Software Testing
Winter 2013 / 2014

List of Contents

1 Student Presentations

There will be 8 presentations to be presented by pairs / triples of students concerning special software test topics.

Duration of a presentation: About 30 to 60 minutes

Contents:

- Introduction, explanation
- Example, online demonstration
- Exercise(s)
- Discussion, summary, outlook

2 Topics

Here you find the different headlines for presentations, attached some hints as a proposal.

2.1 Requirements

Goal: Explanation: What are (good) requirements, Use Cases, User Stories?
Which sources could be used by a tester?

Topics

- Requirements Engineering
  General aspects, why they are so important?
- Functional / Non-functional requirements
- Examples of Use Cases, User Stories
- Examples for sources that testers could gain

Tasks for students

- Design of a Use Case
  - Design of a Test Case / some Test Cases out of the Use Case
- Design of a User Story

Sources

- http://en.wikipedia.org/wiki/Requirements_analysis
- http://en.wikipedia.org/wiki/Use_case
- http://en.wikipedia.org/wiki/User_story

2.2 Review

Goal: Presentation of software review techniques

Topics

- IEEE 1028 generic process for formal reviews
- Activities, roles and responsibilities of a typical formal review
- Types of reviews: Informal review, technical review, walk-through, and inspection
- Outlook and discussion
  - Reviews and testing
  - Advantages of reviews
  - Pair programming

Tasks for students

- Quiz
- Execution of a defined review process as example (e. g. of a written program).

Sources

- http://en.wikipedia.org/wiki/Software_review
- http://www.artima.com/weblogs/viewpost.jsp?thread=167363

2.3 Equivalence Partitioning and Boundary Value Analysis

Goal: Presentation how test cases are going to be defined
Topics

- Boundary Value Analysis
- Equivalence Partitioning
- Proceeding on example(s): Equivalence classes
- Related:
  - Decision Table Testing
  - State Transition Testing

Tasks for students

- Finding equivalence classes
- Evaluation, if given test values represent equivalence classes

Sources

- http://www.softwaretestinghelp.com/what-is-boundary-value-analysis-and-equivalence-partitioning/
- http://www.ruleworks.co.uk/testguide/BS7925-2.htm
- http://users.csc.calpoly.edu/~jdalbey/205/Resources/grocerystore.html
- http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf

2.4 Test Driven Development, Unit Tests

Goal: Explanation of test driven development (TDD) and unit tests
Topics

- Introduction:
  - What is the idea of TDD?
  - What are Unit Tests?
- Definitions
- Example showing how TDD works, how execution of Unit tests work
  E. g. Unit test cases in eclipse
  - JUnit
  - TestNG

Tasks for students

- Definition of Unit Test Cases for a specific problem
- Program Unit tests for a specific problem

Sources

- http://en.wikipedia.org/wiki/Test-driven_development
- http://www.javaworld.com/javaworld/jw-12-2004/jw-1206-tdd.html
- http://www.junit.org
- http://www.testng.org

2.5 Test management tools

Goal: Discussion and presentation of test management tools

Topics

- Why test management tools?
- Criteria to evaluate the “right testing tool”
- Alternative excel?

Tasks for students

- Task focussing on usage of a test management tool
  - Entering test cases
  - Documentation of executing a test
  - Defect management
  - Test report

Sources

- http://en.wikipedia.org/wiki/Test_management_tools
- http://www.testmanagementtools.net/
- http://www.allthingsquality.com/2010/04/things-i-like-to-have-in-my-test.html
- http://www.fitnesse.org
- http://www.testlink.org/

2.6 Code coverage, Cyclomatic complexity

Goal: Explanation of cyclomatic complexity and coverage

Topics

- Definitions:
  - Coverages
    - C0 Statement Coverage
    - C1 Branch Coverage
    - C2 Path Coverage
    - C3 Condition Coverage
  - Cyclomatic complexity (McCabe metrics): How to calculate? How to interpret? Explanation of edges, notes out of graph theory
- Advantages of usage
- Example(s) showing how to calculate

Tasks for students

- Calculation of defined coverages
- Calculation of cyclomatic complexity
- Quiz

Sources

- http://en.wikipedia.org/wiki/Code_coverage
- http://en.wikipedia.org/wiki/Cyclomatic_complexity
- http://users.csc.calpoly.edu/~jdalbey/206/Lectures/BasisPathTutorial/index.html
- http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf

2.7 Defect management

Proposal

Goal: Description how to handle a defect / incident, presentation of defect management tools

Topics

- Statements to a defect, attributes of a defect
  What information is important? Content of a defect / incident report.
- Demo of a defect management tool

Tasks for students

- Description of a defect
- Quiz

Sources

- http://en.wikipedia.org/wiki/Software_bug
- http://www.bugzilla.org
- www.mantisbt.org/

2.8 Test automation

Goal: Discussion of test automation and presentation of test automation tools

Topics

- What is test automation?
  (Capture Replay, regression testing)
- Why and when should we do test automation?
  Pros and Cons test automation
- Criteria to evaluate the “right test automation tool”
- Example: How do specific test automation tools work?
  - Canoo WebTest
  - Selenium

Tasks for students

- Execution of prepared test automation script

Sources

- http://en.wikipedia.org/wiki/Test_automation
- http://webtest.canoo.com/webtest/manual/WebTestHome.html
- http://seleniumhq.org/