Examination
Software Quality and Testing
DIT635

Software Engineering and Management
Chalmers | University of Gothenburg

Thursday June 10, 2019

Time	08:30-12:30
Location	Lindholmen
Responsible teacher	Daniel Strüber (mobile: +46 70 146 7691)
	Jan Schröder (mobile +46 72 974 6296)
Total number of pages	**4** (including this page)
Teacher visits exam hall:	At circa 09:30 and at circa 11:30
Exam (4.5 HEC)	Max score: 60 pts
Grade limits (4.5 HEC)	G: at least 30 pts
	VG: at least 48 pts

ALLOWED AID:

- English dictionary
- **NOT ALLOWED:** Anything else not explicitly mentioned above (including additional books, other notes, previous exams, or any form of electronic device: dictionaries, agendas, computers, mobile phones, etc.)

PLEASE OBSERVE THE FOLLOWING:

- Motivate your answers (a simple statement of facts not answering the question is considered to be invalid);
- Start each question on a new paper;
- Sort your answers in order (by question and sub-task) before handing them in;
- Write your student code on each page and put the number of the question on **every** paper;

Question 1 – Verification, Validation, and Defects (5pts)

Verification and Validation are essential parts of the software development process. Please explain these and related concepts in the following tasks.

  a) Explain the concepts of software verification and validation in your own words (**2pt**) and explain which of them is typically prioritized in the last phases of a project (**1pt**).
  b) Explain the difference between an error and a failure using one example (**2pt**).

Question 2 – Software Quality (10pts)

Software quality is defined as to which degree a system satisfies the stated and implied needs of its various stakeholders, and thus provides value.

  a) Explain the difference between internal quality, external quality, and quality-in use (**3pt**). Provide one example for a quality-in use and one for internal/external quality (**2pt**).
  b) Explain which of the software quality groups from a) are assessed during validation and which are assessed during verification. Why? (**2pt**)
  c) Name and explain three software quality attributes of your choice. (**3pt**)
     Note: The attributes should have either been presented in the course of be part of the references used in the course material (e.g. ISO/IEC 250xx). You cannot define your own quality attributes.

Question 3 – Software Measurement & Analysis (12pts)

Software measurement is often used to quantify software quality to make it more visible to stakeholders. Address the following tasks related to software measurement and analysis.

  a) You have conducted three measurements on a set of Java classes. First, you labeled each class either as containing mostly logical functionality or as containing mostly data-handling functionality. Second, you have measured the number of outgoing calls of each class (coupling measure). Third, you decided for each class if it is rather big, medium, or small in size. Assign each measure to an appropriate **measurement scale** and reason for your decision. (**3pt**)
  b) You are newly hired by a company developing automotive software. Your manager explains that they have a big problem with software complexity and so far they haven't done anything to investigate. He asks you to run **cyclomatic complexity** measures on all files and interpret the results. How would you reason to your manager that this ad-hoc idea is problematic (**2pt**) and how would you approach the complexity assessment in a more systematic way (which general steps would you take?) (**3pt**).
  c) Name and shortly explain two approaches used to **explore relationships** in measurement data and one approach used to **confirming a theory** using measurement data. (**2pt**)
  d) Explain how complexity and technical debt are related. (**2pt**)

Question 4 – Testing Basics (10pts)

In order to conduct and understand testing in practice, some basic concepts need to be under-stood. Solve the following tasks regarding general testing knowledge.

a) Explain the following concepts: Robust Equivalence-Class Testing (**1pt**) and All-Uses Coverage Criterion (**1pt**)

b) Assume a method "`public int calculcateEntryPrice(int age, int time)`" for computing the entry ticket price a public swimming pool. The method is supposed to return an entry price according to the following table:

time \ age	6-17	18-65	66-90
0 to 180 minutes	20 SEK	30 SEK	20 SEK
more than 180 minutes	30 SEK	40 SEK	30 SEK

i) Assume we want to do test the method by using equivalence class testing. What are the dimensions and what are the equivalence classes? (**1pt**)

ii) For each of the following coverage criteria, specify the minimum number of required test cases required in the test suite. Argue how you came up with your results. (**2pt**)
   1) Weak normal equivalence class testing
   2) Weak robust equivalence class testing
   3) Strong normal equivalence class testing
   4) Strong robust equivalence class testing

iii) For the given method, implement (on paper) a JUnit test suite that establishes weak normal equivalence class coverage. Implement the test suite as a class in which each method represents one test case. You may neglect import statements. (**3pt**)

iv) Based on the specification, draw a classification tree with a decision table for testing the program. (**2pt**)

Question 5 – Path-Testing (5pts)

Consider the following program:

```
1: read(x);
2: read(y);
3: if x < 10 then
4:     x := 10;
5: end if
6: while y < x then
7:     y := y + 1;
8: end while
9: x := x + 1
10: write(x);
```

Address the following tasks using the code above.

a) Draw the control-flow graph of the program (**2pt**)
b) Calculate all prime paths of the control-flow graph (**2pt**)
c) Compute the McCabe cyclomatic complexity for this program (**1pt**)

Question 6 – Other Testing (18pts)

Many more techniques are used by practitioners in the field on a daily basis, to address more specific testing needs. That includes A/B-Testing, mutation testing, and inspections.

a) Explain the **A/B testing** process (**1pt**). Which steps are typically involved? (**3pt**) Explain the additional value it provides when using A/B in addition to regular testing (**1pt**).
b) Shortly explain two use cases where you would use **A/B testing** and provide a reason why. Focus your short descriptions on the goal of the use case. (**2pt**)
c) Explain the basic idea of **mutation testing**. In your explanations present three concrete examples how operations are mutated. (**2pt**)
d) Explain the concept of **capture and recapture**, used in code inspections. What is the purpose, how is it conducted, and what is the outcome? (**4pt**)
e) Explain the test-driven development process. What are the typical steps involved? (**3pt**)
f) Explain at least one strength and one weakness of test-driven development. (**2pt**)