Topic Info 2: Agile Working

The agile manifest: the individual and his team plan together

From the beginning of the 1990's agile methods came up in the software development and gained first popularity with extreme programming (Kent Beck) in 1999. Agile methods have become more and more important in the last years. This applies above all - but not exclusively - to the IT sector: 62% of the software development projects partly work agilely in Germany, 16% even consistently, while only 22% of all projects exclusively work with conventional methods.

The agile philosophy underlines the ability of the individual and the self-organization of the development team. By means of "agile methods" it is tried to eliminate the deficiencies of the previous software models which are considered to be too rigid. As a rigid and end-to-end planning is not feasible anymore, the requirements of the software are not certain right from the beginning. Instead, they develop with the growing understanding of the software system which shall be achieved above all by an intensive communication with the customer. To make communication easier, extreme programming uses so-called "user stories", i.e. user stories on the description of system properties or important functions to make communication easier. The planning has to take place in cooperation and is not a separate decision of the project manager. The interaction of a group to plan the scope of the next (software) enable or the provision is called "planning game". Planning games and the estimation of the project expenses and efforts are periodically repeated (iterative process).

During a meeting in Utah 2001, the term agile was chosen for this type of software development and the "agile manifest" (www.agilemanifesto.org/iso/de/manifesto.html) was formulated.

"We open up better ways to develop software by doing it ourselves and **helping** others.
Thanks to this activity we appreciate these values:

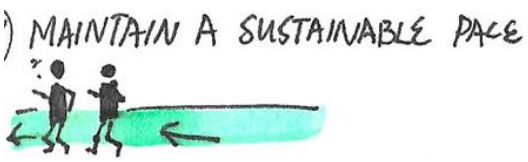
- **Individuals and interaction** are more important than processes and tools
- **Faultlessly working software** is more important than a comprehensive documentation
- **Cooperation with the customer** is more important than contract negotiation
- **Reaction to changes** is more important than following a plan."

Agile principles are the guiding principles for agile work. Sometimes agile principles are also called methods. **In the *agile manifest*, twelve principles are listed**.[1]

- Satisfaction of the customer by early and continuous delivery of valuable software
- Agile processes use changes (even late in the development) for a competitive edge of the customer
- Delivery of perfectly working software in regular, preferably short intervals (only few weeks or months)
- Nearly daily cooperation of experts and developers during the project (ex.: collective code ownership)
- Provision of the environment and the support which is required by motivated individuals to perform the tasks
- Transfer of information, if possible, during the face-to-face discussion
- The functionality of the software is deemed to be the most important measure of progress

- Maintaining a uniform pace of work by principals, developers and users for a sustainable development
- Constant focus on technical excellence and good design
- Simplicity is essential
- Self-organization of the team for planning and implementation
- Self-reflection of the team on the own behavior for an adaptation with respect to an efficiency increase

The principle of the sustainable pace: no overtime



One of the twelve agile principles is the sustainable pace. Beck says that there is the "40-hour week" as one of the twelve practices of the extreme programming. Overtime is a reference to serious problems in the project which have to be localized and solved. The developers are hardly able to encode easily and clearly due to permanent overtime. They become unproductive and program more mistakes. Roman Pichler, as a famous representative, writes correspondingly: in Scrum - the most common agile method - regular overtime is unacceptable.

"Agile software development is a collective term for a number of methods and practices which are based on values and principles of the manifest of agile software development." (Agile Alliance, 2018)

In **Scrum**, the worldwide most common agile practice, there are the roles of the product owner, the team and the Scrum master. The product owner has the role and opinion of the customer and describes the requirements which have to be implemented to achieve the project goal - in the so-called product backlog. This document continuously processes and prioritizes the requirements depending on the overall progress.

The requirements are implemented in so-called sprints of always equal length (2-4 weeks). In the sprint planning, the development team agrees on the requirements together with the product owner. The team chooses the tasks which it processes in the next sprint. It estimates the relating efforts and expenses to guarantee that it is feasible within the selected period of the sprint.

The results of the sprint are presented in the sprint review of the product owner (possibly also customers/users). Only tested and 'operable' functionalities are deemed to be settled.

The team organizes itself absolutely independently for the entire produces and has to have the corresponding resources and skills. The team reflects with work process and cooperation to determine improvement measures in the retrospective after the sprint. The Scrum master as expert for the method supports the team in observing the self-set rules and removing all obstacles.

Dr. Nadine Müller, ver.di (DE)