NUTS AND BOLTS OF WATERFALL SOFTWARE DEVELOPMENT

Waterfall is a model of software development, where the process looks like a flow that goes through the phases of requirements analysis, design, implementation, testing, integration and support. This approach is perfect when you have a fixed list of requirements and a clear view of the final product.

Taking into account the peculiarities of your business, as well as risk criteria, time and involvement of stakeholders, our specialists will be able to determine whether the Waterfall software creation is an effective methodology for your case. In this article we will describe the main traits of this approach, its benefits and shortcomings.

Methods Overview

Following the Waterfall model, developers move from one stage to another strictly in sequence. First, the "requirements specification" stage is completed in full, resulting in a list of requirements for the software.

Once the requirements are fully defined, the team proceeds to the design phase. It is necessary to create documents that detail the way the software should look like and the plan for implementing these requirements for programmers. After the design is completely executed, the developers implement the project.

At the next stage of the process, specialists integrate individual components developed by different teams of programmers. After implementation and integration are completed, the product is tested and debugged; at this stage all the defects that appeared in the previous stages of development are eliminated. After that, the software product is implemented and your developer provides the support required - introduces new functionality and eliminates errors.

The Origin and the Main Stages of Waterfall Software Development

The Waterfall technique is the brainchild of Winston Walker Royce, director of the Lockheed Software Technology Center in Austin, Texas, USA, a pioneer in software building.

Walker's original work "Managing the development of large software systems" describes six stages of product building, which in 1985, the US Department of Defense established as a standard for working with software developers:

- System and software requirements: fixed in the PRD (product requirements document). Analysis: is embodied in models, schemes and business rules.
- Design: the internal architecture of the software and the ways of implementing the requirements are defined. This is not only about the interface and appearance of the software, but also its internal structural logic.
- Coding: the program code is directly written, the software is integrated.
- Testing: bug-testers check the final product, recording in the trackers information about the defects of the program code or functional. In case of errors and the availability of time/finance, bugs are fixed.
- Operations: the product adapts to different operating systems, is updated regularly to fix bugs detected by users and add functionality. The stage also provides technical support to customers.

Such rigid constraints of the sequence allow building a development process that is as transparent and convenient for the customer as possible.

This model implies a strictly consistent and one-time implementation of each phase of the project. The transition from one phase to another is possible only after successful completion of the previous stage. Each stage implies detailed planning and full correctness of the result of the stage.

Project and Documentation

The reverse side of the coin of Waterfall software development is the need to maintain and constantly update the product documentation. Any change must be agreed with the customer. And an insufficient level of requirements creates an increase in budget and timelines of the project, which are quite difficult to assess.

However, Waterfall is highly appropriate for a range of projects because of transparency. Due to the high level of formalisation, it is much easier to manage such a project. It is generally believed that the waterfall software development model reduces risks and clarifies the process when a dozen of people work on the project.

The Waterfall model is suitable for the development of complex and large projects and software systems with strictly defined functionality, when developing large state orders or scientific engineering.

Shortcomings of Waterfall Methodology

Among the shortcomings of the waterfall method I can note:

- The process lacks flexibility, so if the project requires more time and financial resources than possible, then this will have an impact on the testing phase. According to the research of our consulting group, the cost of fixing bugs after the product release is an average of 20 times higher than during a full-fledged multi-stage testing in the development process
- "Resistance" to changes - a rigid framework from the development stages and the condition of providing only a finished product determine an inability to make changes during development
- “Unchangeable” - Although in the first stages the forecast of temporary and financial expenditures may fluctuate, it is impossible to change the project in the direction of cost optimisation in features or concept before the release of the finished product
- Increased risks - the classical testing system implies testing of each component of the project separately, and interaction with the whole environment. When using Waterfall, only the finished software is tested.

The drawbacks of the Waterfall development model are partially corrected in the Waterfall modifications: Sashimi, Waterfall with subprojects and Waterfall development model with risk reduction.

- Sashimi or Waterfall model with layered phases - the most famous among them. The stages as well as in the original technique go one after another, but overlap one another in time.
- Waterfall with subprojects is a model in which you work with three large blocks: conceptualisation, requirements design and architectural structure of the product. Then for each of them you go through the stages (subprojects) of detailed design, coding, and testing. At the end, all components are integrated during the testing phase of the system.
- The waterfall software development model with risk reduction is a modification of the classic Waterfall, to which the risk reduction spirals are added that divide the project into mini projects and correspond to one or several key risks.

Advantages

Among the greatest advantages of the waterfall methodology are:

- Understandable and simple structure of the development process - this reduces the entry threshold for development teams
- Convenient reporting - you can easily track the resources, risks, time spent and finances due to the strict phase of the development process and detailed software documentation.
- Stability of tasks - the tasks facing the product are clear to the team from the very beginning of development, and remain unchanged throughout the process.
- Evaluation of the cost and timing of the project - the time of the release of the finished product, as well as its final cost can be calculated before the start of the software development.

Does the Waterfall Approach Suit Your Project?

This method will be a good choice if you have:

- A medium or a large project, where dozens of programmers and several different project teams should be involved.
- A project in which requirements and boundaries are transparent and you precisely know what the final product should be at the beginning of the software development lifecycle.

And finally, if the key terminator of your project is the implementation period, rather than finance, then Waterfall approach is exactly what you need.