

Info 3

Laboratory 2

Prof. Weber-Wulff

23.04.2020



Lab 2: Software Processes

Mayya Bareeva: s0566169

Lars Vorwerg: s0571683

Pavel Tsvyatkov: s0559632

Livia Konitzer: s0564775

1. **Suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems. Explain your answer according to the type of system being developed.**

a. A virtual reality system to support car maintenance:

Since this is going to be a large-scale project, we thought that the **waterfall model** is best suited here. There are going to be different development teams responsible for the virtual reality system and the car maintenance one. This means the requirements need to be very well understood and the project needs to be planned carefully, because implementing changes at a later stage will come at a very high cost. We also thought about it in a way that there will be specific hardware required (the Virtual Reality set/glasses), so the whole thing needs to be well communicated and structured.

b. A university course management system that replaces a current one:

Reuse oriented development is a more suitable model for that as we already have some parts that are implemented and we just have to reuse the components that we have and put it into a new system. The one of the points of such development is that requirements should be modified if they do not match or do contradict the existing components (in our case new requirements have to reflect the course management system; if the team faces the issues, then other solutions should be found e.g. requirements changed/adapted).

c. An app for determining potential contact with people infected with a highly infectious disease:

Incremental development is a better model because it is mostly used for apps or software products and not by large systems. Also, the requirements are defined by the user's feedback and changes to the system can be done easier and regularly in short periods of time. Since changes are easier to implement, this reduces the costs greatly.

d. An interactive travel planning system that helps users plan journeys with the lowest environmental impact:

Evolutionary development would be the most suitable model to manage an interactive system. Traveling is a service that is always changing and the only way to get new data and information is by users that used a traveling system not known so far. The system would start out with the best known way at the

time, while also being well aware of the system requirements, and would evolve into a better one in time. That would require working closely with customers, taking their feedback and adding or changing features as suggested.

- 2. You have developed a prototype of a software system and your manager is very impressed by it. She proposes that it should be put into use as a production system, with new features added as required. This avoids the expense of system development and makes the system immediately useful. Write a short report for your manager explaining why prototype systems should not normally be used as production systems.**

In general a prototype is just like a “demo” version of a software system. Developing a prototype can help to reduce costs by finding out about potential changes, which will be needed in a system, early on, as it is much more costly to implement changes later on, when something has already been developed. It also helps find errors that were not anticipated in the system requirements or change them accordingly.

However, it’s possible that some of the functionality is left out in order to speed up the process of developing the prototype system, which also means that the quality has been reduced and this is something we definitely don’t want. The prototype also hasn’t been tested in a real environment which means there can be a lot of unidentified bugs, performance problems and big security flaws. Beside the technical difficulties, the software isn’t adapted to the needs of the actual users. Even if the prototype seems to be well suitable for the engineer, it doesn't mean it fits for the people who are actually going to use the software, because they don’t have the same expertise as software engineers. It’s important to consider the issues the users might have, because if they don’t like it, they won’t use it or their performance level will drop.

Therefore, while it is a good way to demonstrate how something should work or test a different design, **it shouldn’t be used in production.**

3. You are working for a software house and developing a system using Scrum. How would you be able to use Scrum in a forced home-office situation like the current Covid-19 crisis? In order to reduce costs and the environmental impact of commuting, senior management proposes continuing with this set-up. What would be the advantages and disadvantages of such an arrangement?

The majority of meetings will be just moved to zoom or slack. Everyone can speak out at *stand up*.

Retrospective: the notes of team members can be created also online on the board.

Ticket estimation meetings can be done via screen sharing (to go through the backlog of ideas) and estimation of the ticket can be shared in a chat.

Advantages:

There is much **more flexibility**. Employees can define the hours when they are available online. They can let the other team members know exactly when and how long he/she is away.

No commuting, which allows the team members not being late for stand-ups and every team member will share one's progress or impediments. Employees have **more time and less stress**, they can get more sleep and don't have to deal with late trains for example.

They also **save money** because they don't have to buy train tickets or tank up their car.

Additionally there is **less CO2 emissions** and **less maintenance** for vehicles.

Disadvantages:

Working from home comes along with **more distractions**. You have to set up a workplace to calmly do your job but that's not possible for everyone.

It's also possible to get a **wage reduction** because the performance rate can go down.

Technical problems can occur like scrum meetings getting interrupted because of a bad connection or slow upload and download rate which **affects the progress**.

There is **less or even weak communication** because not every team member responds straight away, maybe they turned off notifications or take longer lunch breaks.

Not getting in touch with your team members can cause **losing track** of all the work that should be done. It's more likely to overlook a task or miss a deadline and that causes more stress for everyone involved.

- 4. Historically, the introduction of technology has caused profound changes in the labor market and, temporarily at least, displaced people from jobs. Discuss whether the introduction of extensive process automation is likely to have the same consequences for software engineers. If you don't think it will, explain why not. If you think that it will reduce job opportunities, is it ethical for the engineers affected to passively or actively resist the introduction of this technology?**

Nearest future:

Software developers are not going to be affected because they are the only ones who implement automation of the processes. Because of current digitalisation more and more professionals will be hired and needed.

However, for some engineers it would have a detrimental effect because some companies will be willing to find the engineers that work remotely from the countries with cheaper wages. This can affect the salaries that could be reduced because of this slight tendency.

On the other hand, other jobs will be not just affected but dying because of processes automatisation e.g. social services: the major part of assistance will be substituted by the apps/programs or robots.

Further future:

It most certainly will happen. One example would be "Emily Howell" a bot that creates music that can not be differentiated between bot and human. So why would that not apply to software at one point. Job loss was a big thing already starting with Industrialization and we have kept automating ever since. Evolution is in human genes and we won't stop at some point which is ethical as long as people affected by it are treated in such a way, that they don't lose their purpose in society.

Stopping the progress just to keep people in their jobs isn't a future oriented way, but in our society replacing people by machines e.g. can be life-threatening for the ones losing their job. And if it's going on this way without changing the system the digitalization will end in unemployment and poverty for the demographic groups who don't own these productive forces. That's caused by the compulsion to work, but why should people do jobs (that they maybe hate) that a software or machine can do better, faster and with less cost? This could be solved with an unconditional basic income for example. But that is not our job to do.

Reflection

Lars: *When we started discussing the advantages and disadvantages more advantages came to my mind than disadvantages. Most discussions, even though it was the smallest amount of time spent, happened talking about the future and what will happen to society as a result. We had multiple back and forth arguments that were interesting to hear and talk about which was the point of the lecture. For the further project meetings this was a good introduction on how we solve arguments and can cooperate.*

Mayya: *For me, personally, it was not easy to find the disadvantages of working from home and it took time to come up with something. Because, according to my little experience working in a team, we manage the scrum meetings pretty well, everyone in a team has a chance to speak out etc.. As for the exercise with models, we had difficulties because we had to revise the material one more time and analyse each issue separately, so that later on we could come to a common conclusion.*

Pavel: *We were able to have a nice discussion about all the questions, we shared different ideas and we all came to some sort of an agreement in the end. I think this exercise was very interesting, especially the conversation and thoughts we had about the Scrum home-office situation. Even if I'm not used to talking about software process models yet, while discussing the first exercise in and out of class, hearing others share their opinion and thinking about it myself, helped me understand more about the different models.*

Livia: *At the beginning we had some difficulties with task 1 because we all had to dive back into your slides to get a better overview about the different process models, that's why we decided to do this for ourselves separated from the laboratory. But afterwards we had some really productive discussions about the other questions, we collected our thoughts in a shared google doc and just had to form them into sentences after the exercise. For me it's still difficult to think about choosing a suiting process model because there isn't a perfect solution for each project, you always have to consider the requirements and circumstances.*

Time Spent On Exercises

	In class / Out of class
Exercise 1	30 min / 45 min
Exercise 2	20 min / 15 min
Exercise 3	15 min / 20 min
Exercise 4	10 min / 15 min