

**T-76.5613 Ohjelmistojen testaus ja laadunvarmistus**  
*T-76.5613 Software Testing and Quality Assurance*  
Tentti / Examination 10.1.2006

Kirjoita jokaiseen vastauspaperiin seuraavat tiedot / Please give the following information on each sheet of paper

- kurssin nimi ja tunnus / name and code of the course
- tentin päivämäärä / date of the examination
- nimesi ja opiskelijanumerosi / your name and student id

VOIT VASTATA SUOMEKSI TAI ENGLANNIKSI / YOU MAY ANSWER IN FINNISH OR ENGLISH

1. Selitä lyhyesti seuraavat käsitteet / Explain briefly the following concepts: (6 p)
  - a) Ekvivalenssiluokkiin jakaminen / Equivalence class partitioning
  - b) Regressiotestaus / Regression testing
  - c) Testipeti / Test bed
  - d) Tutkiva testaus / Exploratory testing
  - e) Pareto diagrammi / Pareto diagram
  - f) Läpikäynti / Walkthrough
2. Alla on esitetty 5 testaukseen liittyvää väittämää. Vastaa jokaiseen väittämään pitävätkö väittämät paikkansa? Perustele vastauksesi lyhyesti tällä kurssilla oppimaasi hyödyntäen. / Below is 5 statements about software testing. For each statement, tell if the statement is true or false. Use the knowledge that you have gained on this course to explain briefly and give reasons for your arguments. (10p)
  1. Test-Driven Development (TDD) mahdollistaa testaajalle testien suunnittelun aikaisessa vaiheessa, jolloin kehittäjät voivat kirjoittaa toteutuksen testaajien aikaisemmin suunnittelema testejä vasten. / Test-Driven Development (TDD) enables early test design for testers. Developers can then write the implementation against the tests that the testers have written earlier.
  2. Käyttämällä kattavuustyökaluja ja suunnitteleamalla 100% haarakattavuuden toteuttavat testitapaukset, voimme mitä suurimmalla todennäköisyydellä sanoa, että testimme löytävät kaikki merkittävät viat ohjelmistosta. / Using coverage testing tools and designing our test cases to meet 100% branch coverage criteria, we can with high probability say that the tests will find all significant defects in the software.
  3. Mikäli testaajat löytävät ohjelmistosta vain vähän virheitä, ohjelmiston voidaan olettaa olevan korkealaatuinen. / If testers find only few defects in the software, we can conclude that the quality of the software is high.
  4. Todennäköisyys sille, että ohjelmistokomponentista löytyy uusia vikoja on suhteessa siihen kuinka paljon siitä komponentista on jo löytenyt vikoja. / The probability of the existence of additional defects in a software component is proportional to the number of defects already detected in that component.
  5. Ketterässä ohjelmistokehityksessä tyypillinen painotus kehittäjien konstruktiivisiin matalan tason laadunvarmistuskäytäntöihin auttaa arvioimaan objektiivisesti saavutettua laatutasoa ohjelmistokehitysprojektin aikana. / The emphasis on the developers' constructive low level quality assurance practices in agile methods, makes it easier to objectively evaluate the achieved quality level during a software development project.
3. Vikojen raportointi / Defect reporting (8 p)
  - a) Miksi on tärkeää kirjoittaa hyviä vikaraportteja? / Why is it important to write good defect reports? (2p)
  - b) Mitä tietoja vikaraporttiin tulisi kirjata? Kuvaille hyvän vikaraportin tärkeimpiä ominaisuuksia. / What information should be written in a defect report? Describe the important characteristics of a good defect report. (5p)
  - c) Mistä syistä (muita kuin ajan ja resurssien puute) ohjelmistovirheitä jätetään korjaamatta? / For what reasons (other than lack of time and resources) are software defects not always fixed? (1p)

4. Katselmukset / Reviews (8 p)
- a) Mitä ovat tyypilliset tavoitteet ja hyödyt, joihin katselmuksilla pyritään? / *What are the typical goals and benefits of reviews?* (2 p)
  - b) Kuvaile vähintään 3 erilaista katselmustyyppiä ja niiden keskeisimpiä eroja. / *Describe at least 3 different review types and their essential differences.* (3 p)
  - c) Mikä on tarkastuslistojen (tai tarkastussääntöjen) rooli katselmuksissa? Mitä hyötyä niistä on? / *What is the role of checklists (or inspection rules) in reviews? What benefits do the checklists provide?* (3 p)
5. Tehtävänäsi on testata erään www-selaimen "Etsi" toimintoa (Dialogi alla). / *Your task is to test the "find" functionality of a web browser (see the dialog below).* (8 p)
- a) Mitä mustalaaikkotestausmenetelmiä käyttäisit tämän toiminnallisuuden testaamiseksi? Perustele vastauksesi. / *What black-box testing techniques would you use to test this functionality? Give reasoning for your choices.* (2 p)
  - b) Suunnittele testitapaukset valitsemaasi/valitsemiasi tekniikoita käyttäen. Dokumentoi selkeästi testitapaukset sekä käyttämäsi oletukset ja perustele ratkaisusi. / *Design the test cases using the selected technique(s). Write down clearly the test cases and describe your assumptions and reasoning.* (6 p)

