THE MOST POPULAR SOFTWARE DEVELOPMENT MODELS

Abdulaxatov Muzaffarjon

Associate professor of the "Industrial Management and Digital Technology" department of the International Nordic University.

muzaffar.abdulakhatov@gmail.com

Abstract

Software development models provide structured approaches to planning, executing, and managing the process of creating software systems. These models cater to different project requirements, team dynamics, and organizational goals.

Keywords: Software, SDLC, Scrum, Waterfall, Agile, Kanban.

Introduction.

Software development models are systematic approaches used to structure, plan, and control the process of developing information systems. Over the years, several models have emerged, each catering to different project needs, team dynamics, and organizational goals.

The time has come to discuss the most popular software development models. We will delve into the pros and cons of each methodology. Such a comprehensive approach will help you choose the right option for your project.

So why do we need these SDLC methodologies to start with? Each of them aims at increasing the efficiency and speed of the application development process.

For your part, you will know about the tasks the development team is currently working on. Consequently, you will easily monitor and guide their activity. Such transparency will allow you to save money and streamline the software product development process.

Agile. To start with, Agile is not a methodology. It would be more accurate to call it a set of various methodologies and frameworks.

The principles of this approach are noted down in Manifesto for Agile Software Development. Let’s take a look at the most important of them.

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The main idea behind this approach lies in breaking a software development process into smaller parts. They are called iterations. Agile is all about making gradual changes at each iteration.

The key benefits of the Agile approach are:

- Fast delivery of a working solution/functionality/components
- Quick adaptation to the changing circumstances
- Active collaboration
- Quick issue detection
- Transparency
- Never-ending product improvement

If you considering adopting Agile methodologies, bear in mind its downsides, as in:

- Insufficient documentation
- Careful planning can be put aside for the sake of immediate reaction, adaptation, and improvement.

So when will the Agile methodologies work for your project? This option will fit the projects with cyclical processes where incremental results will bring additional value to the product.

Also, you should be ready to take an active part in the software development process. The software development team needs your constant feedback to respond quickly to your evolving needs and requirements.

At the same time, Agile will not work for you if you need a clear picture of the project timeline and budget.

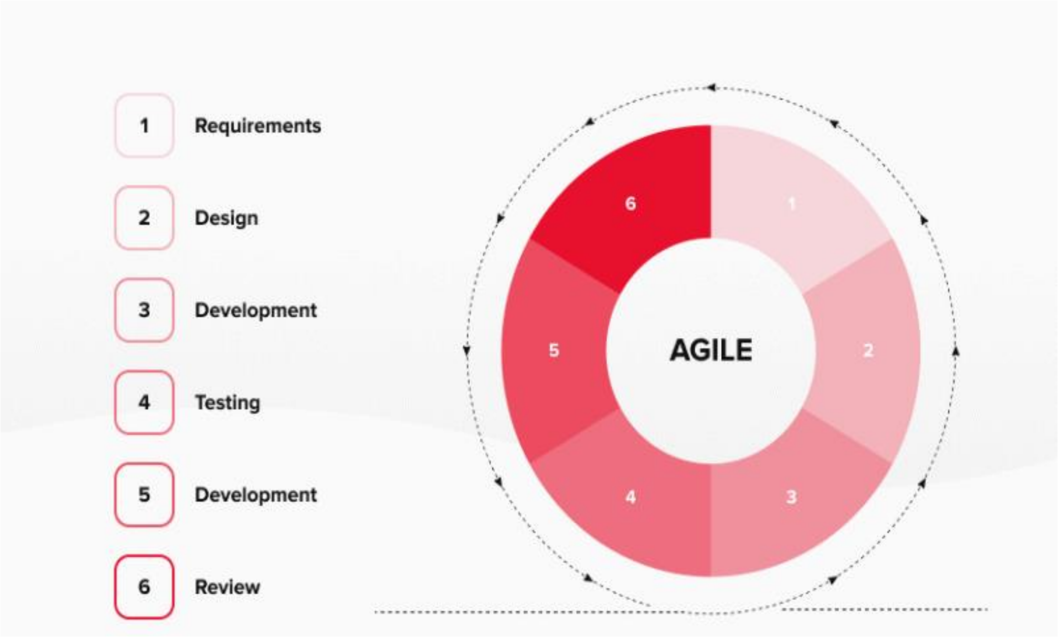

Figure 1. Agile principles for software development

Scrum. Though Agile and Scrum are often confused, the difference between these two terms is more than meets the eye. Agile is a broad concept. It is something like philosophy or orientation. Scrum, in its turn, is a specific methodology that defines how your project will be managed.

Speaking about sprints, they are short phrases that usually last two weeks. Each sprint starts with planning, where the software development team identifies which small part of the project scope should be completed during this time.

The tasks for each sprint are taken from a backlog. It is created by the product owner to prioritize tasks for the project.

Since Scrum can be seen as a subset of Agile, their benefits are pretty similar:

- Increased productivity
- Reduced costs
- Improved user satisfaction
- Fast release
- Great flexibility when it comes to changes.

However, like any other project management methodology, Scrum is not perfect and has the following downsides:

- It can be hard to adopt Scrum in big teams.
- The efficient Scrum adoption requires experienced software development professionals.
- A high level of commitment is a must.

Be aware that Scrum will not work for you if your project has fixed product specifications or when you need a unified solution to many issues. For example, it will not help if you lack product vision, good technical practices, or there is a conflict in the team.

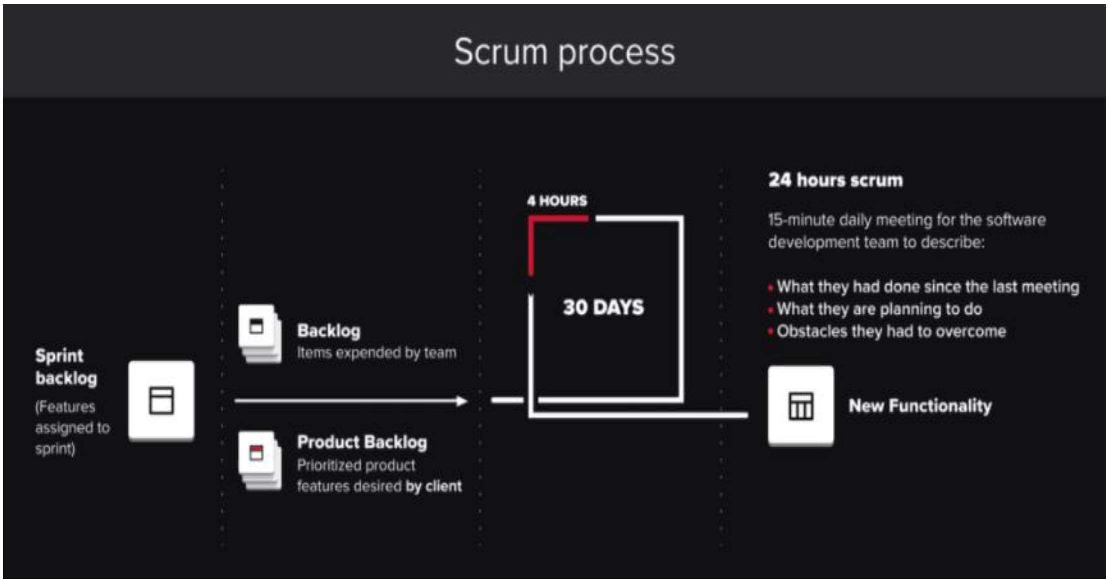
Scrum process
Sprint backlog
(Features assigned to sprint)
Backlog
Items expended by team
Product Backlog
Prioritized product features desired by client
4 HOURS
30 DAYS
24 hours scrum
15-minute daily meeting for the software development team to describe:
What they had done since the last meeting
What they are planning to do
Obstacles they had to overcome
New Functionality

Figure 2. An outline of Scrum process

Kanban. Kanban is a system aimed to visualize your workflow. The Kanban boards are used to track the current status of each task. Each software development phase gets its column.

In the same way, each task takes a separate card on the board. These cards are moved from column to column until the task is completed.

A traditional Kanban board has three columns, which are to-do, doing, and done. The first column is like a backlog. Here, you can see tasks that have not been taken yet. The second column contains tasks in progress. Finally, the last column shows completed tasks.

Finally, the Kanban approach will suit you if you want to concentrate on delivery rather than planning. Also, it is a good option when there are many incoming requests and each of them has a different size and priority.

Waterfall. The idea behind the waterfall methodology is completely different from the philosophy of Agile. You can proceed to the next stage of software development only if the previous one is completed. This linear approach does not suppose overlapping or going back.

In contrast to the Agile methodologies, constant status updates are not necessary. It means developers can work independently from the customer at each stage of the software development process. These stages are requirements, design, implementation, testing, and maintenance.

Ideally, there should be a certain stage gate between software development steps. You need to use it for reviewing and approving the project requirements. Only after that, the design stage can begin.

Let’s take a look at the key advantages of the Waterfall approach:

- A plain and simple sequence of steps
- The overall goal is defined much earlier
- Your active participation is not required
- Straightforward transfer of information at each stage
- It is easy to measure the project’s progress since you know its scope from the beginning.

However, we should not forget about the weak spots of the Waterfall methodology. These are:

- You will get a working solution only at the end of the software development process. As a result, your competitors may enter the market earlier and you will lose the market share.
- You may invest a lot of time and money to find out there is no actual interest in your product.
- Lack of flexibility

- Testing is performed closer to the project's end
- The approach leaves no room for sudden changes.

The Waterfall model works best for software development projects where changes can be too expensive or time-consuming. It can be the right option for stable projects with careful planning. However, it will not fit your project if it requires a great degree of flexibility.

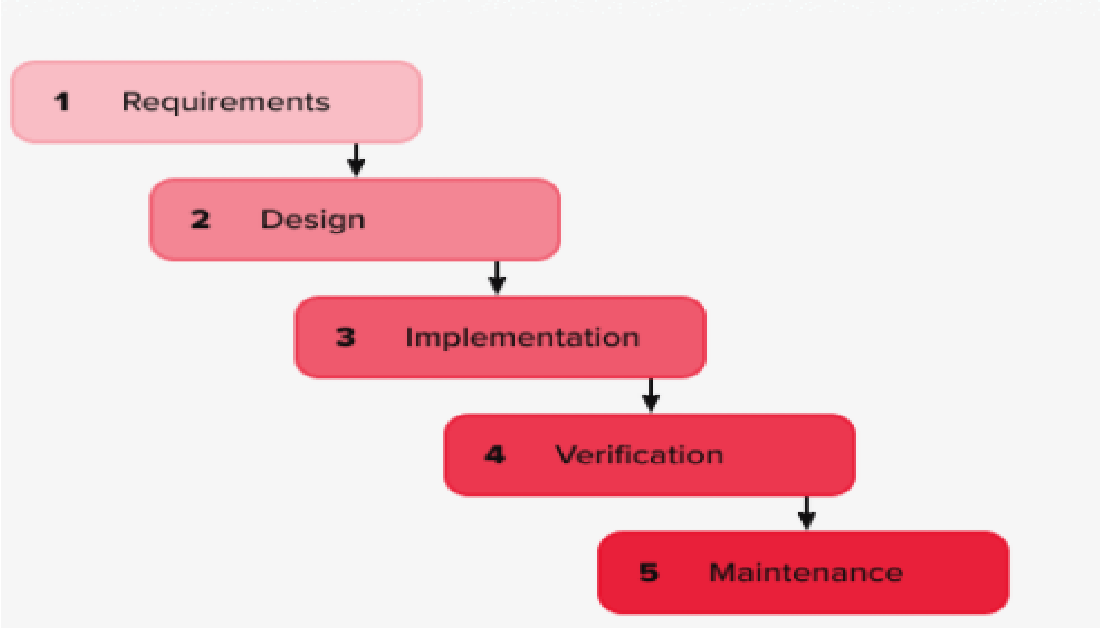

Figure 3. The waterfall software development structure.

Conclusion

In the realm of software development, selecting the right development model is crucial to the success of a project. Understanding the strengths and applications of each model enables project managers and development teams to make informed decisions, aligning their approach with project goals, stakeholder expectations, and resource availability. The right model enhances efficiency, adaptability, and ultimately, the success of software development endeavors.

REFERENCES

1. Abdulakhatov M.M, “Models and methodologies software development” Scientific electronic journal of Management and Future Technologies Volume 1, Issue 2, June, 2024.

2. Abdulakhatov M.M, “The application of digital marketing strategies” Scientific electronic journal of Management and Future Technologies Volume 1, Issue 1, April 2024

3. Abdulakhatov M. M, “The main impacts of digital transformation on small to medium-sized business enterprises” Scientific electronic journal of Digital Economy and Information Technologies No. 2, 2022

4. Abdulakhatov M. M “The digital economy is the basis for economic development” Scientific journal of Economy and Society No. 10, October 7, 2022

5. Kobilov A.U, Abdulakhatov M.M, Rajabov Sh.B, “Artificial intelligence as a technological innovation for economic development of the republic of Uzbekistan” ICFNDS ’22, December 15, 2022, Tashkent, TAS, Uzbekistan.

6. N. Ashrafi, The impact of software process improvement on quality: in theory and practice, Information & Management, 2003. 40(7): pp. 677-690.

7. J. Jiang, et al., An exploration of the relationship between software development process maturity and project performance, Information & Management, 2004(41): pp. 279-288.

8. G. Caprihan, Managing Software Performance in the Globally Distributed Software Development Paradigm, International Conference on Global Software Engineering, 2006.