IN5140

Smart processes and agile methods in Software Engineering

IN5140 - Exam 2016

Exercise 2 (20%)

- What is a retrospective meeting?
- What is process conformance?
- Why is process conformance important in process improvement?
- What is the Plan-Do-Check-Act cycle?
- Describe strengths and shortcomings of a survey in research?
- Describe strengths and shortcomings of an experiment in research?
- What is a case study?
- Describe advantages and disadvantages regarding likert type scales in a survey?
- What is continuous integration in large scale agile?
- What are the main success criteria in software development?

What is a retrospective meeting?

- Retrospective meeting is intended to reveal problems and issues in a finished sprint. Team considers what and how could be potentially improved in near future.
- About the process, not the product.

What is process conformance?

- Process conformance is all about following a predefined formal process model as much as possible.
- How much does the current process deviate from the formal process it originally followed (e.g. Scrum → Scrum-ish)

Why is process conformance important in process improvement?

If a team or company is not trying to conform to a new process model, it makes attempt to make any changes meaningless. Evaluation of new process model also becomes almost impossible, because it was not followed correctly. As a result, following improvements will be hard/impossible to introduce due to lack of robust information about the state of current process.

What is the Plan-Do-Check-Act cycle?

PDCA stands for:

- **Plan** - prioritizing what, how and when should happen,
- **Do** - best effort to fulfill outline plan,
- **Check** - evaluate and analyse results of what has been done,
- **Act** - act accordingly based on observations achieved in "check" stage

Describe strengths and shortcomings of a survey in research?

Strengths:

- Survey is a powerful tool which can provide insight on phenomenon(s), but lacks details and does not allow to drill into observed data
- Can reach a lot of people

Shortcomings:

- Does not reveal meanings, and the important details behind each answer
- Participants might rush through

Describe strengths and shortcomings of an experiment in research?

- Experiments enable researchers to observe phenomenon in isolated, controlled environments.
- Experiments can eliminate variables which are not relevant and can set a focus on the subject of the research.
- On the other hand, experiments might not reflect reality due to their artificial nature (e.g. if it is representative outside of the controlled environment).

What is a case study?

- Case studies are often used in qualitative research
- Case study is a research method which goes deeply into **one** phenomenon in the context of the real world

Describe advantages and disadvantages regarding likert type scales in a survey?

- Likert type scales are very useful when it comes to collection of subjective data. Likert scales often provide quantitative data.
- Hard to compare two items belonging to the same likert scale objectively
    - For instance: Likert scale of happiness with items like very happy and happy. How could one objectively compare them? As the result, it becomes almost impossible to figure out interval distance between two items in likert scales.
- Another disadvantage of data collected through survey based on Likert scale is that it is not possible to drill down on it in order to get more insight / details.

What is continuous integration in large scale agile?

Continuous Integration (CI) is a server which continuously observes changes in the source code and runs automated tests against them.

If a piece of code which breaks the build, CI will send a notification to the team members that something went wrong and needs to be fixed. In large teams it is extremely useful, because it can also automate certain routines like deployment or generation of reports like test coverage.

What are the main success criteria in software development?

It depends, but generally the most important criteria are:

- Staying on time
- Staying on budget
- Delivering quality software which fulfills customer requirements:
  - Robust
  - Efficient
  - Secure
  - Maintainable
  - Size (smaller the codebase is better)

Exercise 3 (20%)

Give a description of the following overall lean principles:

- Satisfying the customer
- Flow
- Visualization
- Avoiding waste
- Supporting change

Satisfying the customer

Focus primarily on requirements and needs of the customer (product owner). Instead of investing too much time into planning, documentation and design of the product, the company has to focus on delivery of the Minimum Viable Product (MVP) in short period of time. Based on feedback from product owner, team can adjust requirements and tailor development process to what customer actually needs instead of potential, not realistic use cases.

Flow

Flow is all about delivering maximum value to the customer.
Instead of focusing only on certain, isolated tasks, it tries to prioritize (maximize) overall progress of system development and to avoid waste.

In Lean, the goal is to deliver working software as soon as possible instead of focusing on boundary tasks like too detailed planning. In order to maximize flow, obstacles in development process should be identified and solved as soon as possible. In order to achieve this, communication between teams should be transparent and honest.

Visualization

Tracking development progress is essential for any project both in Lean and Agile frameworks. The primary idea standing behind it is to provide visual clue of current progress status for everyone involved in the project. There are certain tools and techniques that exist to achieve this, for example:

- Burndown charts - visualizes amount of work which is left to do before estimated delivery time
- Kanban (board) - an agile software development framework which emphasizes visualization of work in progress

Avoiding waste

In the context of Lean, waste can be anything what does not benefit customer. Less waste means better, more productive flow. For example, excessive efforts invested in premature design and documentation are considered to be a waste. Some other possible reasons of waste:

- Overengineering - investing too much time into architecture and design of software product in the beginning is wasteful
- Wishful thinking - focusing on features and functionality which potentially can be useful in future for customer.
- Too much planning and documentation for a product which does not exist yet is also considered to be waste.

Supporting change

To support change, not only developers, but the whole company should perform in an agile way. Managers and project coordinators should always be open for communication and helpful for the customers. Instead of being passive, the company should facilitate and help customer to build better product by sharing experience and providing valuable insight on development process. On the level of engineering, these actions could be applied:

- Short release cycles, facilitate faster feedback loop
- Writing unit and integration tests allows developers to make sure that nothing is broken after implementation of new feature (or changes made to existing one)
- Continuous Integration (CI) enables company with a small team of developers to scale in size over time

Exercise 4 (20%)

Give answers to the following questions:

1. Explain the Goal-Question-Metric (GQM) model.
2. What are internal and external attributes of software?
3. Give one example of an internal attributes and one example of an external attribute of that you have used, or could have used, in your compulsory project in IN5140.

Explain the Goal-Question-Metric (GQM) model

GQM is an evaluation / a measurement technique used to evaluate software changes. The model has 3 steps:

- **Goal**: In order to measure anything, we first need to introduce motivation for proposed changes. The Goal should reflect ideas standing behind proposed changes and expected result.
- **Question** is addressing what should be measured and the actual value of proposed change, and get answers based on the measurements.
- **Metric**: primarily involves evaluation of the actual implementation. Metric can consist of several variables which should provide insight on actual outcome of proposed change (i.e. expectation vs reality)

What are internal and external attributes of software?

Internal attributes corresponds to concrete property of the software, while **external attributes** can represent more general aspect of it on higher level.

Example, software maintainability both from internal and external attribute perspective.

- **External attribute:** Maintainability of the software
- **Internal attributes** which correspond to "Maintainability of the software":
  Test coverage - Size of the codebase in terms of lines - Quantity of the bugs not fixed - Structure - Coherence - Coupling - Modularity

Give one example of an internal attributes and one example of an external attribute of that you have used, or could have used, in your compulsory project in IN5140

No clear answer, will depend on your project.

Most important that you know the difference between internal and external attributes while discussing.

Exercise 5 (20%)

Case:

A Norwegian software company is in charge of the development of a large system that will be used in state schools in Norway. Ten Scrum teams develop the system; 3 teams are located in Norway, 7 in Asia. Some teams have members from both Norway and Asia. The project uses DevOps (Development and Operations). DevOps emphasizes collaboration and communication between software developers and other IT professionals. Building, testing, and releasing of software occur rapidly, frequently, and is claimed to be more reliable than other methods. However, the project contract follows a waterfall model, with defined phases, milestones, releases, etc.

Exercise 5 (20%)

1. **Discuss advantages and disadvantages of having daily standup meetings in this project**
2. Discuss the challenges of working agile in the development teams (using DevOps) when the project contract is based on the waterfall model.

1. Advantages with standup meetings

Daily standup meetings are especially important in the context of DevOps. DevOps encourages fast-paced development which demands even more interaction and collaboration among team members. They allow team members to communicate with each other and discuss problems and issues they are facing. It also enables the teams to be aware of what other peers are currently working on. This technique is very useful when applied across different Scrum teams, but is also very efficient for internal team discussions. It is also a great opportunity for team members of different nationalities to find common language and learn how to collaborate efficiently.

1. Disadvantages with standup meetings

Main challenge of daily standup meetings in the given project is that teams are located in different time zones. It means that teams located in Asia won't be able to efficiently communicate with the teams in Norway. As a result the teams might not be able to discuss urgent problems with each other during daily standup meetings. They still will be able to communicate with each other through email and other tools, but won't have opportunity to discuss cross-team issues face-to-face.

Also, language could be a disadvantage.

Exercise 5 (20%)

1. Discuss advantages and disadvantages of having daily standup meetings in this project
2. **Discuss the challenges of working agile in the development teams (using DevOps) when the project contract is based on the waterfall model.**

2. Challenges

Main challenge is connected to the planning stage which is completely different in waterfall and agile process models. In the waterfall model, planning and design of the system is done in the early beginning of the development process, while in agile (DevOps) scenario planning is continuous.

Unfortunately, developer teams will have to follow milestones mentioned in the contract. However, developers won't be forced to invest too much time into design by following waterfall model (meaning, they can start development earlier, in contrast to the waterfall model) ...

2. Challenges

… DevOps is also embracing short release cycles, which confronts with the waterfall model where releases and milestones are much less frequent. Short release cycles also imply fast feedback loop in which product owner should be involved. Based on the feedback, development teams can adjust goals and move forward faster. Since waterfall model is enforced by contract, feedback is expected only on releases listed in roadmap, which is not sufficient for DevOps.

In order to adapt waterfall to DevOps, development team can prioritize minimal set of absolutely essential features and deliver Minimum Viable Product (MVP) as soon as possible.