

RISKS MITIGATION OF THE AGILE SOFTWARE DEVELOPMENT METHODOLOGIES IMPLEMENTATION

Svirskiy O.V.

Svirskiy Oleg Vitalievich - Master of Economics, Senior Product Manager, VIMEO, NEW YORK

Abstract: *the article analyzes the reasons for the occurrence of possible risks in the implementation of agile methodologies of software development, as well as ways to reduce the likelihood of risk appearance in the development process. The author highlights the key criteria that allow us to identify high-risk projects when implementing agile product management methodologies, such as a stream of incoming tasks, changing the priorities of complex tasks, the need to synchronize several development teams, a low level of self-organization of teams, etc.*

Keywords: *agile methodologies, product development, project management, Scrum, SAFe, Kanban.*

Today, the implementation of agile methodologies (Scrum and analogues) today is becoming the de facto standard in the development of software products of varying degrees of complexity [2]. However, the analysis of the final and intermediate results of the implementation of this approach shows that in some cases the implementation of Scrum or similar methods either does not have a significant impact on the development process (while the organizational and financial costs of implementation can be significant), or has a negative impact of varying degrees of intensity.

In the course of the study, the task was set to develop a number of criteria that may indicate the potential ineffectiveness of the implementation of agile methodologies, as well as formulate recommendations to minimize risks at the decision-making stage.

The use of agile methodologies (as an example, we will consider Scrum as the most universal methodology) implies:

1. Dividing the organization into small, self-organizing cross-functional teams.
2. Dividing the entire scope of work into small specific components.
3. Division of time into short iterations (no more than 1-4 weeks).
4. Optimization of the release plan taking into account the previous changes.
5. Optimization of the process by means of a retrospective review after each iteration. [3, 13-14]

This provides a number of benefits, but if certain criteria are met, it can also be detrimental to the project.

As the criteria that translate the implementation of agile methodologies into the category of high-risk tasks, the study identified:

1. **A constant stream of tasks requiring prompt solutions.** This organization of work in the development department makes planning in the medium term difficult, which negates one of the key benefits of implementing agile methodologies. Implementing Scrum without taking into account the frequency of the appearance of new tasks leads to the fact that between the appearance of a task and its placement in the queue for execution, more time passes than it takes to solve the problem as such. The flexibility of the software development team is reduced significantly, which can be critical for certain business sectors [5] (Fig. 1).

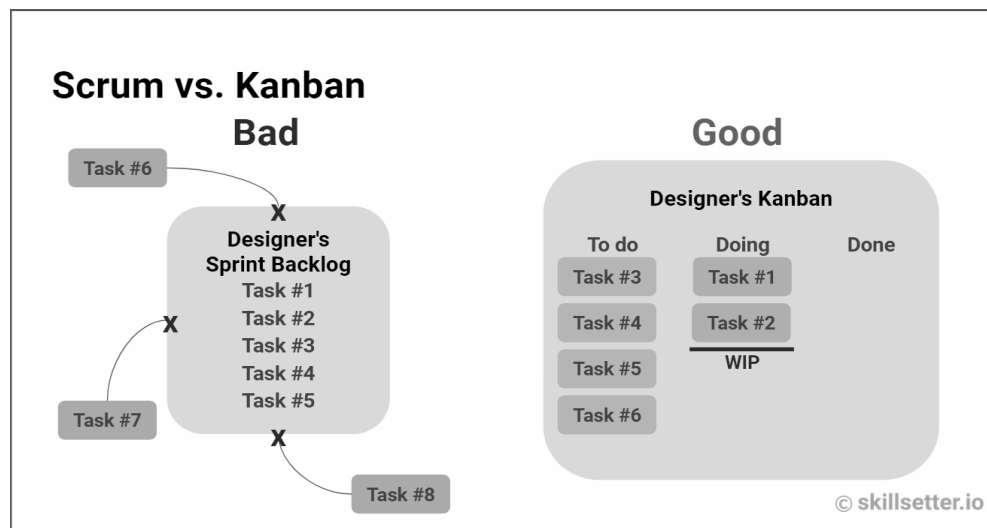


Fig. 1. Task stream prioritization using Kanban

2. **The presence of tasks with high complexity and high priority.** High complexity is understood as a combination of such factors as the impossibility of dividing the task into smaller ones, the impossibility of performing by the forces of one employee (one team) and the impossibility of implementation in a timely manner. Each of these factors in itself increases the risk, but the combination of two or more factors clearly indicates the high complexity of the task and translates it into the category of tasks requiring an individual approach to solving.

3. **The scale of the project, requiring the synchronization of several development teams.** This is not a strict limitation, but it increases the risks when implementing agile methodologies: in a significant number of cases, team synchronization requires a large amount of resources, comparable to the amount required to solve the main task. If the size of the project and the amount of resources allow, then the cost of coordination work can be considered acceptable. But this decision must be made on an individual basis after taking into account all the factors.

4. **Involvement of teams with a low level of self-organization.** The introduction of agile development methodologies demonstrates efficiency with a high independence of teams and a high level of self-management ability. In the absence of either of these two criteria, the potential benefit can be leveled out due to an increase in the consumption of resources for organizing interaction processes. Ignoring this criterion is typical for some satraps in the case when the personal ambitions of the leader or team members do not correspond to the level of self-organization or actual qualifications.

5. **High current performance indicators when working on a large-scale project.** Despite the fact that the introduction of agile methodologies makes it possible to increase the efficiency of software product development, at the implementation stage (it can take from 4 to 12 weeks), performance indicators inevitably decrease. The acceptability of such a reduction should be assessed in advance, and the implementation of Scrum or its analogues should be planned taking into account this factor.

The list of criteria formulated in the course of the research is not exhaustive; it can be specified or supplemented. However, the analysis of a large amount of data clearly demonstrates that it is precisely such prerequisites that are most typical for situations in which the implementation did not lead to the expected positive results.

To avoid risks or minimize them at the planning stage, implementation is necessary:

- Assess the current situation by analyzing the coincidence with the above criteria (coincidence on any of the points already translates the task into the high-risk category).
- Assess development efficiency and the need to improve it.
- Analyze the widest possible range of alternative methodologies that can be applied to optimize the development process (alternatives - Kanban, pair programming, Scaled Agile Framework, addition of the release manager position, etc.) [4].
- Use a combination of Scrum with engineering practices from other agile methodologies (for example, development practices from extreme programming, or the practice of analyzing and collecting requirements from ICONIX) [1, 16].

In conclusion, I would like to note that the high efficiency of agile development methodologies does not speak of their universality - and this is confirmed by practical data. In some cases, the manager's rash actions to implement the methodology that are not based on a comprehensive analysis of relevant data and can cause significant damage. At the same time, a wide range of alternative solutions or a combination of agile methodologies with other solutions, based on high-quality analytics, significantly reduces potential risks, regardless of the scope of development and the qualifications of the team.

References

1. *Wolfson B.* Agile development methodologies. SPb.: Peter, 2012. 112 p.
2. *Cohn Mike.* Succeeding with Agile: Software Development Using Scrum. Addison-Wesley, 2010. Computers. 475 pages
3. *Kniberg H.* Scrum and XP from the Trenches. InfoQ, 2015. 94 p.
4. *Kniberg H., Skarin M.* Kanban and Scrum: Making the most of it. 2010 C4Media Inc. 74 p.
5. *Svirsky O.V.* 5 situations when Scrum is not needed, and what to do about it // Skillsetter.io [Electronic resource]. URL: <https://skillsetter.io/blog/no-scrum/> (date of access: 04.02.2021).