauksessa siirretään Engineering (lappvesa) -testi osaksi MTO with VC -testikokonaisuutta. Execution Grid -välilehdellä näytetään testikokonaisuuden looginen suoritusjärjestys.

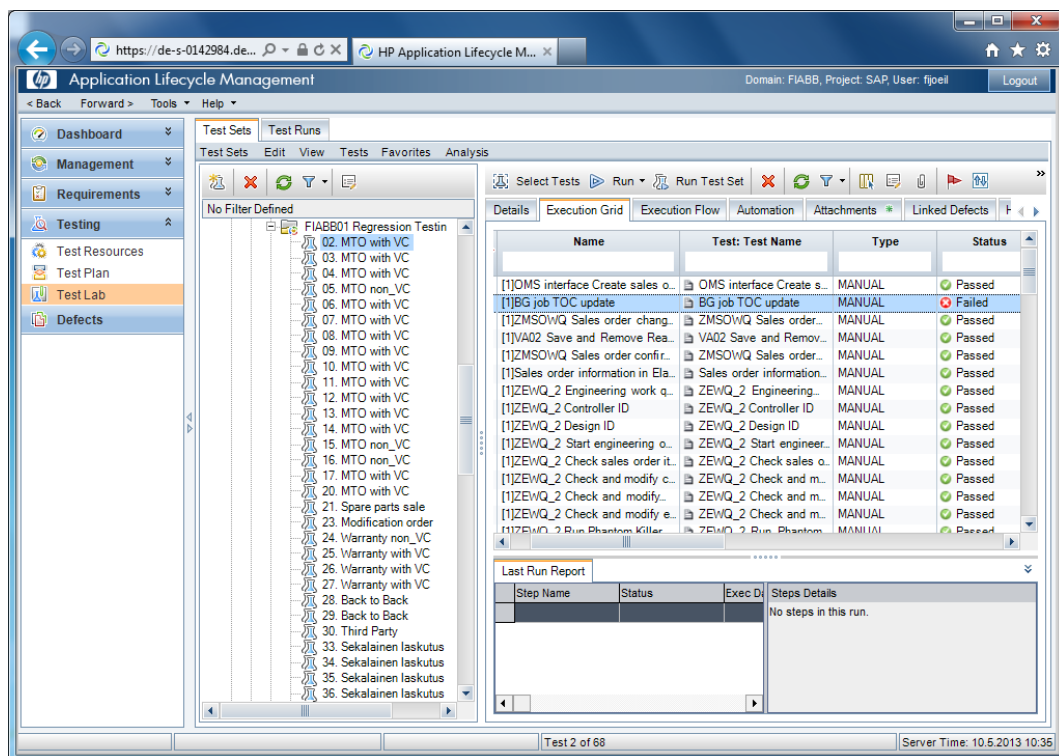
#### 4.4 Testin suorittaminen

Kun testi on rakennettu HP:n ALM Test Plan -sivulla, on testi valmis suoritettavaksi. Testin suoritus tapahtuu Test Lab -sivulla, joka aukeaa ALMin vasemmasta laidasta kohdasta Testing » Test Lab. Tämän jälkeen tulee valita oikea Release Cycle, eli SAP-järjestelmäpäivitys jota halutaan testata. Release Cyclen alta löytyy kaikki sen testin aikana tehtävät testikokonaisuudet. Automaatiotestauksen piirissä olevat testit eivät näy tämän ryhmän alla, vaan tässä näkyy ainoastaan maunaalitestin piirissä olevat.



## Kuva 6. Testin suorittaminen Test Planissa

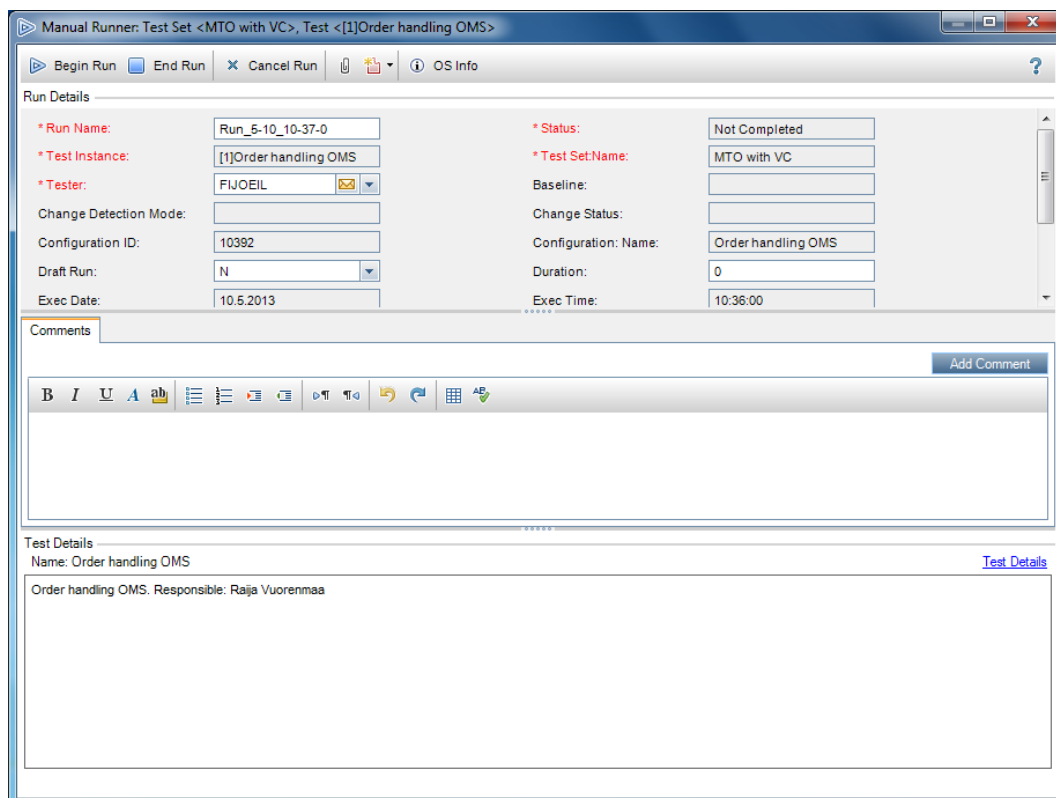
Kun oikea Release Cycle on valittuna, näkyvät kaikki sen aikana suoritettavat testikokonaisuudet. Testaus aloitetaan valitsemalla haluttu testikokonaisuus, mikä kuvassa 6 on 02. MTO with VC. Tämän testin tarkempi kuvaus näkyy Description-kentässä, josta testin tekijä saa tarvittavat tiedot, kuten kaupanumero ja SAP-tilausnumero. Kaikilla testaajilla on tähän kenttään kirjoitusoikeus. Testin aikana tähän kenttään lisätään tietoa ja huomautuksia, joiden halutaan olevan kaikkien testaajien nähtävillä.



**Kuva 7.** Execution Grid testin aikana

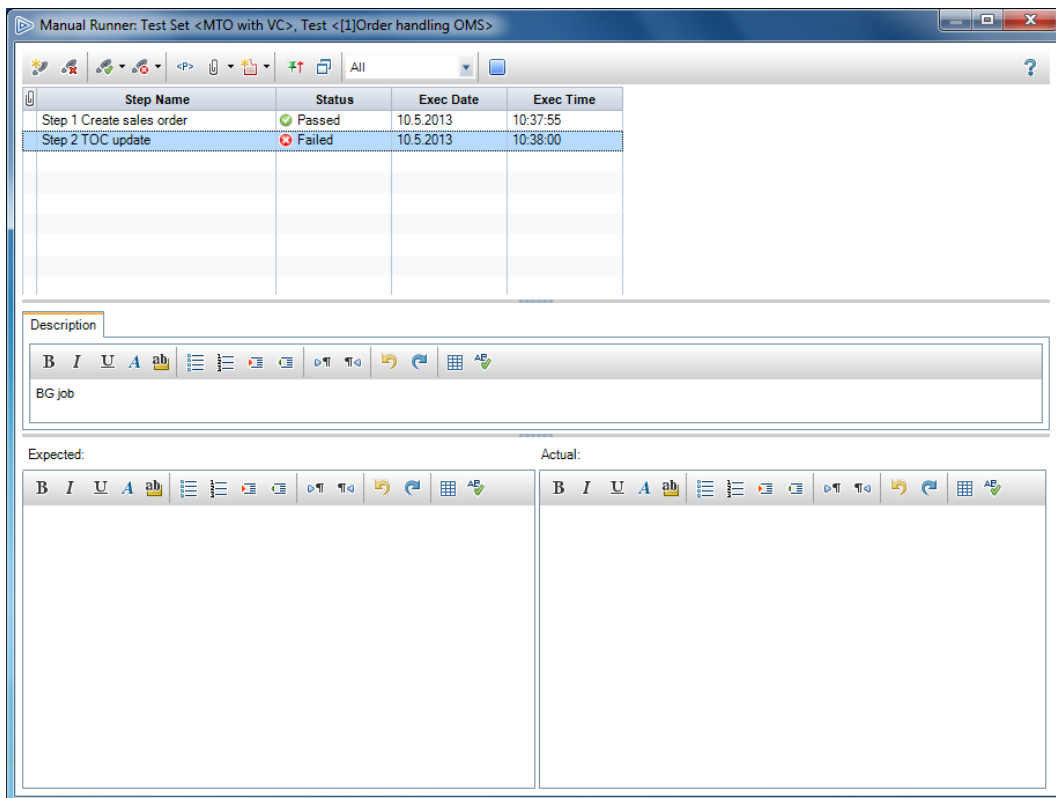
Kuvassa 7 näkyy valitun testin Execution Grid. Status-kentästä näkee nopeasti testin tilan. Kuvan esimerkissä BG job TOC update -testissä on havaittu jokin virhe, ja on tämän takia Failed-tilassa. Muut kuvassa 7 näkyvät testit on suoritettu onnistuneesti. Jos testiä ei ole vielä suoritettu, on se No Run -tilassa.

Testi aloitetaan valitsemalla Execution Grid -sivulta haluttu testi. Tämän jälkeen valitaan Run-valikosta Run with Manual Runner, tai vaihtoehtoisesti näppäinyhdistelmällä Ctrl+Alt+F9.



### Kuva 8. Testin ajo Manual Runnerilla

Kuvassa 8 näkyy Manual Runnerin ensimmäinen vaihe. Tässä ei muuta tarvitse tehdä kuin tarkastaa, että Test Instance on valitun testin mukainen. Kuvan 8 esimerkissä Test Instance on [x]Order handling OMS. Testin ajo aloitetaan painamalla työkalupalkista Begin Run -näppäintä. Jos Test instance ei ole oikein, voidaan testin ajo lakkauttaa painamalla Cancel Run -näppäintä.



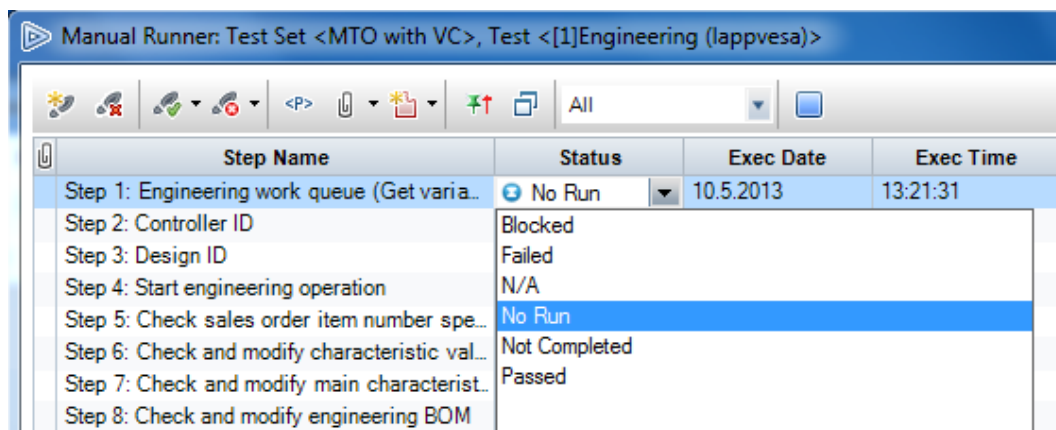
The screenshot shows a window titled "Manual Runner: Test Set <MTO with VC>, Test <[1]Order handling OMS>". It features a table with the following data:

Step Name	Status	Exec Date	Exec Time
Step 1 Create sales order	Passed	10.5.2013	10:37:55
Step 2 TOC update	Failed	10.5.2013	10:38:00

Below the table is a "Description" field containing the text "BG job". At the bottom, there are two empty text areas labeled "Expected:" and "Actual:", each with a rich text editor toolbar.

### Kuva 9. Manual Runner: Testin vaiheistus

Kuvassa 9 näkyy yhden henkilön peräkkäin tekemät testit. Order handling OMS -testi (Test Instancen nimi näkyy ikkunan otsikossa) sisältää kaksi vaihetta. Nämä vaiheet on luotu kappaleessa 4.3.2. Description-kentässä on lyhyt kuvaus vaiheesta, joka tässä tapauksessa on BG job. Status-kentässä voidaan yksittäiset testit kuitata tehdyiksi (**Kuva 9.**).



The screenshot shows the 'Manual Runner' window for a test set. The window title is 'Manual Runner: Test Set <MTO with VC>, Test <[1]Engineering (lappvesa)>'. Below the title bar is a toolbar with various icons. The main area contains a table with the following data:

Step Name	Status	Exec Date	Exec Time
Step 1: Engineering work queue (Get varia...	No Run	10.5.2013	13:21:31
Step 2: Controller ID	Blocked		
Step 3: Design ID	Failed		
Step 4: Start engineering operation	N/A		
Step 5: Check sales order item number spe...	No Run		
Step 6: Check and modify characteristic val...	Not Completed		
Step 7: Check and modify main characterist..	Passed		
Step 8: Check and modify engineering BOM			

**Kuva 10.** Testin mahdolliset Statukset

Kuvassa 10 näkyy kaikki ALMin mahdolliset statukset. Kaikki statukset voidaan nopeinten kuitata hyväksytyiksi painamalla työkalupalkista Pass All -näppäintä, tai näppäinyhdistelmällä Ctrl+A. Tämä kannattaa huomioida, sillä lähes kaikissa muissa ohjelmissa Ctrl+A valitsee kaikki. ALMissä painamalla Ctrl+A kuitenkin kaikki valitaan ja samalla merkataan tehdyiksi. Tämän kanssa on oltava tarkkana, sillä kaikki edelliset merkinnät muuttuu tällä komennolla Passed-tilaan. Näin myös, vaikka käyttäjä olisi käsin vaihtanut jonkun muun testin Failed-tilaan.

Kun kaikki testit on tehty, voidaan tilat tallentaa painamalla End Run -näppäintä, tai vaihtoehtoisesti näppäinkomennolla Ctrl+Q.

Jos yksikin testi merkataan Failed-tilaan, muuttuu koko testikokonaisuus Failed-tilaan. Kaikkien testien ja vaiheiden on oltava Passed-tilassa, jotta testikokonaisuus olisi Passed. Jos kaikkia testejä ja vaiheita ei ole tehty, on testikokonaisuus tilassa No Run tai Not Completed. N/A-tilaa tulee käyttää vain silloin, jos testi ei koske kyseistä testikokonaisuutta. N/A-tilan merkinnät eivät vaikuta testikokonaisuuksien tilaan muuten kuin, että ne mahdollistavat muiden testien suorittamisen loppuun.

#### 4.5 Testissä ilmenneiden ongelmien raportointi

Yksi ALMin tärkeimmistä ominaisuuksista on testissä ilmenneiden ongelmien raportointi. Näitä raportteja kutsutaan HP:n ALMissä Defecteiksi. ABB:n toteutuksessa ALMin kautta tehdyt Defectit, tai ongelmat, ovat nähtävissä ja muokattavissa

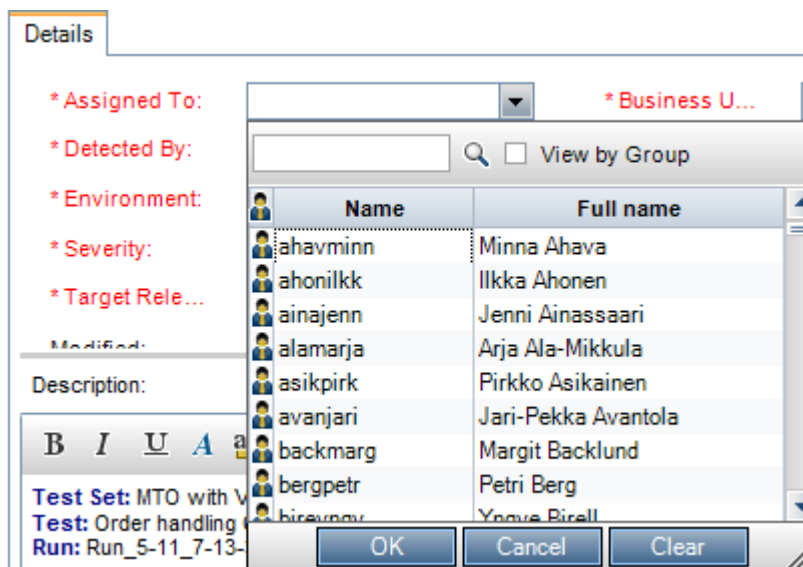
HP:n Quality Centerissä. SAPin kehittäjillä on ABB:lla pääsy HP:n Quality Centeriin, ja näin ollen myös pääsy ALMissä tehtyihin raportteihin. Testien aikana ongelmat voivat parhaimmessa tapauksessa olla korjattuna ja uudelleen testattavana jo muutaman tunnin kuluessa raportoinnista.

### Kuva 11. Uuden Defectin luominen

Kuvassa 11 näkyy New Defect -ikkuna. Ikkuna aukeaa työkalupalkista New Defect -näppäimestä, tai vaihtoehtoisesti Ctrl+N näppäinyhdistelmällä. Ohjelma syöttää automaattisesti osan tiedoista, kuten testin nimen ja vaiheen missä ongelma huomattiin. Koska tiedot tulevat automaattisesti, on ensisijaisen tärkeää raportoida ongelma oikeassa kohtaa testin vaiheistusta.

Testaajan on täytettävä ongelman lyhyt kuvaus Summary-kenttään. Description-kenttään voidaan kirjoittaa ongelmasta vielä tarkemmin. Jos ongelmasta voidaan ottaa näyttökuvaa (screen shot) tai PDF-tuloste, voidaan se tallentaa Attachments-sivulle. Kaikki kentät, jotka on merkattu punaisella tähdellä ovat pakollisia.

Assigned To -kenttään syötetään korjaajan tai vastuuhenkilön nimi. Jos vastuuhenkilön nimi ei ole tiedossa, voidaan ongelma raportoida omalle vastuuhenkilölle, joka puolestaan ohjaa ongelman eteenpäin.



**Kuva 12.** Defect: Assigned To -kenttä

Assigned To -kenttä näyttää automaattisesti käyttäjien ALM tunnuksia, jotka ovat samassa projektissa. Jos käyttäjätunnusta ei löydy listalta, voidaan ongelma raportoida omalle esimiehelle tai vastuuhenkilölle.



## 5 YHTEENVETO

Työ tehtiin ABB Oy Motors and Generatorsin laatuosastolle. Työn tavoitteena oli saada SAP-järjestelmäpäivitysten testauksiin työkalu, jolla voitaisiin helpottaa testien seurantaa ja ongelmien raportointia. Edeltävä toteutus oli Excel-tiedosto, jonne kirjattiin testien aikana löydettyjä virheitä ja ongelmia. Uusi toteutus on rakennettu HP:n Application Lifecycle Management -ohjelman päälle.

Työ aloitettiin keräämällä tietoa edeltävästä toteutuksesta. Testaajia haastateltiin, ja kerättiin tietoa siitä, mikä edeltävässä toteutuksessa oli hyvää ja mikä huonoa. Tiedon perusteella ruvettiin suunnittelemaan uutta toteutusta, jonka oli tarkoitus toimia HP:n Application Lifecycle Management -ohjelmalla. Työn suunnittelun ja toteutuksen apuvälineeksi valittiin Demingin PDCA-ympyrä. Tällä työkalulla työlle luotiin raamit, joiden sisällä projektia vietäisiin eteenpäin.

Demingin PDCA-ympyrän vaiheiden avulla projektia lähdettiin toteuttamaan. Ensimmäinen vaihe oli suunnittelu, johon koko projektin aikana käytettiin eniten aikaa.

Suunnittelun pohjalta tehtiin ensimmäinen toteutus. Tämä toteutus todettiin ensimmäisten testien jälkeen liian monivaiheiseksi, jonka takia jouduttiin palaamaan takaisin suunnittelun pariin. Testaajilta saatujen kommenttien perusteella voitiin todeta se, että monivaiheisuus hidasti testaajia, ja että yhden testaajan tekemät testit haluttiin ryhmitellä yhdeksi kokonaisuudeksi.

Uudelleen suunniteltaessa testien mahdollinen ryhmittäminen otettiin huomioon. Ensimmäisessä versiossa yksi testaaja joutui pahimmassa tapauksessa kuittaamaan yli kaksikymmentä erillistä testiä tehdyksi. Jokainen kuittaus vaati muutaman napin painalluksen, eli yhteensä tuli pahimmassa tapauksessa yli viisikymmentä napinpainallusta. Uudelleen suunnittelun ja toteutuksen myötä testaajat voivat nyt kuitata kaikki saman käyttäjän testit tehdyiksi vain muutamalla napinpainalluksella, ja aikaa säästyy itse testin suorittamiseen.

Eräs työn tavoitteista oli toimiva ongelmien raportointi. HP:n ALMin yksi tärkeimmistä ominaisuuksista on juuri ongelmien raportointi ja seuranta. Tällaista ominaisuutta ei edeltävässä Excel-toteutuksessa ollut. Uudessa toteutuksessa käytetään defektien luomista suoraan ALMista hyväksi. Tämä helpottaa löydettyjen virheiden raportointia huomattavasti. Raporttiin voidaan myös liittää mahdollisia kuvakaappauksia ja PDF-tulosteita. ALMiin tehdyn rakenteen ansiosta myös virheiden aiheuttaja on helpompi paikallistaa, sillä tehty SAP-transaktio ja tehdyt toimenpiteet näkyvät raportin kuvauksessa.

ALMiin saatiin projektin aikana rakennettua vaatimuksia vastaava toteutus. Haasteet, jotka projektin kartoitusvaiheessa määriteltiin, voitiin onnistuneesti kiertää hyvällä suunnittelulla. Lopullinen ALM-toteutus on yksinkertainen, kuitenkin sisältäen kaikki tärkeät tiedot.

Projektia voitaisiin tulevaisuudessa kehittää niin, että jokainen testaaja ylläpitäisi omia testikokonaisuuksiaan. Kun moottoreihin ja tilauksiin tehdään muutoksia, olisi jokaisen testaajan muokattava omia testikokonaisuuksiaan.

Muut ABB:n yksiköt hyötyisivät myös Vaasassa tehdystä ALM-toteutuksesta. Tämä kuitenkin vaatisi kaikkien yksiköiden testikokonaisuuksien kirjaamisen ALMiin. Vaasassa käytettyjä testikokonaisuuksia ei voida käyttää muualla, sillä testien kuvaukset ja käytetyt SAP-transaktiot ovat yksikkökohtaisia.

## 6 JOHTOPÄÄTÖKSET

ALM-toteutuksen suhteen annettiin suhteellisen vapaat kädet. Tämän takia työ oli alussa hyvinkin haastava. Pitkän perehdytyksen ja koulutuksen ansiosta työn kuva kuitenkin rupesi aukeamaan ensimmäisten kuukausien aikana.

Suunnitteluvaihe oli tämän työn osalta ylivoimaisesti eniten aikaa vievä. Suunnittelua hieman hidasti myös se, että työn tilaaja ei vielä alussa tiennyt tarkalleen mitä halusi. Tarkoitukset ja vaatimukset selvisi vasta ensimmäisen toteutuksen aikana. Alun haasteiden jälkeen kuitenkin saatiin erittäin hyvin toimiva toteutus, johon molemmat osapuolet voivat olla tyytyväisiä.

Toteutus ALMiin onnistui varsin hyvin. Mikään ei estä sitä, että edeltävä Excel-toteutus voidaan kokonaan korvata uudella HP ALM -toteutuksella. Seuraava kokonainen SAP-järjestelmäpäivitys tapahtuu vasta tämän työn palautuksen jälkeen, joten nähtäväksi jää tuleeko ALMin rakenteeseen muutoksia vielä tulevaisuudessa. Demingin PDCA-ympyrä tarjoaa kuitenkin jatkuvan kehityksen työkalun, joten mikään ei estä jatkuvan kehityksen myös ALM-toteutuksen suhteen.

HP:n Application Lifecycle Management -ohjelma soveltuu erittäin hyvin tämän tyyppisten testien seurantaan ja ongelmien raportoimiseen. Ei löydy syytä miksi tätä ohjelmaa ei voisi vastaavanlaisessa toteutuksessa käyttää myös tulevaisuudessa. Tämä testauksen apuvälineeksi tarkoitettu ohjelma edesauttaa isojen järjestelmäpäivitysten läpiviemisen ja testaamisen onnistuneesti. ABB:llä tämä tarkoittaa sitä, että testaukseen käytetty aika edesauttaa tehtaan pyörimistä. ALMilla toteutetut testaukset toivottavasti vähentävät järjestelmäpäivityksien aiheuttamien ongelmien kulminoitumisen aina tehtaan tuotannon pysähtymiseen asti.

## LÄHTEET

- /1/ ABB Ydinliiketoiminnat. 2012. ABB Oy. Viitattu 2.5.2013.  
<http://www.abb.fi/cawp/fiabb251/5b3b47abc1e9e75dc2256b20003f96db.aspx>
- /2/ ABB Suomessa. 2012. ABB Oy. Viitattu 2.5.2013.  
<http://www.abb.fi/cawp/fiabb251/0b5e2755355c156dc12579bb003910a4.aspx>
- /3/ Henkilöstöopas: Tervetuloa menestymään kanssamme. 2012. ABB Oy. Ei julkisesti saatavilla.
- /4/ Application Test Software – Development Lifecycle Management. Hewlett-Packard. 2013. Viitattu 5.5.2013. <http://www8.hp.com/us/en/software-solutions/software.html?compURI=1174315#UYZkwZWjRUM>
- /5/ Mitä on ALM? ContriByte Oy. 2012. Viitattu 5.5.2013. <http://www.contri-byte.fi/fi/alm/mita+on+alm/>
- /6/ Schulmeyer, Gordon G. 2007. Handbook of Software Quality. 35–37, 45–49.