

Etunimi Sukunimi (opiskelijanumero ja ryhmä) Kirjasinkoko 16

Tehtävä – 1

Testaussuunnittelu ja testitapaukset

Tehtävä
Ohjelmistotestaus

2023



**Kaakkois-Suomen
ammattikorkeakoulu**

SISÄLLYS

1	JOHDANTO.....	4
2	TESTAUSSUUNNITELMA	4
2.1	Kuvaus testattavasta ohjelmasta	4
2.2	Aikataulu	4
2.3	Testauksen valmistumis kriteerit.....	5
2.4	Henkilöstö.....	5
2.5	Suunnitellut testauksen työvaiheet	5
2.6	Työkalut ja menetelmät.....	6
2.7	Testausympäristö	6
2.8	Testauksen laajuus.....	6
3	TESTITAPAUKSET	7
3.1	Testitapaus 1	8
3.1.1	Testitapauksen nimi ja kuvaus	8
3.1.2	Testitapaukseen liittyvät vaatimusmäärittelyt	8
3.1.3	Ennakkovaatimukset	8
3.1.4	Automatisointi.....	8
3.1.5	Testivaiheet.....	8
3.2	Testitapaus 2	9
3.3	Testitapaus 3	9
3.4	Testitapaus 4	10
3.5	Testitapaus 5	10
3.6	Testitapaus 6	10
3.7	Testitapaus 7	10
3.8	Testitapaus 8	10
3.9	Testitapaus 9	10
3.10	Testitapaus 10	10
4	ARVIO OMASTA OPPIMISESTA	11

LÄHTEET.....	12
--------------	----

1 JOHDANTO

[Kursiivilla olevat osat ovat ohjeita ja esimerkkejä. Ne on tarkoitus poistaa ja korvata omalla tekstillä.]

[Kuvaa tähän kappaleeseen yleisesti mitä olet tehtävässä tehnyt. Huomaa myös viimeisenä oleva kappale, jossa pyydetään arvioimaan omaa oppimista.]

2 TESTAUSSUUNNITELMA

[Tähän kappaleeseen jokainen laatii yksinkertaisen testaus suunnitelman valitsemalleen ohjelmalle.]

[Testaus suunnitelmalle kuten muillekin ohjelmistokehityksen ja ohjelmistotestauksen dokumentaatiolle löytyy erilaisia standardeja. Nämä kuitenkin vain ohjaavat tekemistä. Viimeisin testauksen dokumentaatioon liittyvä standardi on ISO/IEC/IEEE 29119-3:2021. Lyhyt ja luotettavan näköinen kuvaus löytyy wikipediasta: https://en.wikipedia.org/wiki/ISO/IEC_29119 . Itse standardi on maksullinen. Jokainen organisaatio tekee tarpeidensa mukaiset dokumentit ja esimerkiksi sertifiointit voivat määrittää lopullista sisältöä. Testaus suunnitelma ei myöskään ole dokumentti, joka tehdään ja unohdetaan. Sitä on hyvä päivittää ja tarkastella koko projektin ajan. Projektin koon mukaan suunnitelmia voi olla myös useita. Laajassa pitkään jatkuvassa tuotekehityksessä uudet ominaisuudet ja uudet asiakkaat ovat usein uusia projekteja, joille laaditaan omat testaus suunnitelmansa. Olen valinnut tähän suppean listan mielestäni oleellisia asioita pienen projektin kannalta.]

2.1 Kuvaus testattavasta ohjelmasta

[Kuvaus testattavasta ohjelmasta. Tässä voidaan viitata määrittelyihin. Testauksen ja testitapausten kannalta oleelliset asiat ja huomiot on listata tässä.]

2.2 Aikataulu

[Koska testaus on tarkoitus aloittaa, kuinka paljon aikaa eri vaiheiden tekemiseen kuluu ja koska testauksen on tarkoitus olla valmis. Tämä voidaan jättää kurssilla tyhjäksi.]

2.3 Testauksen valmistumis kriteerit

[Tähän kohtaan kirjataan koska tuote tai uusi versio on testauksen näkökulmasta valmis. Usein mittarina käytetään avoimien vikojen määrää. "Testaus on valmis, kun tuotteessa ei ole yhtään avointa kriittistä vikaa ja kaikki suunnitellut testit on kertaalleen ajettu onnistuneesti."]. Voidaan myös asettaa vaatimuksia sille, kuinka paljon vikoja on löydetty ja korjattu. Optimi-tilanteessa vikoja löydetään ensimmäisistä versioista vähän ja toiminnallisuuden lisääntyessä vikojen määrä kasvaa. Jossain vaiheessa tuotteen versiot paranevat ja löydettyjen vikojen määrä lähtee laskuun.

2.4 Henkilöstö

[Lista testaajista ja vastuu alueista. Tämä on jo muutaman hengen tiimissä oleellinen asia, mutta kurssi tehtävässä voidaan jättää tyhjäksi.

2.5 Suunnitellut testauksen työvaiheet

[Tähän kohtaan listataan testauksen työvaiheita. Pienessä projektissa vaiheita voi olla vähemmän. Yksikkötestaus on yleensä kehittäjien vastuulla. Se on kuitenkin hyvä mainita kriteerinä seuraavien testivaiheiden aloittamiselle.]

[Vaiheita ovat esimerkiksi Integraatiotestaus, järjestelmätestaus, hyväksyntätestaus, tuotantotestaus.]

[Otetaan esimerkkinä kännykkään tehty laskin sovellus. Integraatiotestaus voi tarkoittaa ohjelman asennusta ja käynnistämistä. Järjestelmätestausvaiheessa testataan, miten sovellus toimii, hyväksyntätestaus olisi vaihe, jossa räätälöidyn laskimen tilannut taho testaa toiminnallisuutta omasta näkökulmastaan. Tuotantotestaus voisi tarkoittaa ohjelman verkkokauppaan viemistä ja asennuksen kokeilemistä verkkokaupasta kännykkään. Savutestaus (smoke) termiä voidaan käyttää monessa kohdassa. Usein savutestaus tehdään heti tuotantoonviennin jälkeen. Se on nopea testi, jolla varmistetaan asennuksen onnistuminen. Jos siinä havaitaan ongelmia, voidaan vielä tehdä nopea takaisin veto (roll back).]

[Työvaiheille kirjataan aloitus kriteerit:

[Esim. Integraatio testaus voidaan aloittaa, kun uusi versio on valmis ja kaikki yksikkötestit ajettu ja läpäisty.]

2.6 Työkalut ja menetelmät

[Usein tarvitaan erilaisia ohjelmia varsinaisen testauksen suorittamiseen sekä ohjelmia, joilla hallitaan testausta ja raportointia.

Tähän kohtaan listataan web sovelluksen tapauksessa esimerkiksi erilaiset selaimet ja voidaan tehdä rajauksia selainten versioista vaatimusmäärittelyihin perustuen. Samoin automaatio työkalut listataan tässä.

Testauksen hallintaan liittyen mainitaan vianhallintatyökalut ja kuvataan se, miten testitapaukset säilytetään.

2.7 Testausympäristö

[Mitä testattava ohjelma vaatii toimiakseen? Laitteistot, ohjelmistot (käyttöjärjestelmä, jne.), mahdolliset yhteydet (integraatiot) muihin järjestelmiin. Verkkokauppa sovelluksessa integraatiota voidaan tarvita esimerkiksi maksujärjestelmään. On kuitenkin hyvä huomata, että lopullinen testaus voidaan tehdä vasta tuotantoympäristössä. Alemmilla tasoilla rajapintojen toimivuutta testataan testijureilla (stub) tai mahdollisia testijärjestelmiä käyttäen.

2.8 Testauksen laajuus

[Mitä testataan ja erityisesti mitä ei testata. Verkkokaupan toimivuuden tarkistus voidaan rajata vaikka selaimiin ja mainita erikseen, että verkkokauppaa ei testata kännyköillä. Tällaiset rajaukset pitää tietysti aina perustella.]

3 TESTITAPAUKSET

[Lisää tähän suunnittelemasi testitapaukset (vähintään 10 kappaletta)]

[Testitapaukset lisätään joko kuvankaappauksina SquashTM ohjelmasta esimerkin mukaan tai kirjaamalla testitapaukset mallin mukaan tekstinä. Aloita jokainen testitapaus omalta sivultaan. Nämä voi kirjata suomeksi tai englanniksi. Suomenkieliset käännökset ovat usein hiukan kömpelöitä ja IT alalla asioita tehdään usein kansainvälisessä ympäristössä.]

[Seuraavilla sivuilla on mallit testitapausten lisäämiseen kirjaamalla tähän asiakirjaan tai kuvakaappausten lisäämiseen SquashTM ohjelmasta.]

3.1 Testitapaus 1

3.1.1 Testitapauksen nimi ja kuvaus

Nimi: Kirjautuminen

Kuvaus: Testitapaus testaa, että käyttäjä pystyy kirjautumaan järjestelmään turvallisesti.

3.1.2 Testitapaukseen liittyvät vaatimusmäärittelyt

- *R1. Kirjautuminen*
- *NFR1. Salasana on piilotettu*

3.1.3 Ennakkovaatimukset

- *Järjestelmässä on testikäyttäjä, jonka käyttäjätunnus on "tester1" ja salasana "testpass1".*
- *Testitapauksen alussa testikoneelta ei kukaan ole kirjautunut järjestelmään sisälle.*

3.1.4 Automatisointi

- *Manuaalinen testitapaus*
- *Testitapaus voidaan automatisoida*

3.1.5 Testivaiheet

1. *Avaa selaimella palvelimen osoite <http://<server>/>*
2. *Kirjoita käyttäjänimi "username" kenttään*
3. *Kirjoita salasana "password" kenttään*
4. *Paina "Login" nappulaa*

Odotetut tulokset:

1. *Ohjelman kirjautumissivu aukeaa näyttöön*
2. *Käyttäjänimi kenttään voi kirjoittaa tekstiä*
3. *Salasana kenttää voi kirjoittaa tekstiä. Salasana ei näy käyttäjälle.*
4. *Login nappula toimii. Käyttäjä ohjataan ohjelman pääsivulle.*

3.2 Testitapaus 2

[Esimerkki SquashTM ohjelman ruudun kaappauksesta. Lisäksi ohjelmasta liitetään testitapaukset excel muodossa. Tiedosto muodostetaan export toimintoa käyttäen.]

The screenshot shows the Squash Test Management web application interface. The browser address bar indicates the URL: 10.21.1.160:8080/squash/test-case-workspace/test-case/3/steps. The application title is 'Squash Test Management'. The main content area displays the test case 'User login' with a status of 'Work in progress' and a priority of 'Very high'. The test case reference is 'Ohjelmistotestaus User login'. The test case is categorized as 'Ohjelmistotestaus' and 'User login'. The test case steps are listed under the heading 'Prerequisites and test steps'. The prerequisites are: Application server is running and can be accessed by using browser. There is a test user in the system: username: testuser1, password: testpass1. The test steps are: 1. Browse to http://<server address>/, Expected Result: Login page is loaded. 2. Type username 'testuser1' to the username field, Expected Result: Username field is functional and username can be typed there. 3. Type password 'testpass1' to the password field, Expected Result: Password field is functional and password can be typed in. Password is not visible in the form. Password is masked. 4. Click login button, Expected Result: Login button works. User is directed to the application front page.



export-the-test-cases_20230901_141317

3.3 Testitapaus 3

3.4 Testitapaus 4

3.5 Testitapaus 5

3.6 Testitapaus 6

3.7 Testitapaus 7

3.8 Testitapaus 8

3.9 Testitapaus 9

3.10 Testitapaus 10

4 ARVIO OMASTA OPPIMISESTA

[Arvioi tähän kappaleeseen omaa oppimistasi. Mitä osasit ja tiesit ennen harjoitusta? Mitä opit harjoituksen aikana?]

LÄHTEET