
Testiraportti

Projektin nimi

SISÄLLYSLUETTELO

1.	TESTAUKSEN TOTEUTUMINEN JA KATTAVUUS.....	3
1.1	TESTAUSSUUNNITELMA.....	3
1.2	YKSIKKÖTESTAUS.....	3
1.3	INTEGROINTI- JA JÄRJESTELMÄTESTAUS.....	3
2.	HYVÄKSYMISTESTAUS.....	4
3.	EI-TOIMINNALLISTEN OMINAISUUKSIEN TESTAUS.....	6
4.	TESTAUKSEN ARVIOINTI.....	7

1. TESTAUKSEN TOTEUTUMINEN JA KATTAVUUS

1.1 Testausstrategia

- *Kuka on kokonaisvastuussa testauksesta?*
- *Kuvaa ketkä osallistuvat testaukseen.*
- *Kuka testaa ja mitä testaa?*
- *Kuinka kattavasti ohjelma testattiin?*
- *Kuvaa testattavan ohjelmiston rakenne (komponentit/moduulit).*
 - *esim. kaavio komponenteista ja niiden välisistä riippuvuuksista.*
- *Kuvaa kuinka testaus liittyy Scrum-prosessiin*
 - *Mitä tehdään missäkin vaiheessa?*
- *Testauksen yleiset lopetus/hyväksymiskriteerit*

1.2 Yksikkötestaus

- *Kuinka testaus toteutettiin?*
- *Kuinka kattava?*
- *Kuka/ketkä toteuttivat?*
- *Lopetus/hyväksymiskriteerit*

1.3 Integrointi- ja järjestelmätestaus

- *Kuinka testaus toteutettiin?*
- *Kuinka kattava?*
- *Kuka/ketkä toteuttivat?*
- *Lopetus/hyväksymiskriteerit*

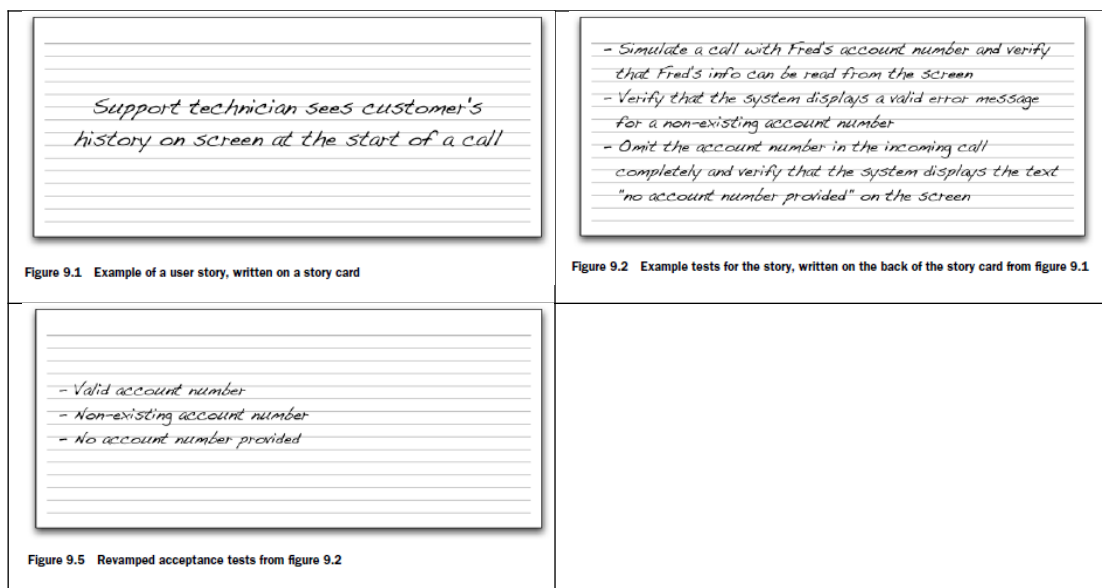
2. HYVÄKSYMISTESTAUS

Testien tarkoituksena on testata järjestelmän toiminnallisuutta ja sitä toimiiko järjestelmä oikein käyttäjän näkökulmasta. Kustakin käyttäjätarinasta/vaatimuksesta muodostetaan hyväksymistestit. Yhdestä vaatimuksesta tulee yleensä useita testejä.

Hyväksymistestit ovat:

- Asiakkaan omistamia
- Asiakkaan ja Scrum-tiimin yhdessä tekemiä
- Vastaavat kysymykseen mitä (ei miten, ei kuvata teknistä toteutusta)
- Kirjoitettu kohdealueen (asiakkaan/käyttäjän) kielellä
- Ytimekkäitä (suppeita), tarkkoja/täsmällisiä ja yksikäsitteisiä

Esim.



Kuvaa hyväksymistestaukset sprinttikohtaisesti, mitä testataan missä vaiheessa.

- Kuinka testaus toteutettiin?
- Kuinka kattava?
- Kuka/ketkä toteuttivat?

Kuvaa testitapauskohtaisesti testien suoritus. Voi olla erillinen testauspöytäkirja esim. Excel-taulukossa.

- Tunniste: ID/Epiikka/Vaatimus/Käyttäjätarina
- Testitapaus
- Prioriteetti (1,2,3)
- Testaus päivämäärät
- Sprintti milloin testattu (yksi tai useampi)
- Odotettu tulos
- Testauksen tulos: OK tai virheen vakavuus (A, B, C)
- Kommentti testauksen tuloksesta, jos ei mennyt läpi tai muuta lisätietoa
- Kuinka toimitaan, jos testaus ei mennyt läpi

Jos testitapaus ei mennyt läpi testauksesta jätetään alkuperäinen kirjaus. Tehdään uusi testikirjaus, kun testaus suoritettu uudelleen.

Prioriteetti

1. Korkea, sovellusta ei voi käyttää, jos testiä ei läpäistä
2. Normaali, testin läpäisy on sovelluksen oikean toiminnan kannalta välttämätöntä
3. Matala, testin läpäisy on suotavaa

Virheen vakavuus:

- A) Kriittinen – Estää järjestelmän toiminnan
- B) Haitallinen – Toiminto toimii väärin, mutta ei estä koko järjestelmän käyttöä
- C) Huomioitava – Ei toiminnallinen virhe, mutta korjattava (esim. visuaalinen)

Lisätietoa:

<http://index-of.es/Java/>

ReallyUsefulEbooks.net_1932394850_Manning.Test.Driven.TDD.and.Acceptance.TDD.for.Java.Developers..pdf

Test Driven, Practical TDD and Acceptance TDD for Java Developers. Lasse Koskela.

<https://docplayer.fi/2593757-Testauslahtoinen-ohjelmistokehitys.html>

Testauslähtöinen ohjelmistokehitys, Jussi Makkonen

https://www.theseus.fi/bitstream/handle/10024/56747/Friman_Ville.pdf?sequence=1

Testauslähtöinen ohjelmistokehitys, Ville Friman

3. EI-TOIMINNALLISTEN OMINAISUUKSIEN TESTAUS

Kuvaa ei-toiminnallisten ominaisuuksien testaus. Mitä testataan, kuinka, kuka vastuussa? Vaatimus – testin kuvaus

Kuvaa testitapauskohtaisesti, kuinka testit suoritettiin, mikä oli testin tulos (hyv/hyl), mahdolliset jatkotoimet.

4. TESTAUKSEN ARVIOINTI

- *Arvio testauksen luotettavuudesta ja riittävydestä?*
- *Mitä tuotteen laadusta voidaan sanoa testauksen jälkeen?*
- *Jääkö tuotteeseen tunnettuja virheitä? Kuinka niiden kanssa tulee toimia?*
- *Mitä opittiin, mitä tekisin toisin?*