
Softwaretechnik/Software Engineering

<http://swt.informatik.uni-freiburg.de/teaching/SS2015/swtv1>

Exercise Sheet 1

Early submission: Wednesday, 2015-04-22, 18:00 Regular submission: Thursday, 2015-04-23, 12:00

Regarding the form of submission, we have the following preferences:

- *perfect*: a PDF submitted via ILIAS
- *kind*: a scanned version of the handwritten proposal submitted via ILIAS — there is a magic print-copy-scan-machine in the pool room which can send the scan to you by mail
- *tolerated*: paper submission

Exercise 1 – (Aspects of Software)

(15/20 Points)

The purpose of this exercise is to get a feeling for different aspects of software which need to be addressed by software engineering. From your experience, knowledge, or the (scientific or popular) literature, name examples of software that illustrate the aspects below. For each example you give, discuss briefly what aspects and to what extent it illustrates them, and indicate your source.

All tasks below also ask for *your* classification of the project of your *Softwarepraktikum*. If you did not or are not currently participating in *Softwarepraktikum*, consider the *Softwarepraktikum* homepage or interview your classmates that are currently or have been enrolled.

- (i) **Size**: what is the smallest/biggest software you know which is (or which you think should be) subject to software engineering as introduced in the lecture? (4)

Hint: Discuss your measure of size (e.g. lines of code, number of modules, person months, length of task description, number of comment lines, duration of testing, etc). Is it a good measure?

- (ii) **Criticality**¹: One can distinguish the following criticality aspects of software.

- a) Safety critical: A software error can lead to loss of life, injuries of people, or to a heavy damage of the environment.
- b) Mission critical: Software whose function plays a central role in its system, that is, an error in the software may cause its system not to fulfil its mission.
- c) Business critical: A software error can lead to a substantial economic loss.
- d) Non-critical: none of the above.

Discuss one example of software for each criticality aspect and in particular their role in their surrounding systems. What is the criticality of the *Softwarepraktikum* project? (3)

- (iii) **The relationship between customers, developers and users**: Formally, one can define the type of a software development project by the relation between the customer C (who orders/pays for the development), the developer D (who actually writes the software), and the user(s) U . They may be organizations or single persons.

Illustrate each of the following cases with an example:

¹See also http://en.wikipedia.org/wiki/Hazard_analysis

- a) $C = D = U$: The customer, the developer and the user are the same.
- b) $C = D, C \neq U$: The customer and developer are the same. The software is developed for external users.
- c) $C = U, C \neq D$: The customer is the user of the software. The development of the software is performed by a contractor.
- d) $C \neq D, D \neq U$: The customer contracts the development of the software externally with the intent of offering it to external users.

Which is the particular situation in the *Softwarepraktikum*? (4)

(iv) **Maintenance**: is every work on an existent software system, that was not planned during development (or could not be planned) and has a direct effect on the users of the software. We distinguish four types:

- a) Adaptive maintenance: the software is changed to satisfy new or changed requirements.
- b) Corrective maintenance: The software is changed to prevent the occurrence of an observed error.
- c) Perfective maintenance: Changes performed to improve certain quality characteristics of the software.
- d) Preventive maintenance: Changes performed for the purpose of preventing problems before they occur, e.g. Y2K.²

Explain each of the types of maintenance work using (an) example(s). For your particular example(s), briefly outline the maintenance tasks and the duration after product completion during which maintenance work

What kinds of maintenance do you expect to perform on the software resulting from the *Softwarepraktikum*, and when do you plan or expect to discontinue maintenance? (4)

²See also <http://en.wikipedia.org/wiki/Y2k>

Exercise 2 – (Software, Formally)

(5/20 Points)

Consider the following Java program S :

```
1 import java.io.RandomAccessFile ;
2
3 public class Program {
4     public static void main(String [] args) throws Throwable {
5         long x = 0;
6         RandomAccessFile f = new RandomAccessFile("test", "r");
7         x = f.length();
8         x = 1 + (x / 1024); // round length from bytes (upwards) to kByte
9     }
10 }
```

- (i) Give the set of *computation paths* $\llbracket S \rrbracket = \{\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \dots, \dots\}$. Assume that system states are composed of the program counter pc and the value of variable x , that is, $\sigma_i = \langle pc, x \rangle$ for all i .

Hint: The prefix of all computation paths relevant for this task is $\sigma_0 = \langle pc = 5, x = \perp \rangle \xrightarrow{\tau} \sigma_1 = \langle pc = 6, x = 0 \rangle \xrightarrow{\tau} \dots$ where \perp denotes an undefined value and the value of the program counter pc is the line number of the statement about to be executed. (3)

- (ii) In the lecture it is mentioned that σ may include any sort of information about the state of the system. In our example above, information such as the state of the execution environment of the program (e.g. the state of the file system, environment variables, date and time) or the state of the virtual machine in which the program is executed (e.g. program arguments, free memory, exceptions) is disregarded.

Give at least one additional interpretation of $\llbracket \cdot \rrbracket_2$ that includes additional information. I.e., give the new resulting set of computation paths $\llbracket S \rrbracket_2$. (2)