

# Software Development Methodologies

by Devmountain Tutorials



## WHAT'S COVERED

In this lesson, you will learn how to apply the software development methodologies to given scenarios. Specifically, this lesson will cover:



## BEFORE YOU START

As information systems have evolved, so has the process of developing software. Recall that the software development life cycle (SDLC) provides general phases for what to do. Software methodologies contain those same phases but include processes, roles, and structures for how teams make it happen.

## 1. What are Software Development Methodologies?

A **software development methodology** is the process for *how* a team completes the software development phases. Due to the complexity of developing software, developers and engineers apply various methodologies to the development process. When you hear methodology, think of it as a “method” for how they do their job. The appropriate methodology depends on the focus, scope, complexity, time, and team.



## HINT

It is important to note that while these methodologies have defined processes and roles, the implementation can vary from company to company. The roles and responsibilities vary between methodologies, but the basic concepts are the same.

SDLC Methodology	Focus	Situations for Use
Waterfall	Accurate	<ul style="list-style-type: none"><li>• Accuracy is critical, and requirements don't change frequently.</li><li>• Comparable work has been done before or a detailed plan can be made with confidence.</li><li>• Changes late in the process are difficult or expensive.</li></ul>
		<ul style="list-style-type: none"><li>• Customer needs to be highly involved with the product because you aren't sure what they need.</li></ul>

Agile	Flexible	<ul style="list-style-type: none"> <li>• Development team collaborates well.</li> <li>• Market is frequently changing.</li> </ul>
Lean	Sustainable	<ul style="list-style-type: none"> <li>• Time and resources need to be managed closely.</li> <li>• Solutions are unclear and problems are complex.</li> <li>• New technology is being created or significant research and development (R&amp;D) is needed.</li> <li>• Development team is technically proficient.</li> </ul>

This isn't an exhaustive list of methodologies; there are many additional methodologies and more details if you're interested. For this course, we'll touch on the big three to give you a general idea of how different teams work.



#### TERM TO KNOW

### Software Development Methodology

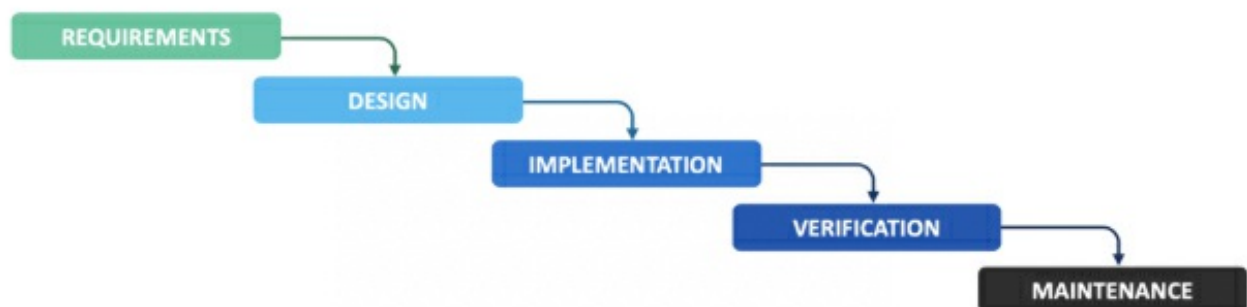
The process for how a team completes the software development phases.

## 2. What is Waterfall Development?

The waterfall methodology is a linear method where development happens sequentially. In this method, the project is broken down to ensure consistency. The steps at the beginning of the method flow down into subsequent steps, similar to a waterfall.

In **waterfall development**, all progress must move in one direction with the emphasis placed on planning and delivery dates. All details about what to build are created during the requirements phase with a statement of work or a scope document.

Changes to the requirements are managed very closely with change requests because redesign can be costly in this process.



### Waterfall Development



#### TERM TO KNOW

## Waterfall Development

A linear method where development happens sequentially. The steps at the beginning of the method flow down into subsequent steps, similar to a waterfall. Development flows in one direction where one step is completed before the next step begins.

---

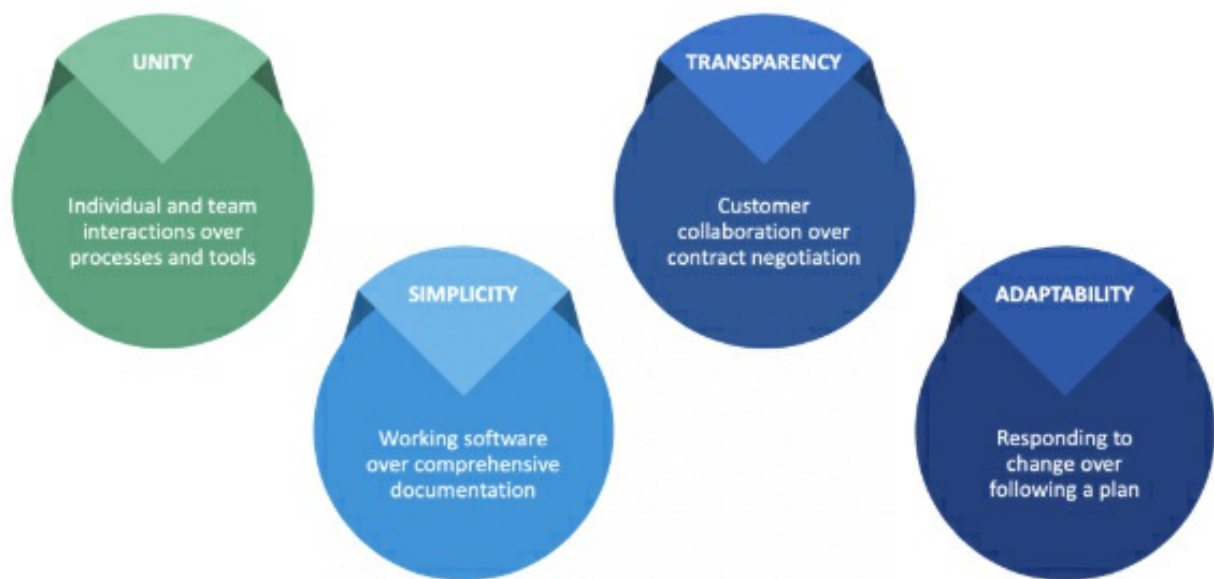
# 3. What is Agile Development?

Agile development includes a group of methodologies based on a 2001 [Agile Manifesto](#) written by 17 developers who wanted to find a better way to create software. They created a set of values and principles that have inspired several methodologies including Scrum, The Scaled Agile Framework (SAFe), and Kanban.

The goal of agile methodologies is to provide a flexible and iterative approach while ensuring a quality product. Agile is great for flexibility and products that need a high level of transparency or collaboration with users.

A key to agile development is an autonomous team that is empowered to make decisions.

This concept has taken the business world by storm as an alternative to command and control management. So even if you don't work in IT, you may hear about the agile principles.



## Agile Development



### TERM TO KNOW

#### Agile Development

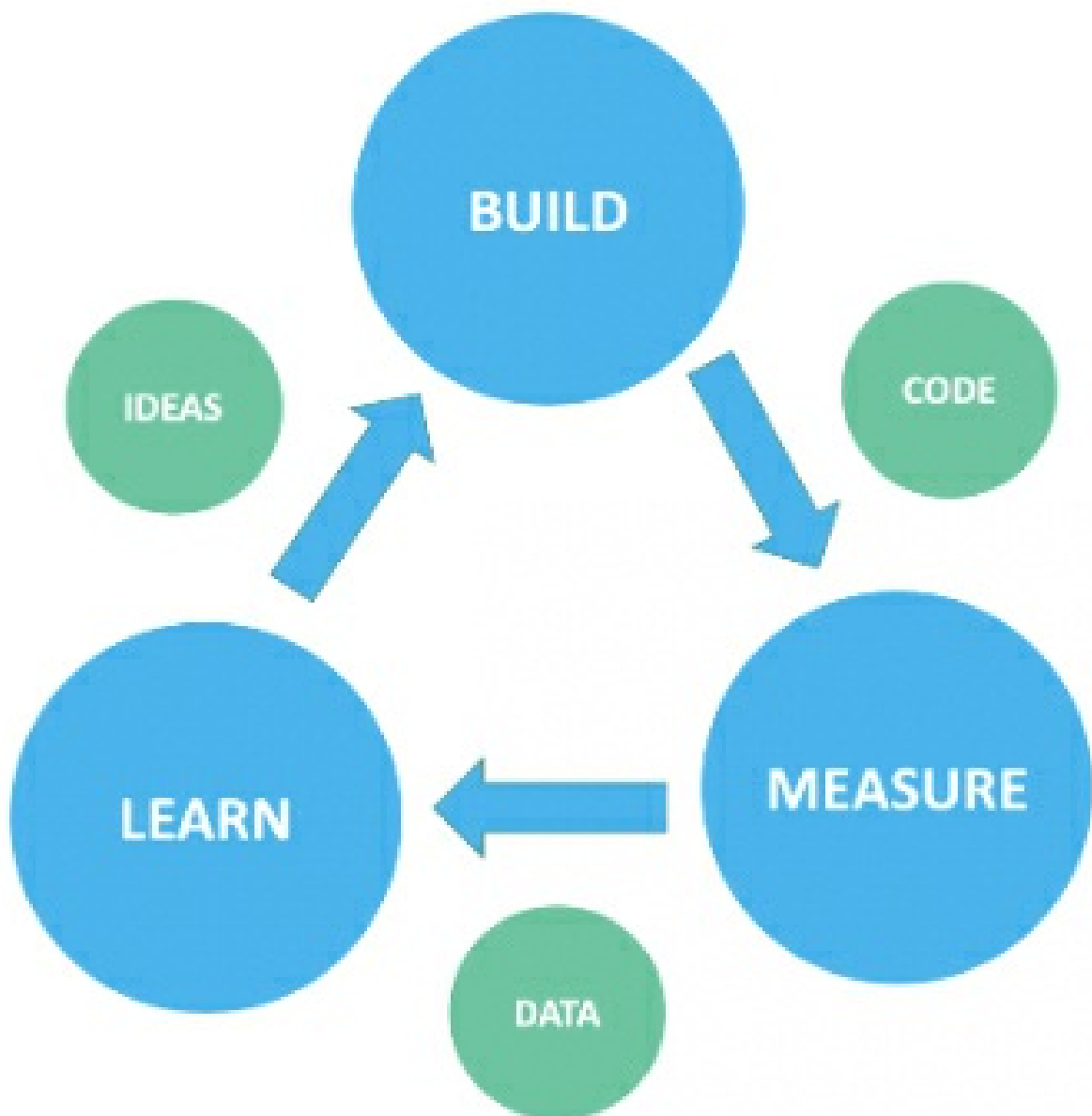
The goal of the agile methodologies is to provide a flexible and iterative approach while ensuring a quality product. Agile is great for flexibility and products that need a high level of transparency or collaboration with users. A key to agile development is an autonomous team that is empowered to make decisions.

## 4. What is Lean Development?

The lean development methodology comes from lean manufacturing and focuses on sustainable development through continuous improvement. The three main steps are build, measure, and learn.

In **lean development**, the team takes an idea and builds a minimum viable product (MVP). The MVP has just enough functionality to provide the team with feedback on how to improve the product. Once the MVP is developed, it is given to potential users to test the functionality and get feedback on how well they liked it.

After reviewing the feedback, the team determines whether it should continue in the same direction or rethink the concept. Several **iterations** of the MVP are developed with new functions added each time based on the feedback until a final product is completed.



**Lean Development**



## TERMS TO KNOW

### Lean Development

Lean development comes from lean manufacturing and focuses on sustainable development through continuous improvement. The three main steps are build, measure, and learn. In lean development, the team takes an idea and builds a minimum viable product (MVP).

### Iteration

Repeat the process for an idea, design, or product until it is ready, as in the developmental changes that are made to a prototype to make it better with each successive design.



## SUMMARY

This lesson explores **software development methodologies**, which provide a structured approach to completing the software development life cycle. Three main methodologies are discussed: **waterfall**, **agile**, and **lean development**. Waterfall is a linear method where development happens sequentially, while agile development provides a flexible and iterative approach, empowering autonomous teams to make decisions. Lean development focuses on sustainable development through continuous improvement, using a build-measure-learn cycle to develop an MVP and gain feedback from potential users to improve the product. The appropriate methodology for a given situation depends on various factors such as focus, scope, complexity, time, and team.

Source: This tutorial was authored by DEVMOUNTAIN and Sophia Learning. Please see our [Terms of Use](#).



## TERMS TO KNOW

### Agile Development

The goal of the agile methodologies is to provide a flexible and iterative approach while ensuring a quality product. Agile is great for flexibility and products that need a high level of transparency or collaboration with users. A key to agile development is an autonomous team who is empowered to make decisions.

### Iteration

Repeat the process for an idea, design, or product until it is ready, as in the developmental changes that are made to a prototype to make it better with each successive design.

### Lean Development

Lean development comes from lean manufacturing and focuses on sustainable development through continuous improvement. The three main steps are build, measure, and learn. In lean development the team takes an idea and builds a minimum viable product (MVP).

### Software Development Methodology

The process for how a team completes the software development phases.

### Waterfall Development

A linear method where development happens sequentially. The steps at the beginning of the method flow down into subsequent steps, similar to a waterfall. Development flows in one direction where one step is completed before the next step begins.

