

# Testauksen hallinta ja johtaminen

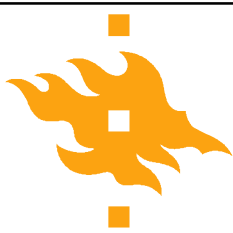
Luento 10  
Antti-Pekka Tuovinen



## Tavoitteet

---

- Yleiskuva seuraavista aiheista
  - Testauksen organisointi
  - Testaussuunnitelma
  - Testauksen kustannukset
  - Testausstrategia
  - Testausaktiviteettien hallinta ja ohjaus
- Testauspäällikön rooli ketterässä kehityksessä



# Testauksen organisointi



## Lähtökohdat

---

- Testausta tehdään projektin kaikissa kehitysvaiheissa (V-malli)
- Testausaktiviteetit pitää suunnitella ja koordinoita osaksi kehitystyötä
- Helpoin ratkaisu on vaatia, että kehittäjät myös testaavat ohjelmiston
  - Kehittäjät saattavat olla kuitenkin sokeita omille virheilleen
  - Testaajan ja kehittäjän *roolien erottaminen* tuottaakin yleensä paremman tuloksen



## Itsenäisen testauksen etuja

---

- Itsenäinen kehitystiimin ulkopuolelta tuleva testaaja (independent tester) havaitsee erilaisia virheitä kuin kehittäjät
- Itsenäinen testaaja voi verifioida kehittäjien (implisiittiset) olettamukset ja tulkinnat, joita he kehitystyön aikana tekevät vaatimuksista ja käyttäjän tarpeista



## Itsenäisen testauksen haittoja

---

- Kommunikaatiovaikeudet, jos kehittäjät ja testaajat ovat liian erillään
  - Alihankinta, off-shoring
- Testauksesta voi tulla *pullonkaula*, jos testaukseen ei ole osoitettu tarvittavia resursseja
- Jos kehittäjät olettavat testaajien löytävän kaikki ongelmat, kehittäjien vastuuntunto laadusta voi heiketä



## Organisointimalleja

---

1. Kehittäjät ovat vastuussa testauksesta, mutta he testaavat toistensa koodeja, eivät omiaan
2. Jokaisessa kehitystiimissä on mukana testaaj(i)a, joka hoitaa kaiken testauksen
3. Projektitiimissä on itsenäisiä testaustiimejä, jotka voidaan miehittää osittain kehitysorganisaation ulkopuolisilla henkilöillä (esim. käyttäjäyhteisön jäsenillä)



## Organisointimalleja

---

4. Tiettyihin testaustehtäviin käytetään erikoistuneita testausasiantuntijoita (esim. suorituskyky, käytettävyys, turvallisuus)
5. Erillinen organisaatioyksikkö (testausosasto, ulkopuolinen testilaboratorio, alihankkija) on vastuussa testauksesta (tai tietyistä testausosastoista)



## Organisaatiomallin valinta

- Riippumatta mallista, organisaation käytössä olisi hyvä olla myös (sisäisiä tai ulkoisia) *testauskonsultteja*, jotka voivat tukea kaikkia projekteja ja tarjota menetelmäapua testauksen eri tehtävissä
- Mallin valintaan vaikuttaa mm. testaustaso



## Komponentti- ja yksikkötestaus

- Testauksen pitää tapahtua elimellisesti osana kehitystyötä
- Yleisin mutta teholtaan heikoin vaihtoehto on antaa kehittäjien itse hoitaa testaus
  - Poikkeuksena ketterät projektit, joissa on vahva laatu- ja testauskulttuuri (TDD, XP)



## Komponentti- ja yksikkötestaus

---

- Organisaatiovaihtoehdot 1 ja 2 tuovat selvän parannuksen testauksen tuloksellisuuteen
- Testauspäällikkö (test manager) voi parantaa tuloksia vaatimalla tiettyjä toimintapoja, seuraamalla aktiivisesti toimintaa ja tarjoamalla valmennusta



## Integraatiotestaus

---

- Jos komponenttien kehittäjät ovat vastuussa myös integroinnista, integrointitestaus voidaan organisoida samoin kuin komponenttitestaus
- Integrointiin tarvitaan yleensä oma tiiminsä tai yhteistiimi, kun integroidaan monen tiimin tuottamia komponentteja
- Projektin koosta ja komponenttien määrästä riippuen malleja 3 ja 5 kannattaa harkita



## Järjestelmätestaus

---

- Järjestelmätestauksen tavoitteena on tuoda asiakkaan ja käyttäjän näkökulma testauksen lähtökohdaksi
- Itsenäinen ja kehitystiimeistä riippumaton testaus on siis ehdottoman tärkeää
- Mallit 3, 4 ja 5 ovat siis ainoat ammatillisesti hyväksyttävät mallit järjestelmätestauksen organisoinnille



## Testausroolit

---

- Ketkä sitten tekevät testausta?
- Mitä taitoja ja tietoja heiltä vaaditaan?



## Testauspäällikkö

---

- *Test manager, Test leader*
- Testauksen suunnittelun ja ohjauksen asiantuntija
- Tiedot ja taidot
  - Ohjelmistojen kehitys (software engineering)
  - Laatutyö (quality management)
  - Projektinhallinta
  - Henkilöstön hallinta ja kehittäminen



## Testauspäällikkö

---

- Tehtäviä
  - Projektin testausstrategian ja –suunnitelman laatiminen
  - Testausresurssien hankinta (rekrytointi, työkalut, jne.)
  - Testausaktiviteettien käynnistys ja valvonta kaikilla testaustasoilla
  - Metriikoiden valinta testauksen edistymisen ja (testauksen ja tuotteen) laadun valvontaa varten
  - Henkilöstöjohtaminen ja kompetenssien kehittäminen
  - Testaustoiminnan edustaminen organisaatiossa





## Testaussuunnittelija

---

- *Test designer, Test analyst*
- Testausmenetelmien ja testien määrittelyn asiantuntija
- Tiedot ja taidot
  - Ohjelmistojen testaus
  - Ohjelmistokehitys
  - Määrittelymenetelmät, formaali spesifiointi, mallinnus



## Testaussuunnittelija

---

- Tehtävät
  - Vaatimusten, spesifikaatioiden ja suunnitelmien/mallien analysointi, katselmointi ja arviointi testattavuuden ja testien suunnittelun kannalta
  - Testitapausten ja testien luonti
  - Testidatan hankinta ja tuottaminen



## Automaatioasiantuntija

---

- *Test automator*
- Testausmenetelmien ja testaustyökalujen asiantuntija
- Tiedot ja taidot
  - Ohjelmistojen testauksen perusteet
  - Ohjelmointi
  - Testaustyökalut ja skriptaus



## Automaatioasiantuntija

---

- Tehtävät
  - Automatisoi testien suorituksen käytettävissä olevia työkalujen ja skriptikielten avulla



## Testiadministraattori (järjestelmäasiantuntija)

---

- *Test administrator*
- Testien suoritusympäristöjen asiantuntija
- Tehtävät
  - Testausympäristöjen asennus ja hallinta



## Testaaja

---

- *Tester*
- Testien suorituksen ja virheiden (incident) raportoinnin ammattilainen
- Tiedot ja taidot
  - Testauksen perustaidot
  - Testityökalujen käyttö
  - Testin kohteen tuntemus



## Testaaja

---

- Tehtävät
  - Testisuunnitelmien ja testitapausten katselmointi
  - Testityökalujen käyttö (myös monitorit)
  - Testien suoritus ja kirjaus
  - Testin tulosten evaluointi
  - Virheiden ja poikkeamien dokumentointi ja raportointi



## Persoonallisia menestystekijöitä

---

- Menestyksekkäiltä testaajilta ja testauksen suunnittelijoilta vaaditaan paitsi teknistä osaamista, myös tiettyjä luonteenpiirteitä
  - Tiimityökykyä, diplomaattisuutta ja korrektiutta
  - Skeptisyyttä – kykyä kyseenalaistaa olettamuksia
  - Sitkeyttä ja määrätietoisuutta
  - Huolellisuutta ja luovuutta
  - Kykyä oppia nopeasti uutta




# Testauksen suunnittelu

HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

Faculty of Science  
Department of Computer Science

[www.cs.helsinki.fi](http://www.cs.helsinki.fi)

18 April 2013 25



# Testauksen suunnittelu

---

- Testaus on osa laadunvarmistusta (quality assurance)
  - Testauksella saadan analyttistä, konkreettista dataa ohjelmiston laadusta ja laadun kehittämisestä projektin aikana
- Testaussuunnitelma on osa laatusuunnitelmaa (QA plan)

HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

Faculty of Science  
Department of Computer Science

[www.cs.helsinki.fi](http://www.cs.helsinki.fi)

18 April 2013 26



## Testauksen suunnittelu

---

- Suunnitteluun sisältyy seuraavia toimia
  - Testaussuunnitelman tekeminen
  - Testien priorisointi
  - Testin aloitus- ja lopetusehtojen määrittely



## Testaussuunnitelma

---

- IEEE 829-1998 –standardi antaa mallin testisuunnitelman sisällölle
  - Kurssikirjassa liite A
- Testauksen yleissuunnittelu



## Testien priorisointi

---

- Kävi projektin aikataulun ja rahoituksen kanssa miten kävi, testit on priorisoitava niin, että tärkeimmät testit tulevat aina suoritetuksi
- Priorisointiperusteita
  - Toimintojen odotettu käyttöiähyys ja virheiden todennäköisyys käytön aikana
  - Virheriski = virheen vakavuus x todennäköisyys
  - Virheen näkyvyys käyttäjälle



## Testien priorisointi

---

- Priorisointiperusteita (jatkuu)
  - Laatuvaatimusten painotukset ja tärkeys asiakkaalle
  - Implementaation kannalta kriittiset komponentit ja toiminnot
  - Ohjelman osien kompleksisuus
  - Ohjelman osat, joihin liittyy suuri *projektiriski* (epäselvät vaatimukset, alihankkijan kyky toimittaa, kehitysorganisaation kyvyt, uusien teknologioiden käyttö jne.)



## Testien priorisointi

---

- Priorisointia on oltava valmis muuttamaan projektin kuluessa
  - Vioilla on tapana kasaantua
  - Paljon vikoja sisältäneiden testikohteiden testitapausten prioriteettia on syytä nostaa



## Testien aloitus- ja lopetusehdot

---

- Tiettyjen edellytysten on yleensä oltava voimassa, ennen kuin testausta voidaan aloittaa
  - Testausympäristö on olemassa ja valmiina käyttöön
  - Testaustyökalut ovat käyttövalmiina testausympäristössä
  - Testin kohden on asennettu ja valmiina
  - Testin tarvitsema data on saatavilla
- Ehdot pitää kirjata testisuunnitelmaan





## Testien aloitus- ja lopetusehdot

- Testauksen lopetusehtojen eksplisiittinen määrittely auttaa torjumaan seuraavia riskejä
  - Ilman selkeitä kriteereitä testauksen lopettamisesta päätetään epäjohdonmukaisin ja satunnaisin perustein
  - Vältetään liika testaaminen, joka johtuu siitä, ettei tiedetä, mitä testauksella tavoitellaan



## Yleisiä lopetuskriteereitä

- Testikattavuus
  - Kuinka monta testitapausta on suoritettu onnistuneesti, kuinka monta vaatimusta on testattu, mikä koodikattavuus on saavutettu
- Tuotteen laatu
  - Löydettyjen vikojen määrä, vikojen kriittisyys, vikojen löytymistähti, luotettavuus jne.
- Jäännösriski
  - Suorittamattomien testien määrä, korjaamattomat viat, epätäydellinen kattavuus
- Taloudelliset rajoitteet
  - Kulut, projektiriski, toimitusaikataulu, markkinatilanteen muutokset




# Kustannukset, Testausstrategia, Testausaktiviteetit

HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

Faculty of Science  
Department of Computer Science

[www.cs.helsinki.fi](http://www.cs.helsinki.fi)

18 April 2013 35



## Testauksen kustannukset

---

- Testaus voi viedä suuren osan ohjelmistoprojektin budjetista (jopa ~50%)
- Jotta testauksen käytetyt resurssit ja aika osataan asettaa oikealle tasolle, on ymmärrettävä mitä kustannuksia testauksessa löytymättä jääneet viat voivat aiheuttaa

HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

Faculty of Science  
Department of Computer Science

[www.cs.helsinki.fi](http://www.cs.helsinki.fi)

18 April 2013 36



## Vikojen aiheuttamat kustannukset

---

- Suorat kustannukset
  - Asiakkaan liiketoimintaprosessille välittömästi aiheutuvat kulut ja tappiot virhetoiminnoista (taloudelliset tappiot ja suoranaiset vahingot laitteille ja henkilöstölle); menetetty liikevaihto
  - Uusien, korjattujen ohjelmistoversioiden asennuksesta, käyttöönotosta ja uudelleen koulutuksesta aiheutuvat menot



## Vikojen aiheuttamat kustannukset

---

- Epäsuorat kustannukset
  - Ohjelmiston toimittajalle aiheutuvat suorat kulut (sopimussakot, hinnan alennukset, käyttäjätuen kysynnän kasvu)
  - Vahingot liiketoiminnalle; asiakkaiden ja maineen menetys, oikeudelliset kulut ja seuraamukset
- Vikojen korjaamisen kustannukset
  - Toimittajan tekemän lisätyön hinta (vikojen jäljitys ja korjaus, testaus, uusien versioiden jakelu ja asennus)
  - Resurssien (usein parhaiden) sitoutuminen korjaustyöhön uuden kehittämisen sijaan



## Vikojen aiheuttamat kustannukset

- Kustannusten arviointi voi olla haastavaa, mutta se on osa projektin riskianalyysiä
  - Asiakkaalle alusta lähtien uustuotantona tehtyyn ohjelmistoon liittyy enemmän riskejä kuin valmisohjelmistoja hyödyntävään ratkaisuun
- Yleisesti ottaen, *mitä aiemmin* vika löytyy kehitysprojektin aikana, *sitä halvempaa* sen korjaaminen on
  - Peukalosääntö: vian korjaamisen hinta kaksinkertaistuu testaustasolta toiselle siirryttäessä



## Testauksen kulut

- Moni tekijä vaikuttaa testauksen hintaan
  - Kehitysprosessin kypsyys (organisaation vakaus, tekemisen ennustettavuus)
  - Ohjelmiston laatu ja testattavuus
  - Testausinfrastruktuuri
  - Henkilöstön osaaminen
  - Laatuvaatimukset
  - Testausstrategia
- Testauspääallikkö ei juuri voi vaikuttaa kuin testausstrategiaan ja jossain määrin infrastruktuuriin ja henkilövalintoihin



## Testauksen työmäärän ja kustannusten arviointi

---

- Kustannuksia voi yleisesti arvioida kahdella tavalla
  - Jakamalla koko testaustyö erillisiin tehtäviin ja pyytämällä tehtävien vastuuhenkilöitä ja/tai asiantuntijoita arvioimaan työmäärää ja muita kustannuksia
  - Käyttämällä aiemmista projekteista kertynyttä dataa työmääristä ja kuluista arvioinnin pohjana
- Jälkimmäinen tapa tuottaa yleensä parempia arvioita, koska testauksen vaatima työmäärä yleensä aliarvioidaan!



## Testausstrategia

---

- Testausstrategian tavoitteena on
  - Optimoida testauksen kustannukset suhteessa löytymättömien virheiden aiheuttamiin kustannuksiin
  - Minimoida riskit
- Käytännössä laskelmien tekeminen voi olla vaikeaa puuttuvan kokemuseräisen datan takia
- Strategiapäätökset tehdään usein pikemminkin intuition kuin rationaalisen tiedon varassa



## Testausstrategia

- Ennaltaehkäisevä vs. reaktiivinen
  - Ennaltaehkäisevässä lähestymistavassa testaajat ovat mukana projektin alusta alkaen, jolloin testauksen suunnittelussa voidaan antaa riittävästi painoa aikaisessa vaiheissa tehtäville katselmoinneille ja analyseille
  - Reaktiivisessa toimintatavassa testauksen suunnittelu tulee mukaan vasta ohjelman toteutuksen ollessa jo pitkällä; tässä tapauksessa tutkiva testaus voi olla paras ja ainoa toimiva strategia
- Ennaltaehkäisevä toiminta on kuitenkin kustannustehokkainta koko ohjelmiston elinkaarta ajatellen



## Testausstrategia

- Analyttinen vs. heuristinen menetelmä
  - Analyttinen testauksen suunnittelu yrittää kvantifoida kustannukset ja kulut sekä niiden väliset riippuvuudet ja tuottaa optimoidun strategian laskelmien avulla
  - Heuristinen suunnittelu nojaa asiantuntijoiden kokemukseen ja peukalosääntöihin
- Käytännössä ollaan jossain näiden ääripäiden välillä ja mukana on muitakin elementtejä
  - Malliperustainen testaus, tilastolliset mallit, **riskianalyysit**, standardit, uudelleenkäyttö, tarkistuslistat, asiantuntijoiden perstuntuma



## Testausaktiviteetit, Virheraporttien käsittely

---

- Kts. Kurssikirja luvut 6.5 – 6.8.



## Testauspäällikön rooli ketterässä kehityksessä



## Ketterä kehitys ja testaus

---

- Ei kurssikirjassa
- Video: *Agile Test Management* by Fran O'Hara
  - <http://www.youtube.com/watch?v=4WngZa3efaU>



## Ketterä kehitys ja testaus

---

- Siirtymä ketterään reaktiivisessa moodissa – ole proaktiivinen
- Kaksi roolia
  - Linjaesimies – rekrytointi, sopivuus ketteriin kehitystiimeihin, valmennus
  - Testauksen johtaminen projekteissa
- Testaus tapahtuu projekteissa, ei erillisessä siilossa
- Testausstrategian muodostamisessa pitäisi pyrkiä ennaltaehkäisevään strategiaan
- Automatisointistrategia, automaation rooli (mitkä tasot)





## Ketterä kehitys ja testaus

---

- Älä ole liian määräävä testausprosessin määrittelyssä projektille; tarjoa neuvoja ja tukea ja ohjeita ja katso että tehdään järkeviä valintoja
- "Definition of done" määrittely avainasemassa – riskit, testaus
- Testaajat ovat mukana projekteissa alusta alkaen, mikä korostaa testauksen merkitystä ja testaajien roolia
- Laatuvaatimusten testauksen haasteet – miten saadaan mukaan näitten testaus heti alusta alkaen eikä vasta projektin lopussa
  - Profilointi ym.



## Ketterä kehitys ja testaus

---

- Esteiden poistaminen testauksen tieltä
- Testaajat voivat opettaa kehittäjiä laatimaan parempia testejä