

Вставлені картинки з посиланнями - це референси/ Показники відповідні, але тошнота досить висока через велику кількість ключовиків (основне слово - agile, але його ніяк не замінити і не забрати, зменшила за рахунок інших)

Agile software development Life Cycle

Contents

1. Intro
2. What is the Agile development Life Cycle?
3. The Agile process flow
 1. Concept / Purpose
 2. Inception / Requirements
 3. Iteration/ Development / Design
 4. Production and testing
 5. Retirement
4. Agile process model vs Waterfall model
 - Waterfall model
 - Agile development model
5. Outcome

Intro to Agile development Life Cycle

Modern software engineering is a highly complex team and cross-functional activity. For the correct and effective interaction of all participants in the process, one or another model, various tools and development methodologies are used. Taking an **Agile workflow** enables

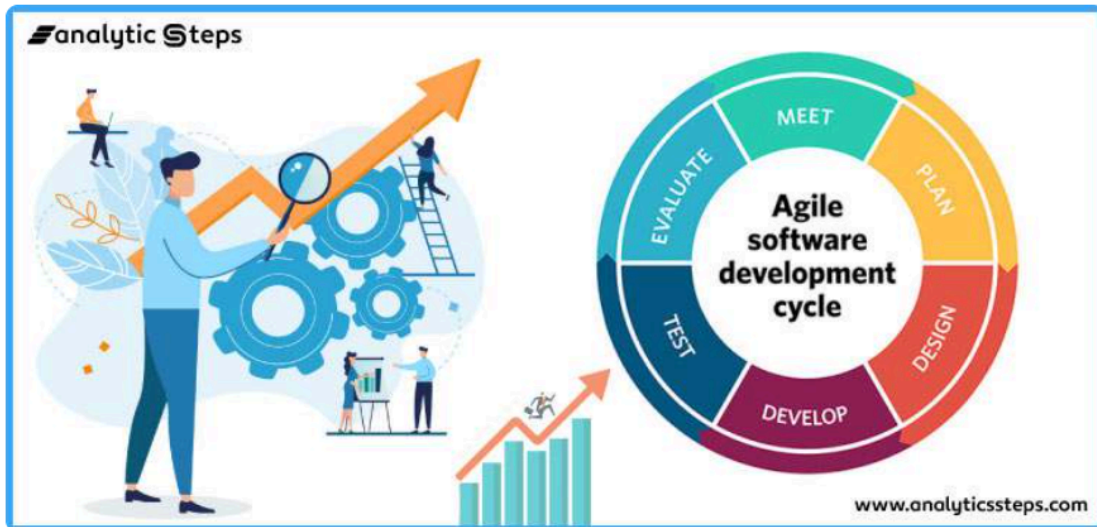
organizations to rapidly meet changing customer needs and achieve tremendous success.

Agile is a set of methods and practices for flexible project management in various application areas, from software development to the implementation of marketing strategies. The agile cycle is used to increase the creation speed of finished products and minimize risks through iterative execution, interactive cooperation of team members, and quick response to changes. Agile methodology is used by developers from Google, Netflix, Spotify, and other companies. Each company that works in this way follows certain Agile methodology steps. Examine the stages of the **Agile Software Development Lifecycle (SDLC)** to determine if the process will fit your team's needs.

What is the Agile development life cycle?

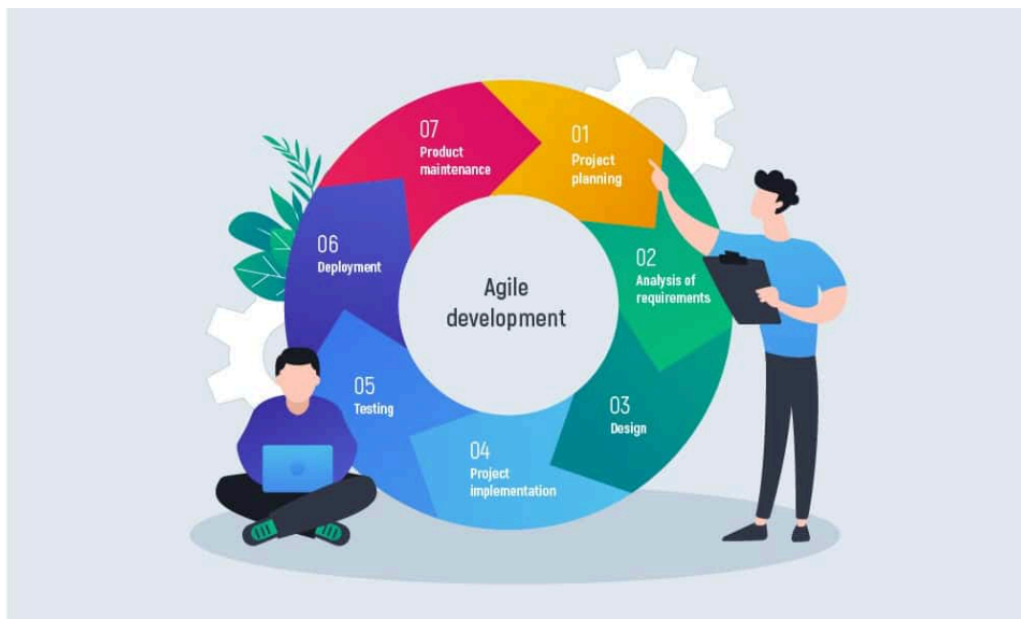
The Agile Development Cycle, or **Software Development Lifecycle Agile**, is a way of integrating an agile project management methodology into the software design process. Agile Software Development Cycle is also known as the iterative software development lifecycle, as it is based on continuous learning from iteration. The Agile life cycle is a way to implement a set of planning and management practices based on iterative and staged execution of tasks as required by the project. Focusing on teamwork, the **Agile system development lifecycle** allows teams to work on a project and make changes and additions during software elaboration to effectively achieve a project goal.

Building software with an agile life cycle consists of small iterations - short cycles - sprints, which are essentially small projects and take from one to four weeks. At the end of a separate productive period, an analysis is carried out and reorientation to new tasks of the next Agile cycle. The number of sprints can be any. Also, Agile methodology is very wide and offers a few different **Agile frameworks**, that you can use. Scrum and Kanban are two of the most common.



Today, in the technological world, we have come so far with most advanced technology whether it is machine learning or artificial intelligence. Research and development teams of several enterprises across the globe

sprints that are self-integral agile development steps. Here's how the steps of the agile development cycle diagram look like:



Each of these steps is important, includes its own requirements, challenges and assessments. However, the biggest impact on the **agile software development lifecycle** is done by the first two steps where

What's
Develk

Top A
7 Agil
Lifecy
Top 5

Reaso
Metho

What's

Agile l
a differ

How A

Wrapp



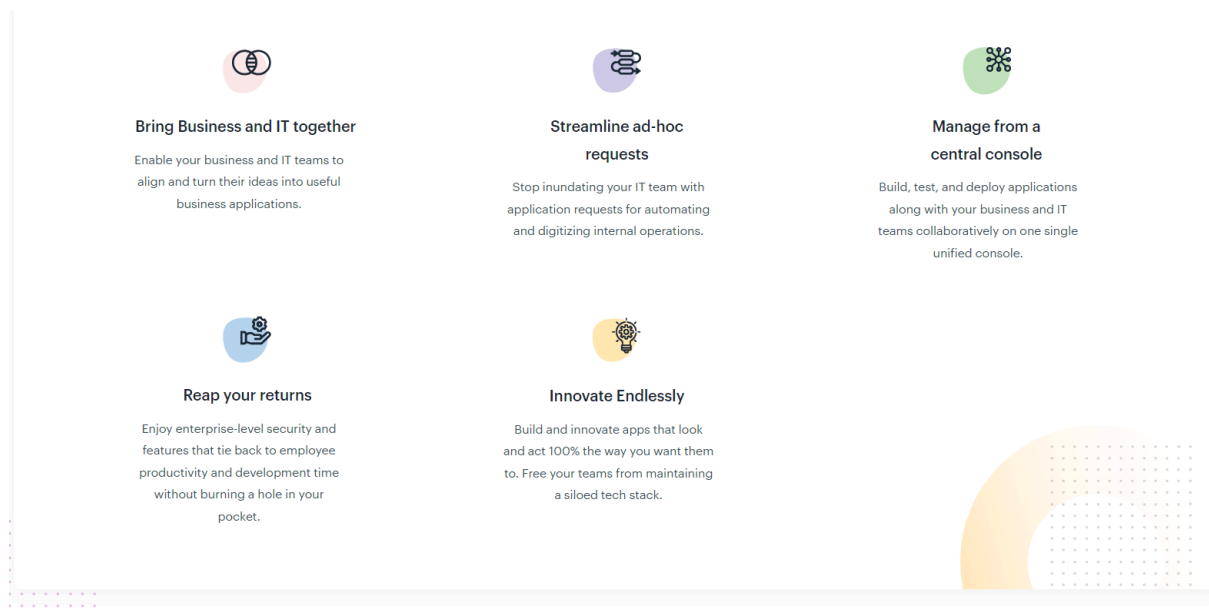
AGILE

The Agile process flow

Let's look closer at the **Agile methodology phases**. The Agile development life cycle can be divided into 5 different phases:

1. Concept / Purpose

It is the first of the agile SDLC phases. At this point, the purpose of the software and how the user interface will look are fine-tuned based on the requirements and interactions of the customer. It also identifies the business opportunity and the time it takes to complete the project. Thus, this Agile SDLC stage clearly clarifies the client's elaboration expectations.



2. Inception / Requirements

Agile software development process provides some initial requirements that have to be discussed and executed in this Agile stage, such as:

- Identifying team members (designers and UI / UX developers)
- Design modeling using architectural strategy and flowcharts
- Collection of initial support and fund

Solving these issues in the inception phase in agile is very important to quality work on the project.

3. Iteration/ Development / Design

After the first agile steps, the team of specialists starts working on software engineering based on the first iteration. The goal is to create a working product with minimal functionality at the end of the first sprint, which is then continually changed before being finalized. At this Agile project phase, the following steps are performed:

- Team association with stakeholders/customers
- Prioritization and implementation of iterations and functions
- Study and develop each iteration/sprint

- The regular release of working solutions
- Testing at the end of each step to ensure quality

4. Production and testing

Following the agile process steps, the software is reviewed by the team for any defects, errors, faults, and the registration of winnings and losses, prior to the release of the final product. This agile product life cycle phase also maintains the current software version and feedback from the client and users. The customer and users are guided by how to use the software. These agile development phases end when the release is scheduled for retirement. Various tasks performed during the production and testing phase:

- System testing and debugging
- Completion of system development and user documentation
- User guidance and training
- Expand the system

5. Retirement

The agile development cycle diagram shows the next stage - retirement. The last phase is dedicated to the decommissioning of the system. Old or obsolete systems are removed and replaced with new systems/software, which has minimal impact on business processes and keeps the process running smoothly. Agile methodology software development life cycle implies various reasons for releasing a system during the retirement phase, including:

- The system is out of date
- The release is no longer supported
- The system must be updated to a newer version to improve the business model

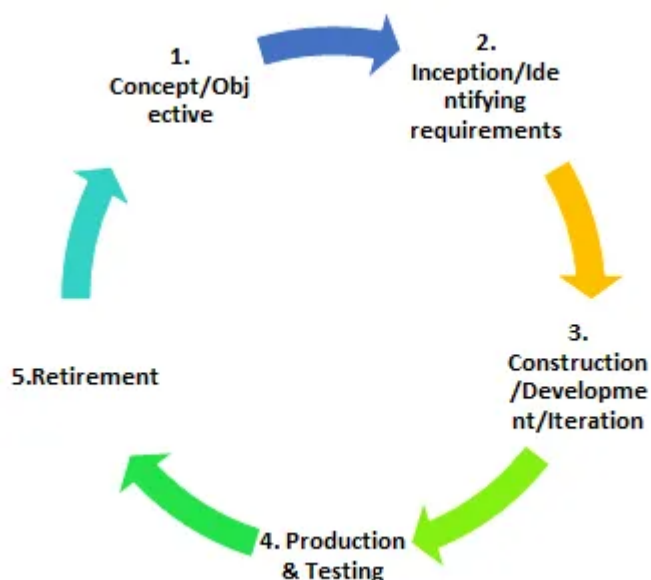
The agile model of software development life cycle concentrates on developing the right product with minimal cost and no error. The Agile

Development Life Cycle is the continuous growth and improvement of the development cycle. It covers change until the client's expectations are met.

Below, you can see an **Agile workflow diagram**. The agile process diagram shows how this approach works. Use agile system development and these steps to improve your business results.

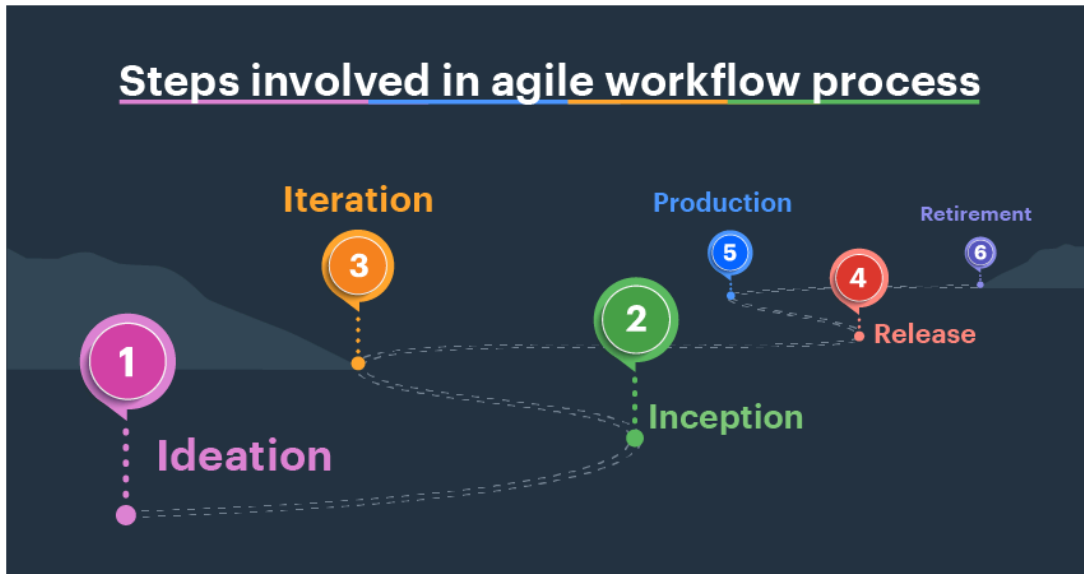
<https://www.smartsheet.com/understanding-agile-software-development-lifecycle-and-process-workflow>

Agile Software Development Lifecycle



of the customers and stakeholders.

Steps involved in agile workflow process



The agile workflow process generally remains the same regardless of the project involved. Here are the

<https://www.pngwing.com/en/free-png-dqddn>



Agile process model vs Waterfall model

There is some variability in the passage of the life cycle stages during the design and implementation of a product on the market. For each product, this happens in its own way. Still, in order to somehow manage the process, models of the software life cycle were formulated - a simplified and generalized idea of how the product develops. To better understand the principle and benefits of the **Agile development process flow**, we will compare it with another popular model - waterfall.

Waterfall model

A waterfall model is a model of the software elaboration process, the life cycle of which looks like a flow. The main essence of the Waterfall model is that the stages depend on each other and the next one begins when the previous one is completed, thus forming a progressive (waterfall) forward movement.

Teams of different stages do not communicate with each other, each team is clearly responsible for its stage. At any moment, you can easily understand whether this or that control point was passed or not, and whether the deadlines were met. For these reasons, long-term and substantial projects spanning decades and involving numerous participating organizations are primarily guided by a waterfall.

The disadvantages of this model are getting the result after going through all the stages and the difficulty of identifying errors. It's hard to go back. It is not clear what to return: if a failure occurs at some stage, its consequences are visible only at the end. As a result of the completion of the steps, intermediate products are formed that cannot be changed in subsequent steps.

Specifications:

- The exact cost and term are indicated in the contract
- It will take time to write and agree on the terms of reference
- To change the implementation requirements, you must sign an additional agreement
- The project cost is higher
- The cost of work includes stock in case of unforeseen labor costs

Agile development model

The agile SDLC model (iterative model) involves dividing the project into parts (stages, iterations) and passing through the stages of the life cycle

at each of them. Each agile model phase is complete in itself, the set of stages forms the final result.

With each agile development stage, it approaches the final desired result, or the requirements for the result are clarified in the course of design. Accordingly, at any moment in the agile SDLC process, the current iteration may be the last or next one on the way to completion.

Using an agile development process flow reduces the risks of global failure and waste of the entire budget, obtaining unjustified expectations and misunderstanding of processes by both the client and each member of the team. **Agile process workflow** also allows you to complete design at the end of any iteration (in a waterfall model, you must complete all stages first).

Specifications:

- Detailed TOR is not required
- At the start, there is no exact understanding of the budget, the estimate is approximate
- You can start developing immediately after signing the contract
- It is easy to change one or another implementation requirement if they have lost their relevance or the vision of the project has changed
- The project cost is lower
- The actual time spent is paid

Let's summarize it.

Waterfall model

1 Requirement

2 Design

3 Development

4 Testing

5 Support

Agile project lifecycle

1. Planning

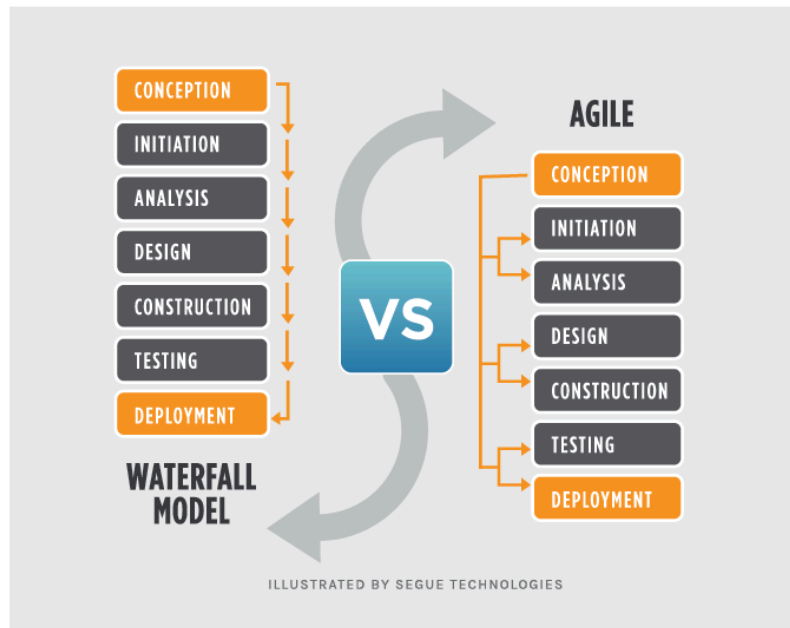
2. Development

3. Testing

4. Demonstration

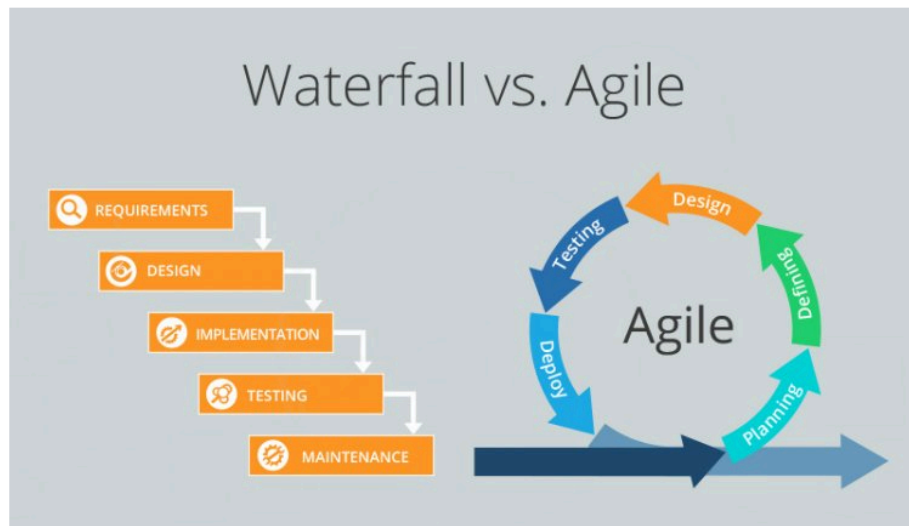
5. Implementation

The use of the Agile development lifecycle is justified in large projects, stretched out in time, with constant changes in user requirements; as well as in other cases where precise planning is impossible. The waterfall is suitable for small projects with well-defined requirements and the presence of specialists with the necessary qualifications. An excellent example of successful implementation of agile project management process flow - GudHub, an all-in-one, Low-code Platform. You can use the built-in toolkit in GudHub, run your applications in the platform cloud, or plugin an existing set of Agile requirements.



One of the first decisions we face for each of our project implementations at Segue is “Which development methodology should we use?” This is a topic that gets a lot of discussion (and often heated debate). If this is not something you’ve worked with before, a definition of development

Agile vs Waterfall: практика, которую вы должны знать



1. Степень гибкости

Outcome

Many teams that use software development lifecycle agile today combine techniques from different platforms with their own practices. Some teams are implementing the agile life cycle model and some rules (such as regular stand-ups, retrospectives, backlog maintenance, etc.), while others, such as agile marketing teams following the principles of the Agile Marketing Manifesto, are creating new agile practices.

In the future, with the **product development lifecycle agile**, teams will value their own performance over dogma. Openness, trust, and autonomy are slowly becoming valuable cultural resources for companies looking to attract the best talent and use them to their fullest. The experience of these companies confirms that teams can take different approaches to agile if they follow the fundamental principles of this methodology.

In this article, we tried to give you not only an agile methodology phases explanation but also bring you closer to understanding what the agile software development life cycle is. We also brought you an agile methodology diagram with **5 stages of agile development** and tried to prove its effectiveness by comparing it with another model.