Solutions to Exercise Sheet 0

Exercise 1 – Are Software Engineers Managers?

- The software development is almost always done in a team. Effective teamwork can bring great results, but it needs to be well organized. The communication between the members of a group, a proper distribution of work, setting priorities or time scheduling are responsibilities of a project manager.

  Why is the work distribution important? A proper work distribution can increase the productivity and improve the quality of out coming product. There are few common mistakes that can be done:

  - not knowing or not paying attention to the abilities and key skills of team members. This could lead to time waste and frustration among the developers.
  - On the other hand not providing the less experienced employees new challenges. A good manager should give the team members a possibility to develop themselves, otherwise people can lose the interest in work.
  - To give too much work to the most productive, most experienced or most responsible member of a team. An overworked employee would not do his best, would feel stressed and used, and it would spread the rest of the team. As we see the work distribution can be a challenging task that requires not only management or executive skills, but also understanding of the capabilities and psychology of team members.

  This problem is not specific to software development, however it is applicable and very important for people working in IT.

  Why 'normal' managers couldn't do this: Experience in managing would be beneficial to this, but in software development it is especially critical to have at least some knowledge of the tasks one distributes to others. For example it can be harmful to the development process if managers don't understand the real value of testing or underestimate the cost of implementing a feature.

- Managing the interaction between different Software components (eg. libraries etc.) Defining the project boundaries can be a challenging task. For software development it is essential to determine the standards of the work, APIs, code quality, architecture and to manage resources(e.g. time, money, RAM, CPU-Power). This has to be done by a project manager. However this management activity requires knowledges and experience in computer science. Technical limitations has to be considered. Unreasonable time demands for example or money limitations can lead to degradation of the quality or in the worst case to not accomplishing the task, which is waste of time and money.

- Good software reflects to domain it tries to solve. Most software is developed for other industries than our own. So developer have to understand the domain they are working in. We can't build a banking software without understanding how accounting works. To achieve this communication with domain experts is essential. Although translation between different teams/ departments is a typical management activity in software development it requires a requires a thorough understanding of our industry.

Exercise 2 – Does Software Reliability Matter?

The heartbleed vulnerability

The heartbleed bug was a vulnerability introduced to the Codebase of OpenSSl in December 2011. Since March 2012, affected Versions of OpenSSL allowed atackers to extract information from the RAM of connected servers which was often used before to store private keys and other important information.

The vulnerability was caused by the implementation of a the 'heartbeat', which is usually used to determine wether a TSL-encrypted connection to a server is still usable and the other partner is responding. This should have been done by sending a message and getting the same content of it back, but the message could be altered in a way which would lead to the system to send up to 64 KiB unused memory following the buffer which was used to store the received information. Because this memory was likely to be used by OpenSSL before, it often contained sensible information.

The exact damage which was caused is not exactly known, because the bug was not noticed for nearly two years, but there were some incidents where Social , and one where researchers could extract their own private keys from a machine.
The bug could have been notecied way earlier:
If the implementation which was adopted into the codebase would have been checked better, the vulnerability would not have existed.

Provide a general description of the case, followed by a more detailed description of the software-related issue and its consequences. Quantify the damages caused and argue why the case is relevant. Discuss in how far the incident (following official reports, or in your opinion) is related to issues with requirements, design, quality assurance, management, or usage under specified conditions.
Sources: http://heartbleed.com/ https://en.wikipedia.org/wiki/Heartbleed

Survey

1. Expectations

Our expectations for the Softwaretechnik/Software Engineering course are:

- Understanding the procedure of software production, including common mishaps at each step
- Learning more about the planning process in general as well as specifically applied to Software Development
- to gain knowledge in requirement analysis, process scheduling (..), organization, resource distribution, design and testing
- to learn how to use basic and maybe some advanced techniques, models and patterns in software development

- learn the modern techniques of software engineering (agil instead of waterfall), e.g. D(omain)D(riven)D(esign), T(est)D(riven)D(esign), B(ehaviour)D(riven)D(esign)
- getting to know valuable Design-Patterns and being able to apply them more generally
- to see a focus on software architecture (more than just some design patterns, e.g. scalability, ...) in general, specifically how to design and maybe use them
- to get smarter, have some fun, and learn a lot along the way, not only for the further studying or working but also for life
- think about Software in a more General way, as in having a deeper knowledge about it's usual background
- getting tools (roughly specific ideas) for attacking Problems with the help of Software, or simply training that

We think that this will help us with our career because management experience (even if only in theory) can help in any job additionally to it being useful for Life on a more general scale. As upcoming Software Engineers it's more than useful having at least some basic knowledge about not only how to code, or rather, in the actual field of Software Engineering. Additionally, hobby projects or help on Open-Source Projects in spare time could be done more efficiently, knowing some of the involved processes, usually used structures for more efficiancy and more. Getting better at problem solving is another one, working as a team and having fun just makes studying (and later on working) a whole lot easier.

2. Previous Experience

	0	1	2	3	4	5	6	7	8	9	10
Project Management (cf. Exercise 1)											
Nils Hagner	x										
Michael Fleig			x								
Anush Davtyan			x								
Felix Karg							x				
Requirements Engineering (capturing and managing requirements from users or clients)											
Nils Hagner	x										
Michael Fleig						x					
Anush Davtyan					x						
Felix Karg					x						
Programming (writing code, fixing bugs)											
Nils Hagner		x									
Michael Fleig										x	
Anush Davtyan					x						
Felix Karg										x	
Design Modelling (creating an architecture or behavior model of a solution)											
Nils Hagner	x										
Michael Fleig											x
Anush Davtyan				x							
Felix Karg								x			
Software Quality Assurance (e.g., testing, code review, formal verification)											
Nils Hagner	x										
Michael Fleig									x		
Anush Davtyan				x							
Felix Karg									x		

3. Regarding the Softwarepraktikum...

	Nils Hagner	Michael Fleig	Anush Davtyan	Felix Karg
I will be participating in it this semester.				
I have already taken part.				
I will participate in it in the following semesters.	x	x	x	x
It is not part of my study plan.				

4. Language

□ German.
□ I prefer German, but English is okay.
☒ I prefer English, but German is okay.

□ English.