Software development methodologies

Waterfall Methodology

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

Indice

1. Introduction
   - Key Takeaways
   - What is the Waterfall methodology?
2. Stages of the Waterfall
   - Stage #1: Requirements
   - Stage #2: Design
   - Stage #3: Implementation
   - Stage #4: Verification
   - Stage #5: Maintenance
3. Benefits of Waterfall methodology
4. Limitations of Waterfall methodology

Indice

1. Introduction
   - Key Takeaways
   - What is the Waterfall methodology?
2. Stages of the Waterfall
   - Stage #1: Requirements
   - Stage #2: Design
   - Stage #3: Implementation
   - Stage #4: Verification
   - Stage #5: Maintenance
3. Benefits of Waterfall methodology
4. Limitations of Waterfall methodology

Key Takeaways

- Waterfall methodology is a linear, sequential project management approach with distinct phases and rigid progression;
- It works well with projects that have well defined requirements and predictable outcomes but lacks flexibility for change;
- Waterfall's strengths include clear structure, documentation, and upfront planning, but it can struggle with innovation and feedback;
- Choose waterfall for projects with fixed requirements.

Key Takeaways

- Waterfall methodology is a linear, sequential project management approach with distinct phases and rigid progression;
- It works well with projects that have well defined requirements and predictable outcomes but lacks flexibility for change;
- Waterfall's strengths include clear structure, documentation, and upfront planning, but it can struggle with innovation and feedback;
- Choose waterfall for projects with fixed requirements.

Key Takeaways

- Waterfall methodology is a linear, sequential project management approach with distinct phases and rigid progression;
- It works well with projects that have well defined requirements and predictable outcomes but lacks flexibility for change;
- Waterfall's strengths include clear structure, documentation, and upfront planning, but it can struggle with innovation and feedback;
- Choose waterfall for projects with fixed requirements.

Key Takeaways

- Waterfall methodology is a linear, sequential project management approach with distinct phases and rigid progression;
- It works well with projects that have well defined requirements and predictable outcomes but lacks flexibility for change;
- Waterfall's strengths include clear structure, documentation, and upfront planning, but it can struggle with innovation and feedback;
- Choose waterfall for projects with fixed requirements.

What is the Waterfall methodology? (1)

Waterfall methodology is a well-established project management workflow. Like a waterfall, each process phase cascades downward sequentially through **five stages** (requirements, design, implementation, verification, and maintenance).

The methodology comes from computer scientist Winston Royce's 1970 research paper on software development. Although Royce never named this model "waterfall", he gets credit for creating a linear, rigorous project management system.

Approfondimento: [1]

What is the Waterfall methodology? (1)

Waterfall methodology is a well-established project management workflow.
Like a waterfall, each process phase cascades downward sequentially through **five stages** (requirements, design, implementation, verification, and maintenance).

The methodology comes from computer scientist Winston Royce's 1970 research paper on software development. Although Royce never named this model "waterfall", he gets credit for creating a linear, rigorous project management system.

Approfondimento: [1]

What is the Waterfall methodology? (2)

Unlike other methods, such as the Agile methodology, Waterfall doesn't allow flexibility.

You must finish one phase before beginning the next. Your team can't move forward until they resolve any problems. Moreover, as our introduction to project management guide outlines, your team can't address bugs or technical debt if it's already moved on to the next project phase.

What is the Waterfall methodology? (2)

Unlike other methods, such as the Agile methodology, Waterfall doesn't allow flexibility.

You must finish one phase before beginning the next. Your team can't move forward until they resolve any problems. Moreover, as our introduction to project management guide outlines, your team can't address bugs or technical debt if it's already moved on to the next project phase.

Indice

1. Introduction
   - Key Takeaways
   - What is the Waterfall methodology?
2. Stages of the Waterfall
   - Stage #1: Requirements
   - Stage #2: Design
   - Stage #3: Implementation
   - Stage #4: Verification
   - Stage #5: Maintenance
3. Benefits of Waterfall methodology
4. Limitations of Waterfall methodology

Stage #1: Requirements (1)

The requirements phase states what the system should do.
At this stage, you determine the project's scope, from business obligations to user needs. This gives you a 30,000-foot overview of the entire project.

The requirements should specify:

- Resources required for the project;
- What each team member will work on and at what stage;
- A timeline for the entire project, outlining how long each stage will take;
- Details on each stage of the process.

Stage #1: Requirements (1)

The requirements phase states what the system should do.
At this stage, you determine the project's scope, from business obligations to user needs. This gives you a 30,000-foot overview of the entire project.

The requirements should specify:

- Resources required for the project;
- What each team member will work on and at what stage;
- A timeline for the entire project, outlining how long each stage will take;
- Details on each stage of the process.

Stage #1: Requirements (2)

But these requirements *"may range from very abstract to a detailed mathematical specification,"* writes Steven Zeil, professor of computer science at Old Dominion University.

That's because requirements might not outline an exact implementation, and that's something development addresses in later stages.

Stage #1: Requirements (2)

But these requirements *“may range from very abstract to a detailed mathematical specification,”* writes Steven Zeil, professor of computer science at Old Dominion University.

That’s because requirements might not outline an exact implementation, and that’s something development addresses in later stages.

Stage #2: Design (1)

After gathering all the requirements, it's time to move on to the design stage.
Here, designers develop solutions that meet the requirements.

In this stage, designers:

- Create schedules and project milestones;
- Determine the exact deliverables;
- Create designs and/or blueprints for deliverables.

Stage #2: Design (1)

After gathering all the requirements, it's time to move on to the design stage.
Here, designers develop solutions that meet the requirements.

In this stage, designers:

- Create schedules and project milestones;
- Determine the exact deliverables;
- Create designs and/or blueprints for deliverables.

Stage #2: Design (2)

Deliverables could include software or they could consist of a physical product. For instance, designers determine the system architecture and use cases for software. For a physical product, they figure out its exact specifications for production.

Stage #3: Implementation

Once the design is finalized and approved, it's time to implement it.
Design hands off their specifications to developers to build.

To accomplish this, developers:

- Create an implementation plan;
- Collect any data or research needed for the build;
- Assign specific tasks and allocate resources among the team.

Here is where you might even find out that parts of the design that can't be implemented. **If it's a huge issue, you must step back and re-enter the design phase.**

Stage #3: Implementation

Once the design is finalized and approved, it's time to implement it.
Design hands off their specifications to developers to build.

To accomplish this, developers:

- Create an implementation plan;
- Collect any data or research needed for the build;
- Assign specific tasks and allocate resources among the team.

Here is where you might even find out that parts of the design that can't be implemented. **If it's a huge issue, you must step back and re-enter the design phase.**

Stage #3: Implementation

Once the design is finalized and approved, it's time to implement it.
Design hands off their specifications to developers to build.

To accomplish this, developers:

- Create an implementation plan;
- Collect any data or research needed for the build;
- Assign specific tasks and allocate resources among the team.

Here is where you might even find out that parts of the design that can't be implemented. **If it's a huge issue, you must step back and re-enter the design phase.**

Stage #4: Verification

After the developers code the design, it's time for Quality Assurance (QA).
It's important to test for all use cases to ensure a good user experience. That's because you don't want to release a buggy product to customers.

QA also:

- Writes test cases;
- Documents any bugs and errors to be fixed;
- Tests one aspect at a time;
- Determines which QA metrics to track;
- Covers a variety of use case scenarios and environments.

Stage #4: Verification

After the developers code the design, it's time for Quality Assurance (QA).
It's important to test for all use cases to ensure a good user experience. That's because you don't want to release a buggy product to customers.

QA also:

- Writes test cases;
- Documents any bugs and errors to be fixed;
- Tests one aspect at a time;
- Determines which QA metrics to track;
- Covers a variety of use case scenarios and environments.

Stage #5: Maintenance

After the product release, devs might have to squash bugs. Customers let your support staff know of any issues that come up. Then, it's up to the team to address those requests and release newer versions of your product.

As you can see, each stage depends on the one that comes before it. It doesn't allow for much error between or within phases.

For example, if a stakeholder wants to add a requirement when you're in the verification phase, you'll have to re-examine the entirety of your project. That could mean tossing the whole thing out and starting over.

Stage #5: Maintenance

After the product release, devs might have to squash bugs. Customers let your support staff know of any issues that come up. Then, it's up to the team to address those requests and release newer versions of your product.

As you can see, each stage depends on the one that comes before it. It doesn't allow for much error between or within phases.

For example, if a stakeholder wants to add a requirement when you're in the verification phase, you'll have to re-examine the entirety of your project. That could mean tossing the whole thing out and starting over.

Stage #5: Maintenance

After the product release, devs might have to squash bugs. Customers let your support staff know of any issues that come up. Then, it's up to the team to address those requests and release newer versions of your product.

As you can see, each stage depends on the one that comes before it. It doesn't allow for much error between or within phases.

For example, if a stakeholder wants to add a requirement when you're in the verification phase, you'll have to re-examine the entirety of your project. That could mean tossing the whole thing out and starting over.

Indice

1. Introduction
   - Key Takeaways
   - What is the Waterfall methodology?
2. Stages of the Waterfall
   - Stage #1: Requirements
   - Stage #2: Design
   - Stage #3: Implementation
   - Stage #4: Verification
   - Stage #5: Maintenance
3. Benefits of Waterfall methodology
4. Limitations of Waterfall methodology

Benefits of Waterfall methodology (1)

The benefits of Waterfall methodology have made it a lasting workflow for projects that rely on a fixed outcome. A 2020 survey found that 56% of project professionals had used traditional, or Waterfall, models in the previous year.

A few benefits of Waterfall planning include:

Clear project structure

Waterfall leaves little room for confusion because of rigorous planning. There is a clear end goal in sight that you're working toward.

Benefits of Waterfall methodology (1)

The benefits of Waterfall methodology have made it a lasting workflow for projects that rely on a fixed outcome. A 2020 survey found that 56% of project professionals had used traditional, or Waterfall, models in the previous year.

A few benefits of Waterfall planning include:

Clear project structure

Waterfall leaves little room for confusion because of rigorous planning. There is a clear end goal in sight that you're working toward.

Benefits of Waterfall methodology (2)

Set costs

The rigorous planning ensures that the time and cost of the project are known upfront.

Easier tracking

Assessing progress is faster because there is less cross-functional work.

A replicable process

If a project succeeds, you can use the process again for another project with similar requirements.

Benefits of Waterfall methodology (2)

Set costs

The rigorous planning ensures that the time and cost of the project are known upfront.

Easier tracking

Assessing progress is faster because there is less cross-functional work.

A replicable process

If a project succeeds, you can use the process again for another project with similar requirements.

Benefits of Waterfall methodology (2)

Set costs

The rigorous planning ensures that the time and cost of the project are known upfront.

Easier tracking

Assessing progress is faster because there is less cross-functional work.

A replicable process

If a project succeeds, you can use the process again for another project with similar requirements.

Benefits of Waterfall methodology (3)

Comprehensive project documentation

The Waterfall methodology provides you with a blueprint and a historical project record so you can have a comprehensive overview of a project.

Improved risk management

The abundance of upfront planning reduces risk. It allows developers to catch design problems before writing any code.

Benefits of Waterfall methodology (3)

Comprehensive project documentation

The Waterfall methodology provides you with a blueprint and a historical project record so you can have a comprehensive overview of a project.

Improved risk management

The abundance of upfront planning reduces risk. It allows developers to catch design problems before writing any code.

Benefits of Waterfall methodology (4)

Enhanced responsibility and accountability

Teams take responsibility within each process phase. Each phase has a clear set of goals, milestones, and timelines.

More precise execution for a non-expert workforce

Waterfall allows less-experienced team members to plug into the process.

Fewer delays because of additional requirements

Since your team knows the needs upfront, there isn't a chance for additional asks from stakeholders or customers.

Benefits of Waterfall methodology (4)

Enhanced responsibility and accountability

Teams take responsibility within each process phase. Each phase has a clear set of goals, milestones, and timelines.

More precise execution for a non-expert workforce

Waterfall allows less-experienced team members to plug into the process.

Fewer delays because of additional requirements

Since your team knows the needs upfront, there isn't a chance for additional asks from stakeholders or customers.

Benefits of Waterfall methodology (4)

Enhanced responsibility and accountability

Teams take responsibility within each process phase. Each phase has a clear set of goals, milestones, and timelines.

More precise execution for a non-expert workforce

Waterfall allows less-experienced team members to plug into the process.

Fewer delays because of additional requirements

Since your team knows the needs upfront, there isn't a chance for additional asks from stakeholders or customers.

Indice

1. Introduction
   - Key Takeaways
   - What is the Waterfall methodology?
2. Stages of the Waterfall
   - Stage #1: Requirements
   - Stage #2: Design
   - Stage #3: Implementation
   - Stage #4: Verification
   - Stage #5: Maintenance
3. Benefits of Waterfall methodology
4. Limitations of Waterfall methodology

Limitations of Waterfall methodology (1)

Waterfall isn't without its limitations, which is why many product teams opt for an Agile methodology. The Waterfall method works wonders for predictable projects but falls apart on a project with many variables and unknowns.

Let's look at some other limitations of Waterfall planning:

Longer delivery times

The delivery of the final product could take longer than usual because of the inflexible step-by-step process, unlike in an iterative process like Agile or Lean.

Limitations of Waterfall methodology (1)

Waterfall isn't without its limitations, which is why many product teams opt for an Agile methodology. The Waterfall method works wonders for predictable projects but falls apart on a project with many variables and unknowns.

Let's look at some other limitations of Waterfall planning:

Longer delivery times

The delivery of the final product could take longer than usual because of the inflexible step-by-step process, unlike in an iterative process like Agile or Lean.

Limitations of Waterfall methodology (2)

Limited flexibility for innovation

Any unexpected occurrence can spell doom for a project with this model. One issue could move the project two steps back.

Limited opportunities for client feedback

Once the requirement phase is complete, the project is out of the hands of the client.

Limitations of Waterfall methodology (2)

Limited flexibility for innovation

Any unexpected occurrence can spell doom for a project with this model. One issue could move the project two steps back.

Limited opportunities for client feedback

Once the requirement phase is complete, the project is out of the hands of the client.

Limitations of Waterfall methodology (3)

Tons of feature requests

Because clients have little say during the project's execution, there can be a lot of change requests after launch, such as addition of new features to the existing code. This can create further maintenance issues and prolong the launch.

Deadline creep

If there's a significant issue in one phase, everything grinds to a halt. Nothing can move forward until the team addresses the problem. It may even require you to go back to a previous phase to address the issue.

Limitations of Waterfall methodology (3)

Tons of feature requests

Because clients have little say during the project's execution, there can be a lot of change requests after launch, such as addition of new features to the existing code. This can create further maintenance issues and prolong the launch.

Deadline creep

If there's a significant issue in one phase, everything grinds to a halt. Nothing can move forward until the team addresses the problem. It may even require you to go back to a previous phase to address the issue.

Riferimenti e approfondimenti

[1] Atlassian, *Waterfall Methodology: A Comprehensive Guide*, [Online; accessed 25-February-2026], 2026. indirizzo: `https://www.atlassian.com/agile/project-management/waterfall-methodology`.