FINAL EXAM
Software Management

Code: 192340041
Date: 20 June 2011
Time: 8:45-12:15

General instructions

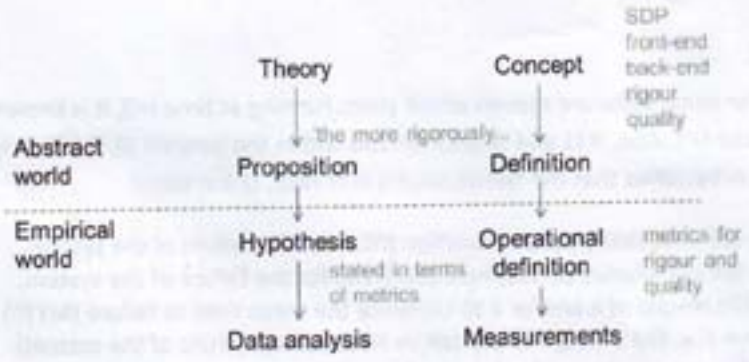
- This exam has 15 questions with a total of 100 points. The mark is determined by the sum of the marks obtained for answering each question.
- In this exam you are not allowed to use any study material (books, notes or copies of the sheets).
- Answers to the questions below should be concise ('to the point') but not cryptic. Be sure you write these answers so that the teachers can understand them.

Question 1

(4 points) Explain how product quality (small q) relates to process quality (big Q) in software development, and how they both relate to users' requirements.

Question 2

(10 points) In lecture 1 we separated the abstract world from the empirical world as in the figure below when discussing the role of measurements. Explain why this is necessary. Give an example to illustrate your explanation.

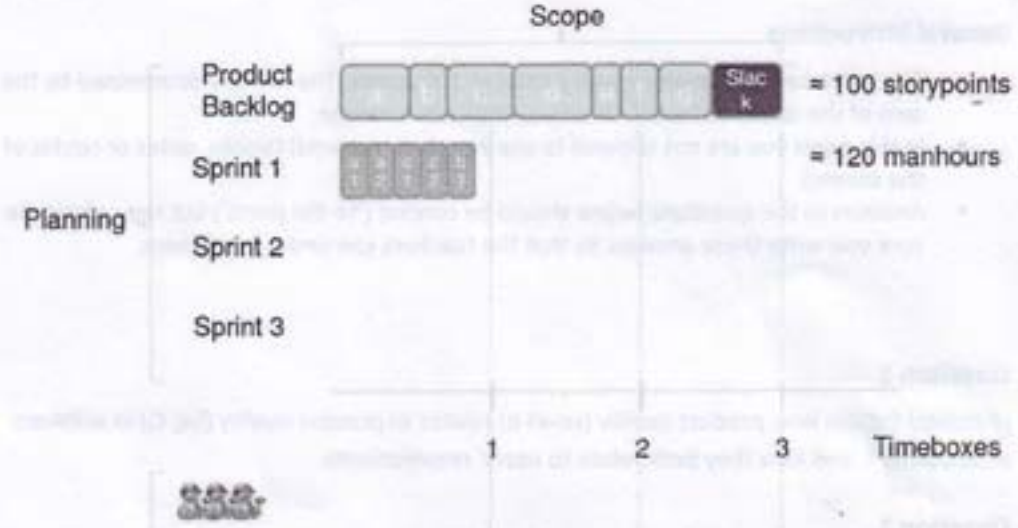


Question 3

(6 points) Why is Method content separated from Processes in SPEM (and in the Eclipse Process Framework)? How did you exploit this in assignment 1?

Question 4

(4 points) In lecture 3, the relationships among time, cost and scope in SCRUM have been discussed. What has been the purpose of fixing scope and time in sprints, as indicated below?



Question 5

(10 points) For some software system which starts running at time t=0, it is known that it will fail before time t=T. Also, if I1 and I2 are intervals within the interval [0,T] which have equal length, the probabilities that the failure occurs in I1 resp. I2 are equal.

a. Give the probability density function (PDF) for the failure of the system.
b. Give the cumulative density function (CDF) for the failure of the system.
c. Use the results of a and/or b to compute the mean time to failure (MTTF) for this system (i.e. the average of the failure times of many runs of the system).
d. Use the results of a and/or b to compute the median time to failure for this system (i.e. the time at which the failure is expected to have occurred in 50% of the runs of the system).

Question 6

(8 points) Give a graphical sketch of each of the following functions:

a. the probability density function of the exponential model
b. the probability density function of the Rayleigh model
c. the cumulative density function of the exponential model
d. the cumulative density function of the Rayleigh model

Note: no justifications, no formulas, no proofs, no explanations required, just sketches.

Question 7

(4 points)

a. Formulate the Jelinski-Moranda Model for software reliability growth.
b. Which are the (not necessarily realistic) assumptions which underlie this model?

Question 8

(8 points) Explain the purpose of the metric Lines of Code (LOC). What are the approaches for calculating LOC and how is LOC affected by the presence of several product releases?

Question 9

(6 points)

a. Explain which value of the BMI metric is an indication for a good management of the fix process.
b. Give the formula used for calculating the BMI metric.

Question 10

(6 points) Explain the 80-20 principle for identification of quality problems. Which quality tool is suitable for showing the relation between the defect causes and defect count?

Question 11

(4 points) The complexity metrics aim at predicting the defect rate: the higher the complexity is the higher the (expected) defect rate is. Considering this definition, answer the following questions:

a. The direct measure of program size (for example in terms of lines of code) is no complexity metric. Explain why it is or is not the case.
b. How can a complexity metric be derived from it?

Question 12

(10 points) Consider the following piece of code and answer the questions below.

```
if (k < 2)
{
  if (k > 3)
    x = x*k;
}
```

a. What are the values of the primitive measures from Halstead's Software Science and what is the (predicted) number of faults in the program? Do not just write the numbers but also explain your solution path, for example naming the operands. (Note: For calculating the log2 function you can round up; then log2(x) is the number of bits you need when writing x as binary number.)

b. Explain two of the disadvantages of Halstead's Software Science.

Question 13

(6 points) Select two metrics from the Chidamber Kemerer (CK) Metrics suite, and for each of them give the following information:

a. Name.
b. Intuition behind.
c. Calculation formula.
d. Possible explanation for a large value of this metric.

Question 14

(6 points)

a. What is the cyclomatic complexity of a module which contains three statements of the form "it-then-else", two statements of the form "if-then-elseif-else" and one "do-while" loop whose body contains a statement of the form "if-then"? Explain your solution path.
b. The cyclomatic complexity is an indicator for the number of defects in a module. Nevertheless, it frequently happens that modules have the same or similar cyclomatic complexities but they appear differently difficult to understand. What is a limitation of the cyclomatic complexity (explain it) and how can this be overcome?

Question 15

(8 points)

a. Characterize internal software measures and give an example.
b. What can you say about modules with a large fan-out?
c. What do you need to consider when choosing a complexity metric for your project?
d. Metrics are a way to determine the quality of existing code and give hints how to improve the quality. Name one method that proactively aims at producing code in a good quality. Explain the rationale and the approach of this method and give an example.

Good luck!