

Software Product Development Agile Technologies

Nowadays, the so-called Agile-methodologies, especially such as Scrum and Kanban are becoming increasingly popular. Scrum and Kanban software development technologies represent agile methodologies and are often used as synonyms. In fact, there are some significant differences between two of these two agile methodologies. Understanding these differences is a key to choosing the way that works best for your environment.

Index Terms — Scrum, Kanban, agile methodology, innovative methodology of project management, IT project management, project-oriented management.

Research Survey — Scrum and Kanban represent agile methodologies. In fact, there are some significant differences between these two agile methodologies. Understanding these differences is a key to choosing the way that works best for your environment. Scrum is a tool that is used to organize work at small, manageable units that can be completed by means of cross-functional team during a given period of time. The so-called sprint, which usually lasts for 2-4 weeks. In order to plan, organize, manage and optimize the process, Scrum identifies three proposed roles: the product owner is responsible for the start of planning, prioritization and communication with the rest of the company. Scrum Master is responsible for the process control during each sprint, and team members whose responsibility is to perform assignments of each Sprint, such as generation of a source code. Another common tool used by Scrum teams is a Scrum Board - a visual representation of the work flow, by means of manageable lay-outs, referred to as "stories", each story navigating along the board with a "delay", in the event that previous stories are in progress and have not yet been completed.

For the purposes of efficiency, Kanban can also be used as a tool in work organization. Similar to Scrum, Kanban also encourages the work to be broken down by manageable lay-outs, and uses Daily Kanban (very similar to Daily Scrum), in order to visualize the work incrementally the workflow. Where Scrum limits the amount of time, which is allocated for a particular scope of the work (using sprints), Kanban limits the amount of the work, allowed in the same state [1]–[3].

Both Scrum and Kanban are capable of splitting and performing large and complex tasks effectively. Both of them attach great importance to continuous improvement and optimization of the work process. In addition, both Scrum and Kanban put a very similar emphasis on the highly visible workflow, which keeps all of the team members in the touch with the progress of the work.

As has already been mentioned above, there are some differences both in philosophy, and in the practical application of Scrum and Kanban. As far as there are many individual differences, they can be grouped in the following manner:

1. Limited time iterations are mandatory in Scrum, while they are optional in Kanban. There may be individual frequencies for planning, release and improvement of the processes. Event-triggered iterations are also possible instead of time-limited iterations.

2. The Scrum Team undertakes to have a specific amount of work perform during this iteration. At the same time, in Kanban these responsibilities are optional.

3. It is performance that is used as the basic metric for planning and improvement of processes in Scrum. It is time of performance of the task that is used as the basic metric for planning and improvement of processes in Kanban

4. In Scrum, cross-function teams are mandatory. In contrast to Kanban, where cross-functional teams are optional. However, non-diversified teams are allowed.

5. In Scrum, tasks should be broken down into smaller ones so that they could be performed within one Sprint. In Kanban, there is no specific task sizes.

If we analyze the above mentioned, we can say that Kanban differs from other agile methodologies in the following:

Firstly, Kanban is not a specific process, but a system of values. So is Scrum. This means that no one will tell you what you should do and how you should do it step by step..

Secondly, the whole Kanban can be described in a simple phrase - "Reducing the work which is currently performed (work in progress)".

Thirdly, Kanban is a more "agile" methodology in comparison with Scrum. This means that it will not suit all the teams and all the projects. Moreover, this also means that the team, which uses Kanban, should be even more prepared for agile work than the team, which uses Scrum.

Speaking about practices in the use of Scrum and Kanban in software development projects, the following differences can be distinguished:

1. According to the schedule, Scrum processes are receiving special attention. The team provides a priority list of story points, which arte to be performed. The team must decide how many story points, in their opinion, they can have performed within one Sprint. Anything that is beyond the scope, should be postponed until the next sprint. The effective Scrum team gets to know its potential, within a number of sprints, and their estimates will be improved and optimized over the time. Then, every two weeks (depending on the number), the team generates a plan, and conducts a retrospective, in order to discuss the optimization of the process, and subsequently it passes into the next Sprint. This iterative process is designed to provide accurate estimates of the workflow and effective management of multiple projects. There are no required periods of time or iterations in Kanban team. Although the Kanban method is iterative in its nature, continuous improvement is expected to take place in the evolution, as the work is continuously performed. Restrictions placed on the different conditions in the workflow are not regulated at the beginning.

2. There are at least three roles, in the Scum team, that should be assigned to handle the work effectively: Product Owner, Scrum Master, and the team members. Each role has its own set of responsibilities, and they must work together in order to achieve an organized and effective balance. The team itself should be also a cross-

functional team, and have all the necessary resources to have the work performed. There are no established roles in Kanban. From the practical viewpoint, it only makes sense for project managers or supervisors, in particular, those involved in large, more complex Kanban projects, but this role, in theory, has to be developed in accordance with the needs of the project and organization. The Kanban team does not have to be a single-function team, since the Kanban workflow is designed for the use of any and all the teams participating in the project. Thus, a team of specialists and a separate team of generalists can work on different aspects of the same Kanban project, from the same board, and it is considered to be normal.

3. Scrum and Kanban are quite similar, but these are different things. In Scrum Board, columns are marked to reflect periods in the workflow, starting from and ending behind the sprint that is being performed or has been done. All user stories, added to the board at the beginning of each sprint, should be in the last column at the end of the sprint, unless the sprint was unsuccessful. After the sprint retrospective, the board is cleared and is aimed at the next sprint. The Kanban board has similarly labeled columns, in order to show the states of the workflow, but there is a vital difference: each column publishes the maximum number of allowed stories at any period of time. This ensures the compliance of certain Kanban limitations prescribed for each condition. Since each column has a limited number of stories allowed, there are no time boxes (for example, the length of the sprint); there are no reasons to reset the Kanban board. Incrementally the working process, it will continue to flow as long as the project is continued, with new stories added, if necessary, and with performed stories reassessed, if necessary.

Summing up the practical side of using Kanban software project development, three basic rules can be distinguished:

1. Break down the work into tasks, and particularize them.
2. Restrict the WIP (work in progress or work done simultaneously) at each stage of production.
3. Measure the cycle time (average time for the performance of the task), and constantly optimize the process in order to reduce the time spent.

Kanban has a total of 3 basic rules. For comparison, Scrum has 9 basic rules, XP has 13 basic rules; classic RUP has over 120 basic rules.

Recently, Kanban has been gaining a lot of popularity in the sphere of software development. Basically, Kanban is perfect for the following types of activities:

1. start-up work;
2. software support groups, where the is not that important "plan", but the response speed to changes is of vital importance ;
3. testing groups working separately from the development teams;
4. support services;
5. etc.

The experience leads us to the conclusion, that there is no need to choose between agile methodologies of Scrum and Kanban. One can get the best of both, Scrum and Kanban, the best that they can give to your own projects, and one can build your own project management methodology. Choosing the better from among these two methodologies, something, that meets your needs.

Agile methodologies provide an opportunity to experiment and adapt them to your own needs and objectives. In addition, one should keep in mind that Scrum and Kanban are only tools, they are adaptive and agile. Projects are inseparably linked with the strategy and ideology of the company. Project management rules are based on the objectives and process management rules.

Conclusion

Both Scrum and Kanban represent powerful, proven process tools that can significantly improve the quality of management of the software product development projects.

Kanban represents even a more "agile" methodology in comparison with SCRUM. It means that it will not suit all the teams and projects. Moreover, this also means that the team, which uses KANBAN, should be even more prepared for agile work than the team, which uses SCRUM.

It is quite acceptable to create a hybrid methodology based on Scrum and Kanban, if it suits your needs better.

References

[1] Innovative Projects and Programs Management Guidance: . V. 1, Version 1.2 / Trans. in Russian. language, ed. by S.D. Bushuyev. - K.: Nauk. svit, 2009. - 173 p.

[2] L.A. Ponomarenko Innovation Projects Management Computer Technologies. - K.: Kyiv. Nat. Trade and Economy. University Press, 2001. - 423 p.

[3] Guidance to the Pantology of Project Management (PMBOK® Guidance) Third Edition 2004 Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073-3299 USA / USA. – 401 p.