

**Testaussuunnitelma**

Lauri Girsén

Laura Huotari

Noora Jokinen

Marja Jämbeck

Juuso Kyyrä

Web-palvelujen määrittely ja suunnittelu- kurssin harjoitustyö

Syksy 2017

Tieto- ja viestintätekniikan koulutusohjelma

Tekniikan ja liikenteen ala

|  |  |  |  |
| --- | --- | --- | --- |
| Versionumero | Päivämäärä | Tekijä | Tehdyt muutokset |
| versio 0.1 | 14.11.2017 | Marja | Testaussuunnitelman ensimmäinen versio |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Versionhallinta

Sisältö

[1. Johdanto 4](#_Toc498399013)

[1.1. Dokumentin tarkoitus ja kattavuus 4](#_Toc498399014)

[1.2. Määritelmät, termit ja lyhenteet 4](#_Toc498399015)

[1.3. Viitteet muihin dokumentteihin 4](#_Toc498399016)

[2. Testauksen kohde ja tavoitteet 5](#_Toc498399017)

[2.1. Testauksen kohde 5](#_Toc498399018)

[2.2. Testauksen tavoitteet 5](#_Toc498399019)

[3. Testausympäristö 6](#_Toc498399020)

[3.1. Laitteisto 6](#_Toc498399021)

[3.2. Ohjelmisto 6](#_Toc498399022)

[3.3. Testiajurit 6](#_Toc498399023)

[3.4. Tynkämoduulit 7](#_Toc498399024)

[4. Testauksen organisointi ja raportointi 7](#_Toc498399025)

[4.1. Testauksen organisointi ja aikataulu 7](#_Toc498399026)

[4.2. Testauksen vastuualueet 7](#_Toc498399027)

[4.3. Testauksen raportointi 7](#_Toc498399028)

[4.4. Yksittäisen raportin sisältö 8](#_Toc498399029)

[4.5. Testausbudjetti 9](#_Toc498399030)

[5. Testauksesta yleisesti 9](#_Toc498399031)

[5.1. Testauksen kriteerit 9](#_Toc498399032)

[5.2. Virheiden luokittelu 9](#_Toc498399033)

[5.3. Riskit ja niiden hallinta 10](#_Toc498399034)

[5.4. Korjaukset ja niiden hallinta 10](#_Toc498399035)

[6. Testaustasot 11](#_Toc498399036)

[6.1. Yksikkö / moduulitestaus 11](#_Toc498399037)

[6.2. Integrointitestaus 11](#_Toc498399038)

[6.3. Järjestelmätestaus 11](#_Toc498399039)

[6.4. Hyväksymistestaus 12](#_Toc498399040)

[7. Erikoistilanteet 12](#_Toc498399041)

[8. Ei-testattavat toiminnot 12](#_Toc498399042)

[9. Testauksen arviointi 12](#_Toc498399043)

[9.1. Arvio testauksen kattavuudesta 12](#_Toc498399044)

[9.2. Arvio testauksen tuloksista 12](#_Toc498399045)

# 1. Johdanto

## 1.1. Dokumentin tarkoitus ja kattavuus

Tässä dokumentissa kuvataan Platz: palveluiden osto- ja myyntipalvelun testaussuunnitelma ja toteutettavan järjestelmän osa-alueet. Projektin suunnitteluvaiheessa tuotetaan n. 15 testitapausta järjestelmän keskeisistä käyttötapauksista johtamalla. Testitapauksia tarkennetaan projektin aikana suunnittelun yksityiskohtien tarkentuessa kattamaan laajemmin tuotettavaa järjestelmää.

## 1.2. Määritelmät, termit ja lyhenteet

**Testaustaso** Määrittelee mille järjestelmän tasolle testaus ulottuu. Alimmalla tasolla testataan yksittäisiä luokkia tai metodeita; ylimmällä käytettävyyttä ja järjestelmän toimivuutta kohdeympäristössä.

**Testauskierros** Testauskierroksella käydään läpi määritellyt testit. Näiden tulokset raportoidaan ja päätetään suoritettavista korjauksista. Kun korjaukset on tehty, testataan uudestaan kunnes todetaan virheet korjatuiksi tai saavutetaan vaadittu hyväksymisraja.

**Hyväksymisraja**  Järjestelmä täyttää sille asetetun vaatimusmäärittelyn ja testikierroksilla on löydetty kaikki järjestelmän toimintaa haittaavat virheet.

## 1.3. Viitteet muihin dokumentteihin

Testauksen kannalta olennaisia dokumentteja ovat:

* vaatimusmäärittely
* käyttötapauskuvaus
* testitapaukset.

# 2. Testauksen kohde ja tavoitteet

## 2.1. Testauksen kohde

Platz – palveluiden osto- ja myyntipalvelun järjestelmät sekä niihin liittyvät tukitoiminnot.

## 2.2. Testauksen tavoitteet

Testauksen tavoitteena on varmistaa, että järjestelmä vastaa sille asetettuja tavoitteita. Testauksella pyritään karsimaan järjestelmän virheitä mahdollisimman aikaisessa vaiheessa. Testaus pyritään suorittamaan siten, että kaikki osa-alueet tulevat katettua, mutta samaa asiaa ei testata turhaan moneen kertaan.

Kun virheitä löydetään, virheet raportoidaan ja korjataan. Korjaukset tarkistetaan suorittamalla epäonnistunut testi uudestaan. Kun yhden testaustason kaikki suunnitellut testit on suoritettu hyväksytysti, voidaan siirtyä ylemmälle tasolle. Testaus päättyy, kun ylinkin testaustaso on suoritettu hyväksytysti.

Jokaisen testauskierroksen jälkeen järjestelmä on käyttökelpoinen, mutta toimintojen määrä kasvaa kierros kierrokselta.

Testausnäkökulmia:

* Käytettävyys
* Käytettävyystestauksella voidaan varmistaa, että järjestelmä on kaikkien kohderyhmien käytettävissä ja käyttöliittymä täyttää muut käytettävyyskriteerit (määritelty vaatimusmäärittelyssä).
* Ylläpidettävyys
  + Ylläpidettävyys liittyy järjestelmän kompleksisuuteen, mutta ei ole suoraan laadullisesti arvioitavissa. Ylläpidettävyyteen liittyvät järjestelmäasennusten- ja päivitysten helppous sekä järjestelmälle tehtävien huoltotöiden vaatima aika.
* Siirrettävyys
  + Kuinka helposti järjestelmä on siirrettävissä ympäristöstä toiseen; millä tavoin järjestelmän integrointikomponentit on rakennettu tai kuinka paljon järjestelmässä on sovelluspalvelimista riippuvia osia.
* Modulaarisuus ja hajautus
  + Kuinka helposti järjestelmä on laajennettavissa erillisillä komponenteilla ja kuinka helposti järjestelmä on skaalattavissa (load balancing, yms.) kun tarvitaan laajennusta.
* Tietoturva
  + Järjestelmästä voidaan etsiä yleiset tietoturva-aukot ja www-palveluita koskevat haavoittuvuudet.
* Suorituskyky
  + Järjestelmän suorituskyky on testattavissa esimerkiksi kuormitustesteillä.

# 3. Testausympäristö

## 3.1. Laitteisto

Testaus suoritetaan arkkitehtuurisuunnitelman mukaisessa tulevassa tuotantoympäristössä. Tarkennetaan myöhemmin.

## 3.2. Ohjelmisto

Palvelinympäristö, jonne palvelu pystytetään, on dokumentoitu arkkitehtuurisuunnitelmassa.

## 3.3. Testiajurit

Yksittäisten osa-alueiden testaukseen voidaan tehdä testiajurit, joilla kyseiset alueet voidaan testata itsenäisesti.

## 3.4. Tynkämoduulit

Testausvaiheessa vielä toteuttamatta olevat järjestelmän osat mallinnetaan tynkämoduuleilla (stubeilla).

# 4. Testauksen organisointi ja raportointi

## 4.1. Testauksen organisointi ja aikataulu

Kunkin osa-alueen vastuuhenkilö tekee kyseistä osa-aluetta koskevat testitapaukset ja suorittavat ne osa-alueen valmistuttua. Kaikkien testitapauksien tulokset raportoidaan. Integrointitestauksen suorittamisessa ovat mukana kaikki osa-alueiden vastuuhenkilöt, joiden osa-alueet osallistuvat testaukseen.

## 4.2. Testauksen vastuualueet

Testauksen vastuu-alueet jaetaan projektiryhmän jäsenten kesken.

## 4.3. Testauksen raportointi

Testauksen aikana pidetään testauspäiväkirjaa, johon kaikki suoritetut testaukset sekä niiden tulokset kirjataan. Löydetyt virheet raportoidaan muille toteutuksessa mukana oleville jäsenille. Virheen korjaa virheellisen osa-alueen vastuuhenkilö, tai projektin jäsenet yhteistyönä. Korjauksen testaa virheen löytäjä.

Virheet raportoidaan virheenraportointilomakkeelle, johon täytetään testitapaus, testaaja, testausaika, virheluokka, vakavuus, jne.

Jokaisen testauksen yhteydessä tulee laatia seuraavat dokumentit:

* Testauspäiväkirja
* Testilokit (esittää testausistunnon kulun)
* Virheaportit (testausistunnossa löydetyt virheet testauksen kohteessa)
* Testausraportti (aineisto ja tulokset testauksesta)

## 4.4. Yksittäisen raportin sisältö

Yksittäisestä raportista pitää käydä selville:

**Päivämäärä** Päivämäärä, jolloin testattu ja päivämäärä, jolloin korjattu.

**Testannut** Testauksen suorittaja (nimi, e-mail).

**Korjannut** Virheen korjannut henkilö tai henkilöt (nimi, e-mail).

**Kohdealue** Kohdealue, jolla ongelma esiintyy (sivu, komponentti, tms.). Kohdealueesta on eroteltava, liittyykö ongelma mielipiteisiin, käyttöliitymän ulkoasuun, toimintaan tai tietosisältöön.

**Testialusta** Millä/missä ongelma on todettu; esim. käyttöliittymissä selain ja selainversio.

**Tarkka sijainti** <URL:n> nimi mikäli testataan käyttöliittymää tai tarkka polku komponenttiin/luokkaan.

**Tiivistelmä** Lyhyt kuvaus ongelman sisällöstä.

**Kuvaus ongelmasta**  Yksityiskohtainen kuvaus ongelman esiintymisestä. Tämän pitäisi sisältää niin paljon taustatietoja kuin on saatavilla.

**Toistettavuus** Kuinka usein ongelma voidaan toistaa. Arvot ovat ”joka kerta”, ”satunnaisesti”, ”ei ole yritetty toistaa” ja ”ei saatu toistettua.

**Vaiheet toistamisessa** Kuvataan askelin (klikkauksin tai toimenpitein) kuinka ongelma saadaan toistettua.

**Saadut tulokset** Mitä toiminnasta seurasi.

**Odotetut tulokset** Mitä toiminnasta olisi pitänyt seurata.

## 4.5. Testausbudjetti

Mahdollinen testaukseen käytettävä budjetti määritellään myöhemmin.

# 5. Testauksesta yleisesti

## 5.1. Testauksen kriteerit

* Hyväksymiskriteerit ja hylkäämiskriteerit
  + Hyväksytään, kun järjestelmä vastaa vaatimusmäärittelyä. Vaatimuksia voidaan tarkentaa tarvittaessa.
* Testauksen keskeyttäminen, testauksen jatkaminen, testauksen lopettaminen
* Testaus päättyy, kun projektiryhmä on päättänyt testauksen.

## 5.2. Virheiden luokittelu

Esimerkiksi:

Virheiden luokittelu käyttäjän kokeman vaikutuksen mukaan:

**Vähäinen**: Virheellä on vain esteettinen vaikutus (sarakkeen verran sivussa oleva porrastus raportissa) tai järjestelmän käyttäytyminen on ihmisen kannalta kummallista (esim. nimet katkaistaan tulostuksessa satunnaisesti).

**Estävä**: Järjestelmä ei suostu käsittelemään kelvollista syötettä tai tuloste on harhaanjohtava ja virhe huonontaa suorituskykyä.

**Vakava**: Järjestelmä kadottaa tapahtuman tai tapahtuma muuttuu toiseksi.

**Erittäin vakava**: Erittäin vakavia virheitä tapahtuu jatkuvasti (eikä vain harvinaisten erikoistapausten kohdalla).

**Hyvin vakava**: Järjestelmän tietokannan eheys hajoaa siten, että sen kuntoon laittaminen on vaikeaa. Järjestelmän suoritus päättyy hallitsemattomasti itsekseen tai järjestelmä tuottaa virheellistä dataa.

**Erittäin vakava**: Järjestelmä aiheuttaa virheitä muissa järjestelmissä tai ympäristössään.

Virheiden luokittelu vaihemallin kannalta katsottuna:

* Syy virheellisessä vaatimusmäärittelyssä
* Syy virheellisessä suunnittelussa
* Syy ohjelmoinnissa

## 5.3. Riskit ja niiden hallinta

Tässä kuvataan testaukseen vaikuttavat riskitekijät. Kullekin riskitekijälle on suunniteltava riskinhallintatoimenpiteet.

## 5.4. Korjaukset ja niiden hallinta

Löydetyistä virheistä määritellään niiden virhestatukset. Laadunvarmistuksessa määritellään kunkin virheen vakavuuden perusteella miten löytyneisiin virheisiin reagoidaan. Reagointitapoja ovat:

**Uusi virhe** Virhe on uusi ja ennalta raportoimaton tai vanhan virheen uusi ilmentymä.

**Korjattu** Tunnistettu ongelma on korjattu ja korjaus on siirretty järjestelmään.

**Ei korjata** Kuvattua ongelmaa ei korjata, koska ongelma ei johdu virheestä vaan on osa järjestelmän toimintatapaa.

**Raportoitu aikaisemmin** Ongelma on selvitetty jo aikaisemmin. Ilmoitettava ongelman numero ja ajankohta.

**Ei voida toistaa** Ongelma on ilmeisesti kohdattu ja jotkin tulokset vastaavat tätä, mutta ongelmaa ei voida toistaa testiympäristössä. Ongelma jätetään avoimeksi ja tutkitaan myöhemmin.

**Siirretty** Ongelma ei johdu tuotetusta järjestelmästä vaan esimerkiksi jostakin taustajärjestelmästä. Ongelmaan ei voida reagoida järjestelmän toteutuksessa tai ratkaisu riippuu muiden järjestelmien toimittajien korjauksista.

# 6. Testaustasot

## 6.1. Yksikkö / moduulitestaus

Yksikkötestauksessa (unit testing) keskitytään yhden osa-alueen toiminnan tarkasteluun. Toimintaa verrataan vaatimusmäärittelyyn. Testaus voidaan suorittaa heti osa-alueen valmistuttua.

## 6.2. Integrointitestaus

Integrointitestauksessa keskitytään projektin osa-alueiden välisiin rajapintoihin palvelua pystyttäessä. Testataan erillisten komponenttien välistä vuorovaikutusta sekä kommunikointia ja toiminnallisuutta niiden välillä. Testaus voidaan suorittaa heti, kun palvelua pystytetään ja osa-alueita yhdistetään toimivaksi kokonaisuudeksi.

## 6.3. Järjestelmätestaus

Järjestelmätestaus suoritetaan viimeisenä ennen palvelun lopullista julkaisua. Sen ideana on testata, toimiiko palvelu kuten määrittely- ja suunnitteluvaiheessa on suunniteltu. Tarkoituksena löytää ja korjata mahdollisimman paljon määrittelyn ja luodun systeemin välisiä ristiriitoja ja virheitä. Testauksen kohteena on koko valmis järjestelmä, ja saatuja tuloksia verrataan aiemmin luotuun määrittelydokumentaatioon. Sovellusta tarkastellaan loppukäyttäjän näkökulmasta käyttäen apuna erilaisia käyttötapauksia. Tavoitteena löytää virheitä koko järjestelmän toiminnallisuudesta. Järjestelmätestaus suoritetaan mahdollisuuksien mukaan aidossa ympäristössä.

## 6.4. Hyväksymistestaus

Mahdollinen loppukäyttäjän suorittama testaus.

# 7. Erikoistilanteet

Järjestelmän toiminnasta eristetään joitakin keskeisiä erikoistilanteita, jotka vaativat tarkkaa testausta.

Näitä ovat esimerkiksi

* järjestelmän osan tai taustajärjestelmän kaatuminen
* yhteyksien pettäminen kahden eri liikkuvuusjärjestelmän välillä
* yhteyksien pettäminen liikkuvuusjärjestelmän ja muun järjestelmän osan välillä integrointirajapinnan välillä
* järjestelmän kaatumisesta (osan kaatumisesta) aiheutuvien virheilmoitusten näyttäminen ja käyttäjän informoiminen.

# 8. Ei-testattavat toiminnot

Ei-testattavat toiminnot ovat sellaisia osa-alueita, joille ei voida suorittaa kattavaa testausta tai niiden yksityiskohtainen testaaminen ei ole järjestelmän toiminnan kannalta merkityksellistä.

# 9. Testauksen arviointi

## 9.1. Arvio testauksen kattavuudesta

## 9.2. Arvio testauksen tuloksista