

---

---

# Agile Principles

Agile is a mindset

---

# History of the Waterfall Model



**First introduced by Winston Royce**

**Was not referred to as Waterfall until later**

**Requirements specification**

**Design**

**Construction**

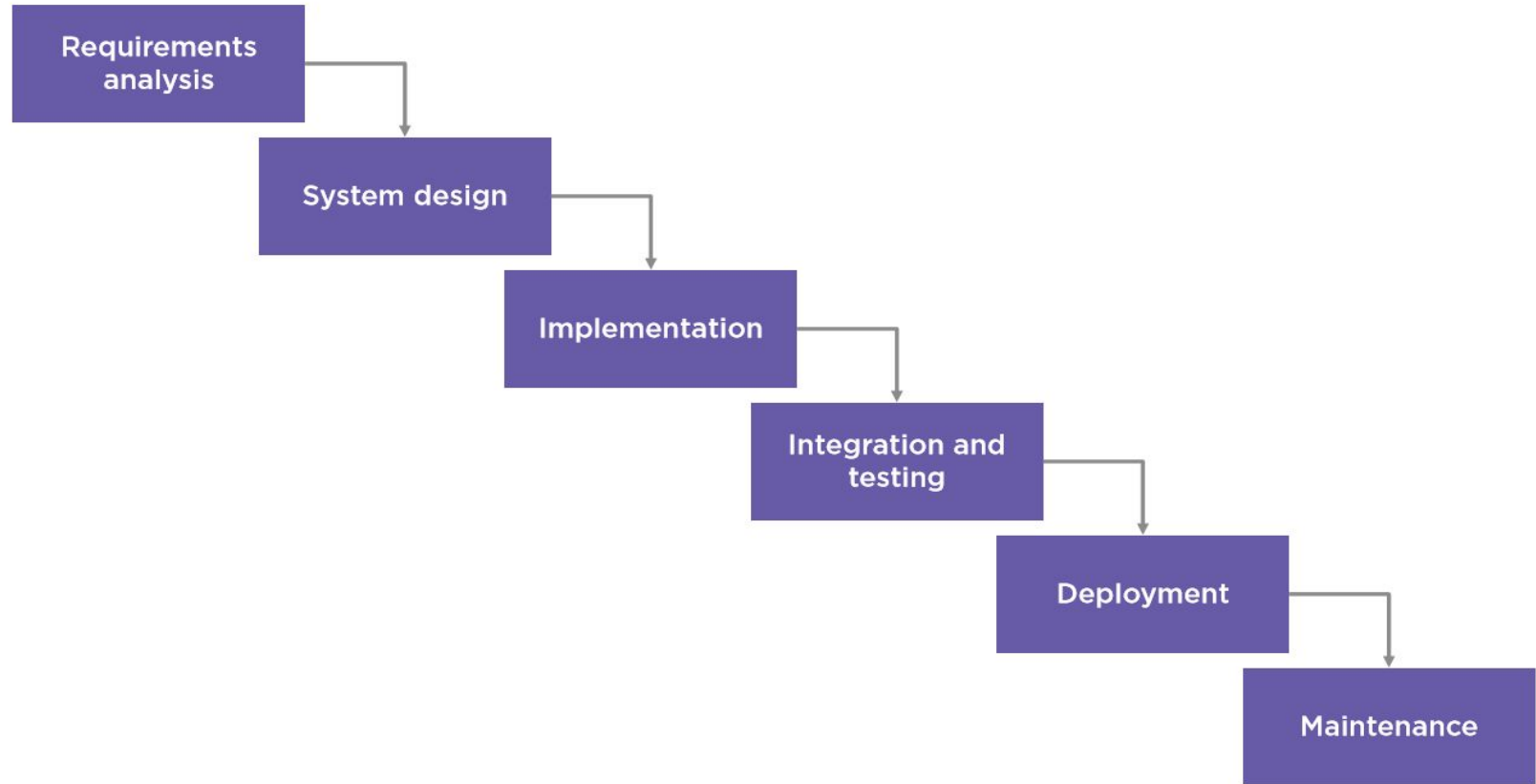
**Integration**

**Testing and debugging**

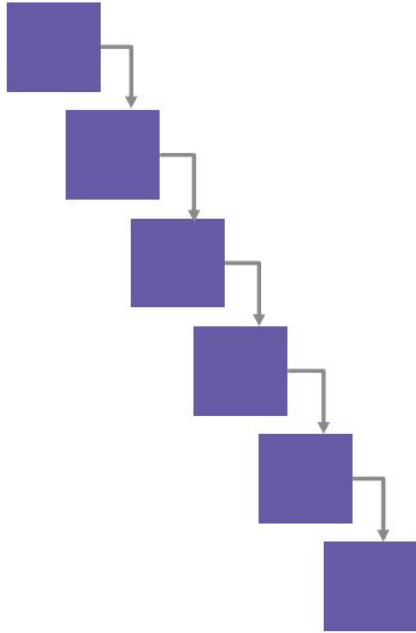
**Installation**

**Maintenance**

# How does Waterfall work?



# Where is Waterfall suitable?



**Requirements are well documented,  
clear and fixed**

**Product definition is stable**

**Technology is well understood**

**No ambiguous requirements**

**The project is short**

**Suitable resources available**

# History of the V-Model



**Modified version of the waterfall model**

**Designed to be non-linear**

**Described by Paul E. Brook in 1986**

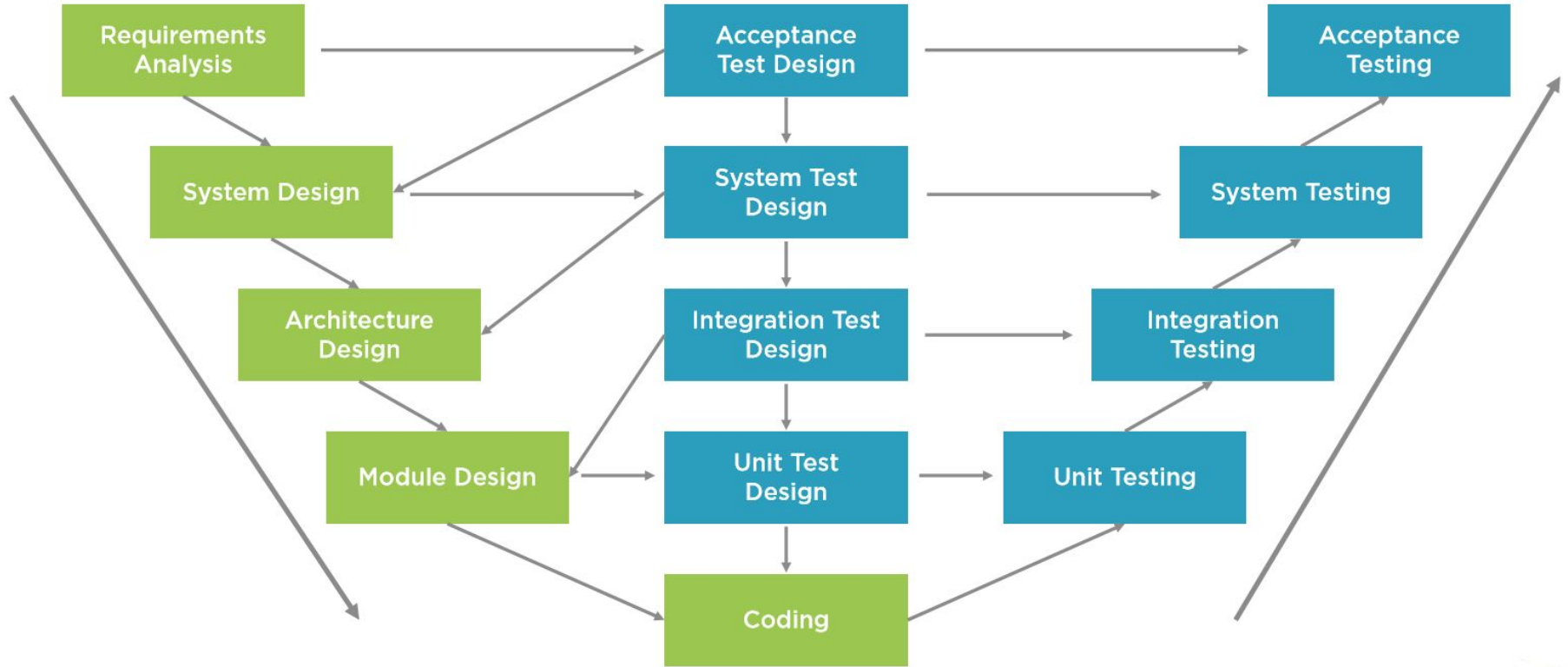
**Testing phase for each corresponding development stage**

# How Does the V-Model Work?

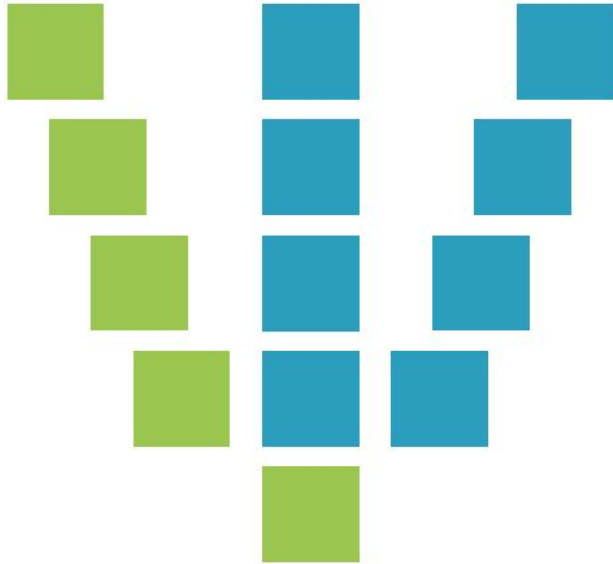
**Verification phases on  
one side**

**Validation phases on the  
other side**

# How does the V-model work?



# Where is the V-model suitable?



**Requirements are well defined,  
documented and fixed in place**

**Product definition is stable**

**Technology is well understood**

**No ambiguous requirements**

**The project is short**



# What is Agile?



**Software evolves over time**

**Adaptive planning, evolutionary development, early delivery**

**Deliver value to the business sooner**

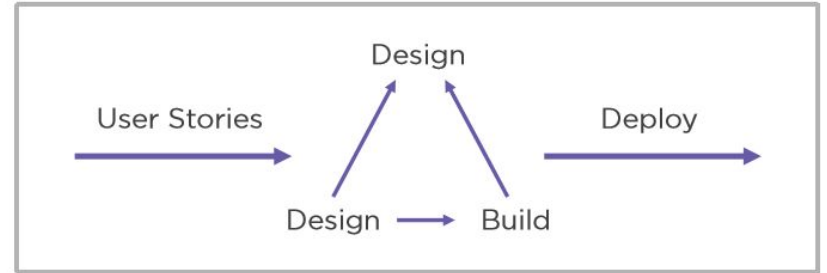
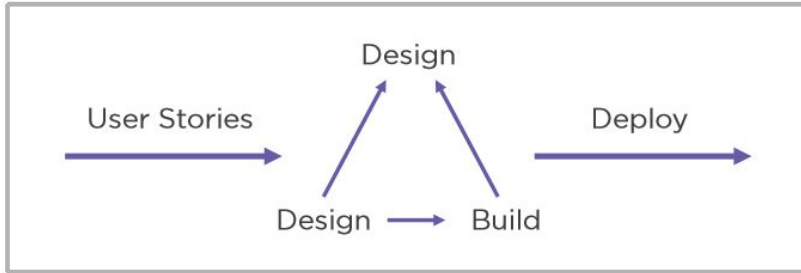
**Flexible response to change**

# What is Agile?

## Traditional Waterfall



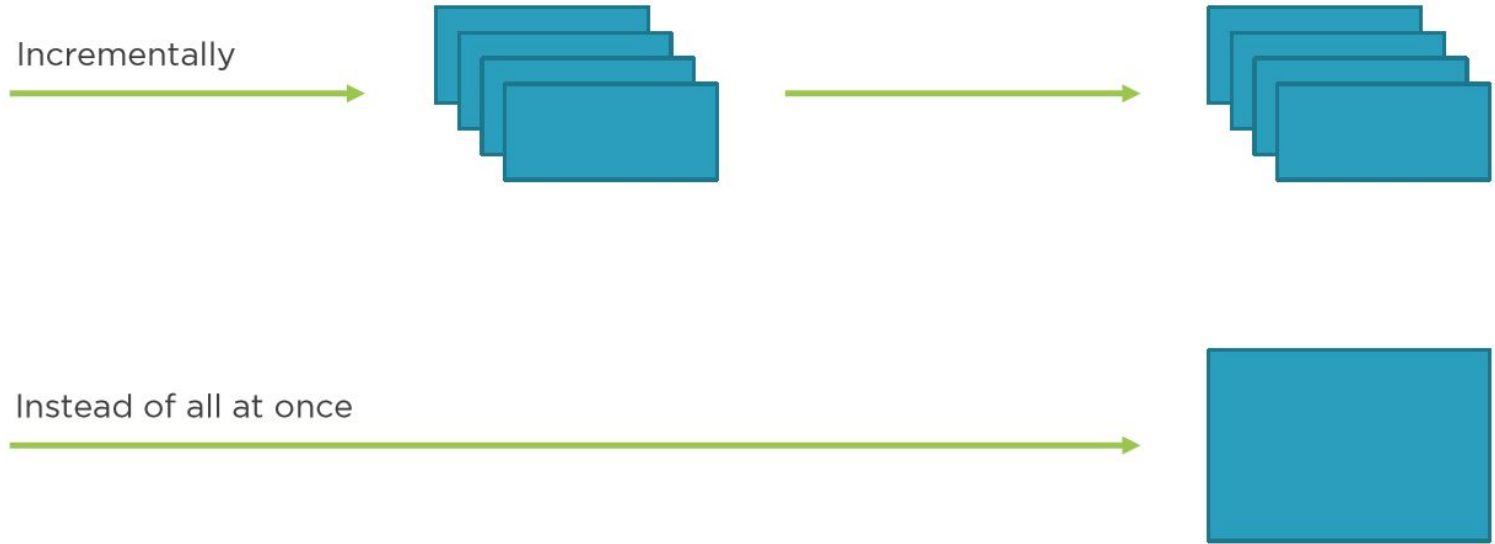
## Agile Development



## Project Timeline



# What is Agile?



# Agile Manifesto

17 people met to find  
common ground about  
software development

They formed “The Agile  
Alliance” and signed the  
Agile manifesto.

[agilemanifesto.org](https://agilemanifesto.org)

# Agile Manifesto



**We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:**

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

**That is, while there is value in the items on the right, we value the items on the left more**

# Agile Manifesto Principles

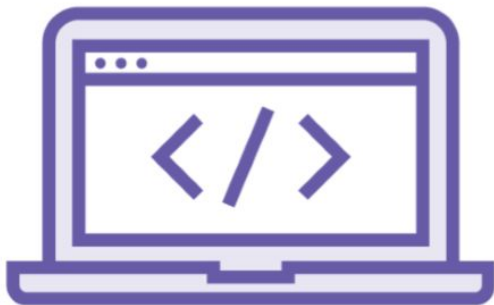


**12 principles behind the agile manifesto**

**Split into 3 groups:**

- Regular delivery of software
- Team communication
- Excellence in design

# Regular Delivery of Software



The highest priority is to satisfy the customer through early and continuous delivery of valuable software

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

Working software is the primary measure of progress

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely

# Team Communication

Business people and developers must work together daily throughout the project

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

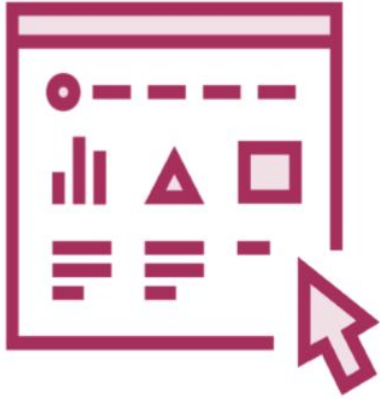
The best architectures, requirements, and designs emerge from self-organizing teams

Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly



# Excellence in Design



Continuous attention  
to technical excellence  
and good design  
enhances agility



Simplicity, the art of  
maximizing the amount  
of work not done,  
is essential



Agile processes  
harness change for  
the customer's  
competitive advantage

# Agile Methodologies Overview



Scrum

Extreme programming (XP)

Crystal

Dynamic systems development method (DSDM)

Feature driven design (FDD)

Lean software development

Kanban

# Roles in an Agile Team

Agile teams are software development teams first and members of a department second

A team should have a product or domain expert

A team has members with cross functional skills

A team should have some form of leadership role

A team can benefit from an agile coach or mentor



**Good luck!**