Waterfall Development and Its Problems

Stephen Haunts

@stephenhaunts | www.stephenhaunts.com

History of the Waterfall Model

- First introduced by Winston Royce

http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

History of the Waterfall Model

- First introduced by Winston Royce
- Was not referred to as Waterfall until later

History of the Waterfall Model

- Requirements specification

History of the Waterfall Model

- Requirements specification
- Design

History of the Waterfall Model

- Requirements specification
- Design
- Construction

History of the Waterfall Model

- Requirements specification
- Design
- Construction
- Integration

History of the Waterfall Model

- Requirements specification
- Design
- Construction
- Integration
- Testing and debugging

History of the Waterfall Model

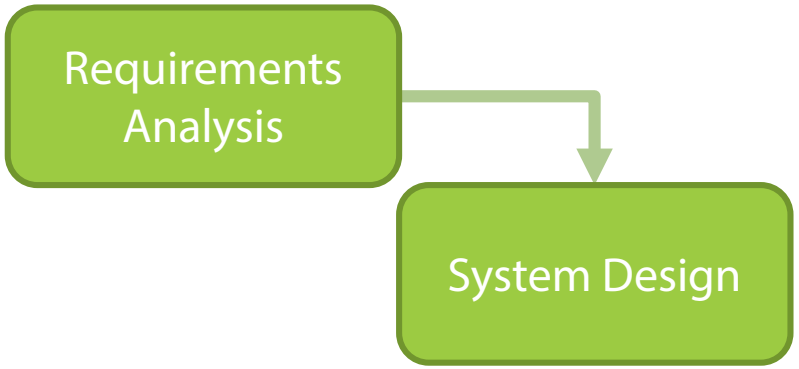
- Installation

History of the Waterfall Model

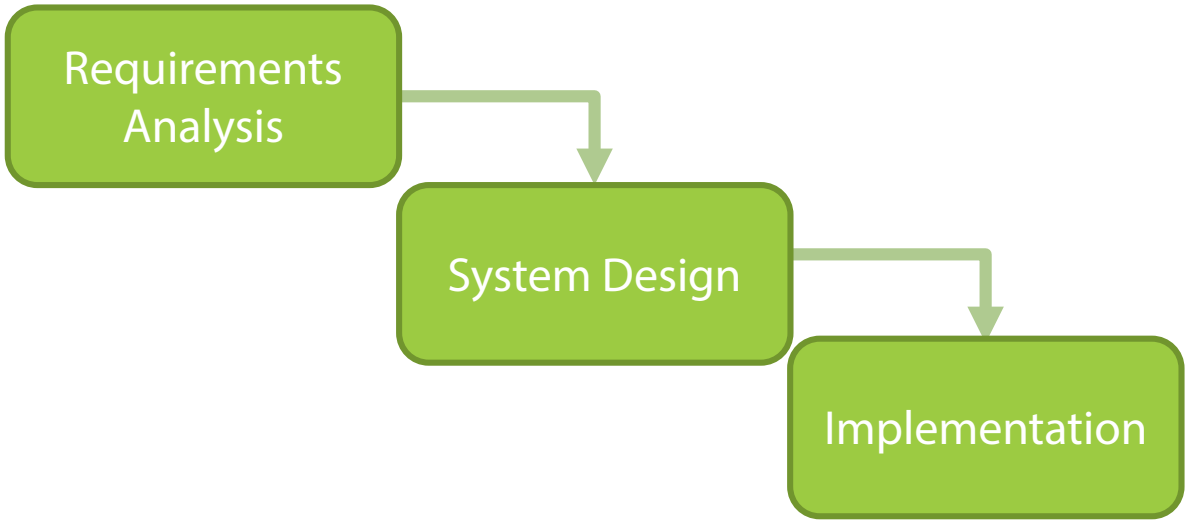
- Installation
- Maintenance

How Does Waterfall Work?

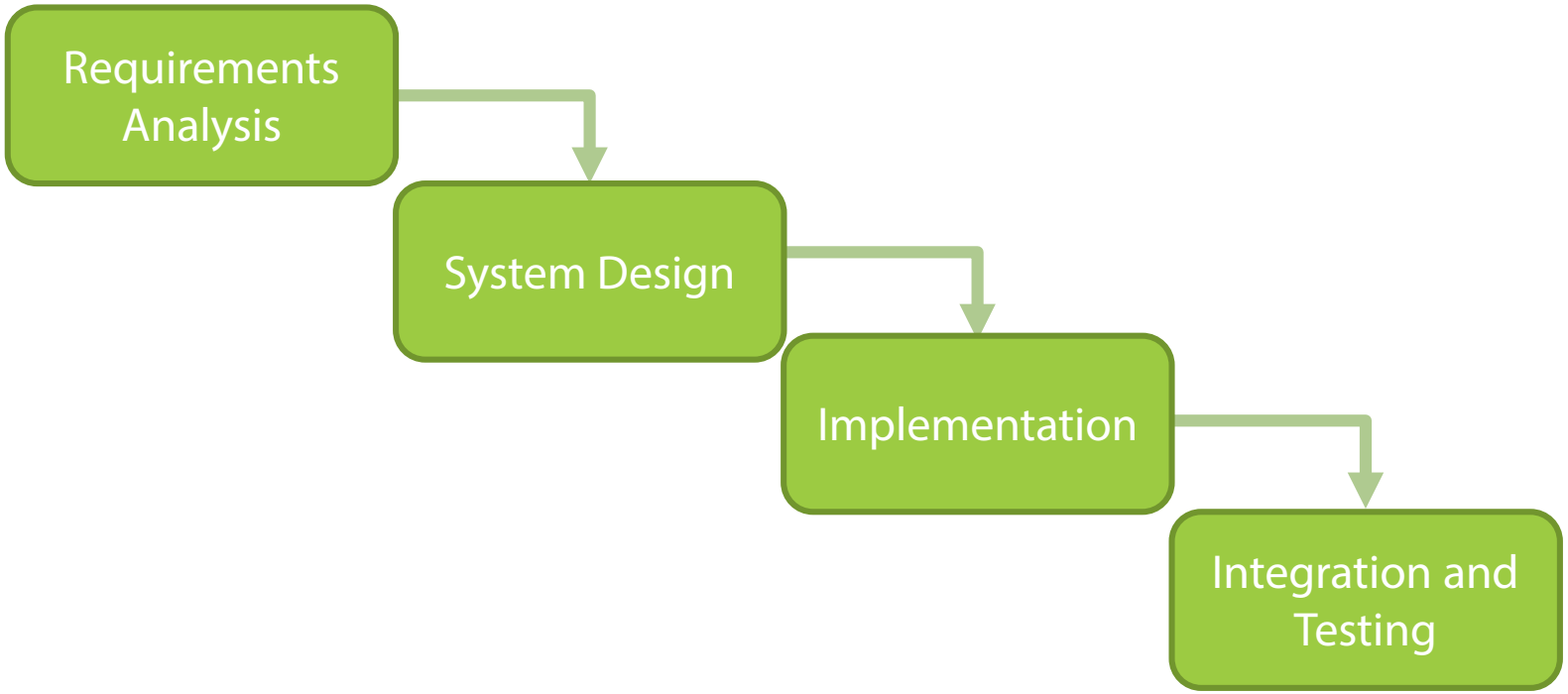
How Does Waterfall Work?



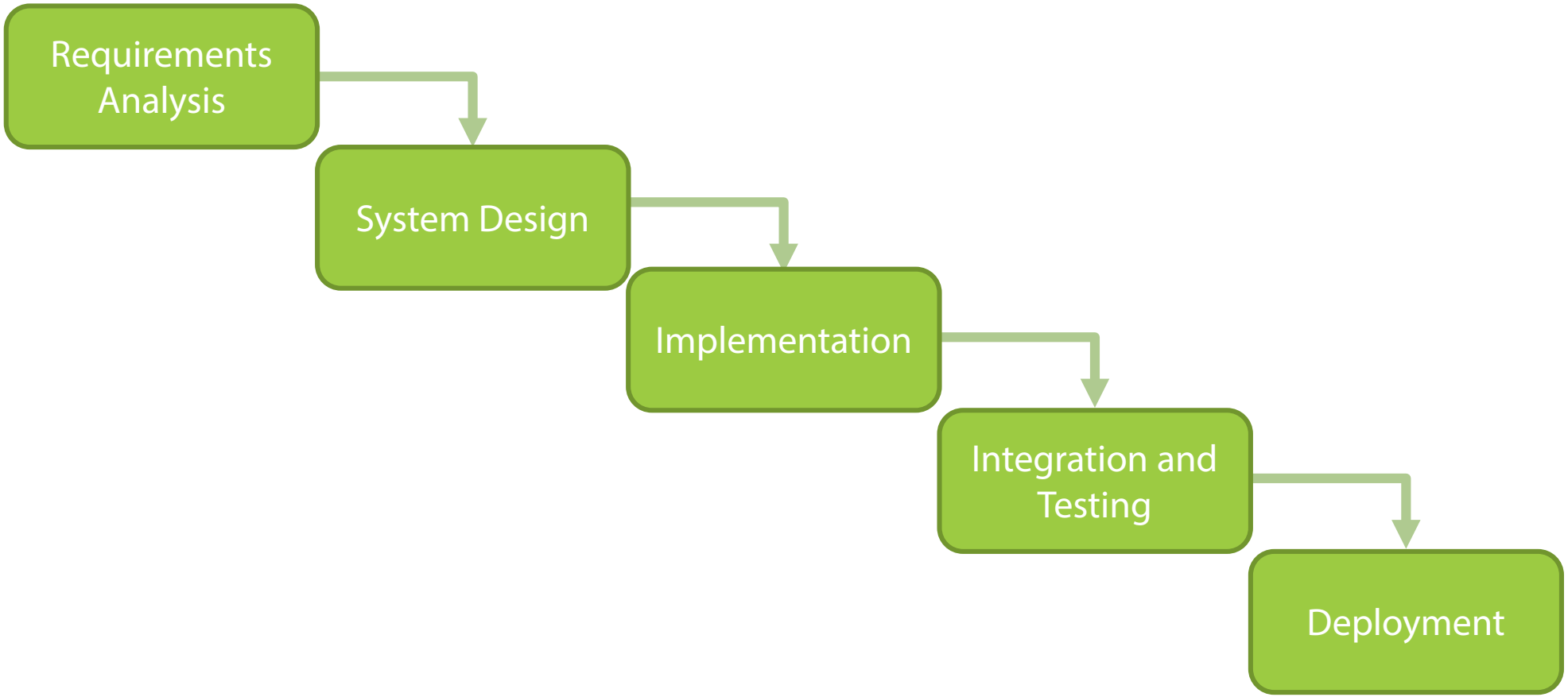
How Does Waterfall Work?



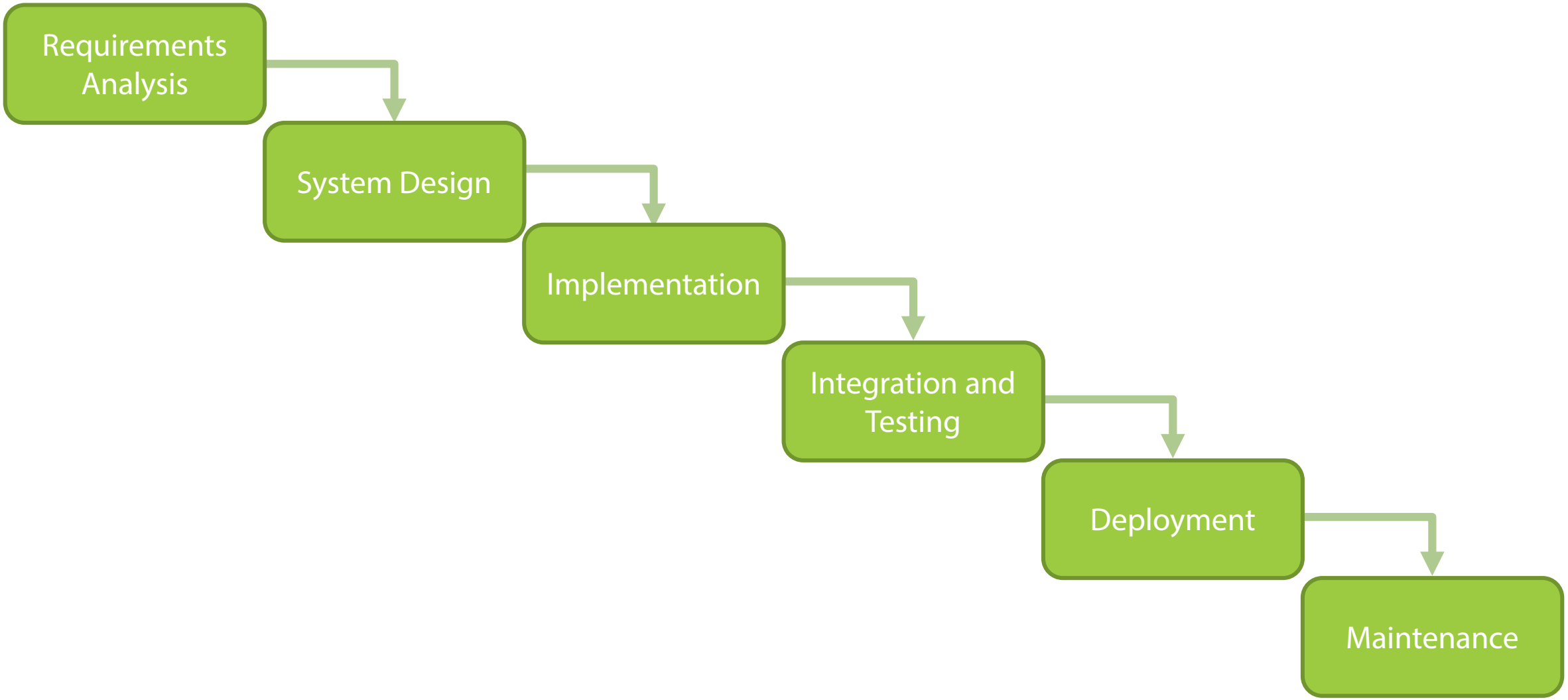
How Does Waterfall Work?



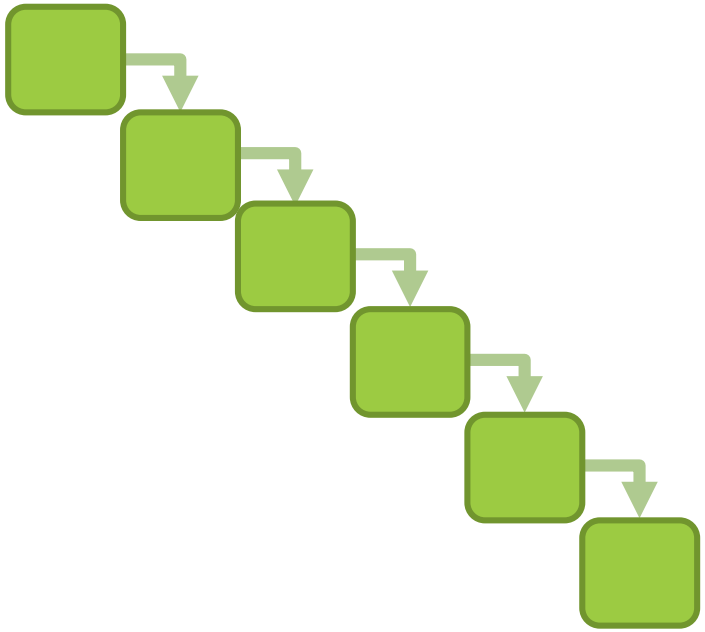
How Does Waterfall Work?



How Does Waterfall Work?

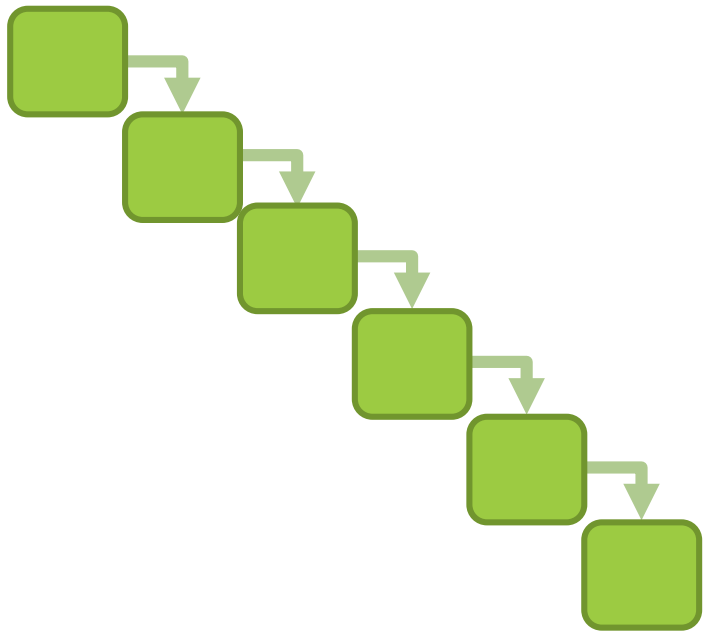


Where Is Waterfall Suitable?



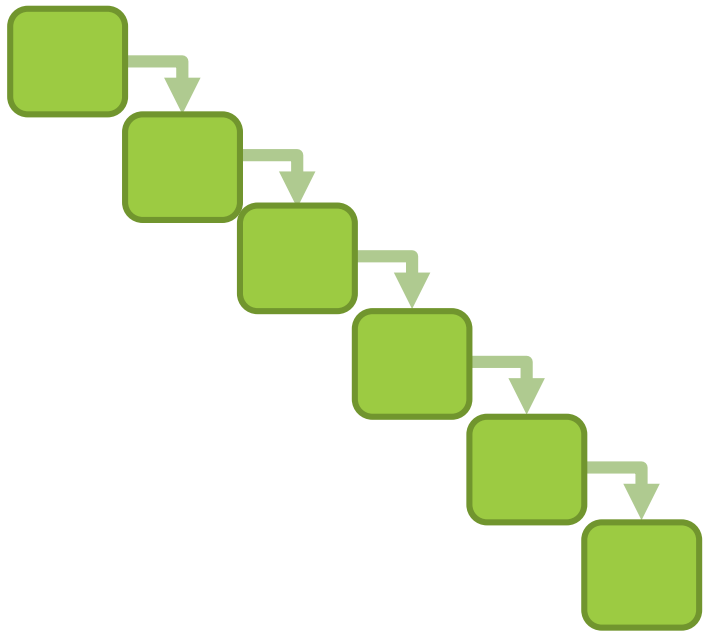
- Requirements are well documented, clear and fixed

Where Is Waterfall Suitable?



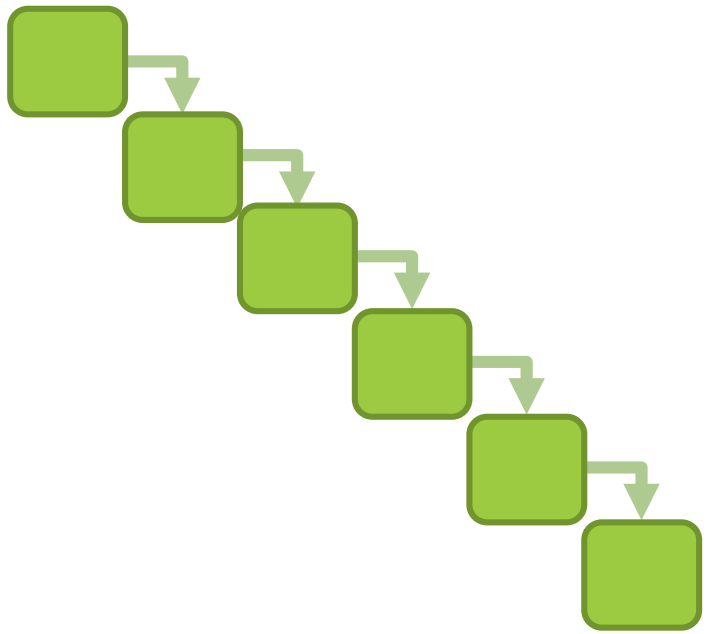
- Requirements are well documented, clear and fixed
- Product definition is stable

Where Is Waterfall Suitable?



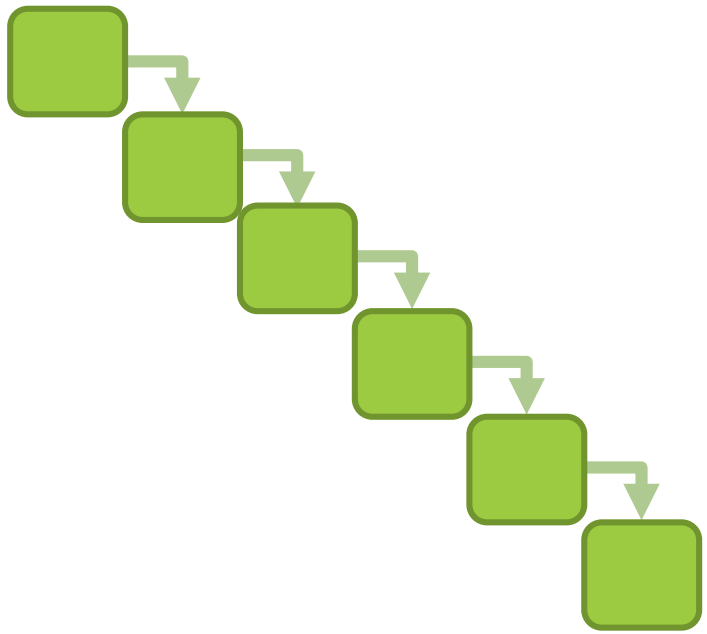
- Requirements are well documented, clear and fixed
- Product definition is stable
- Technology is well understood

Where Is Waterfall Suitable?



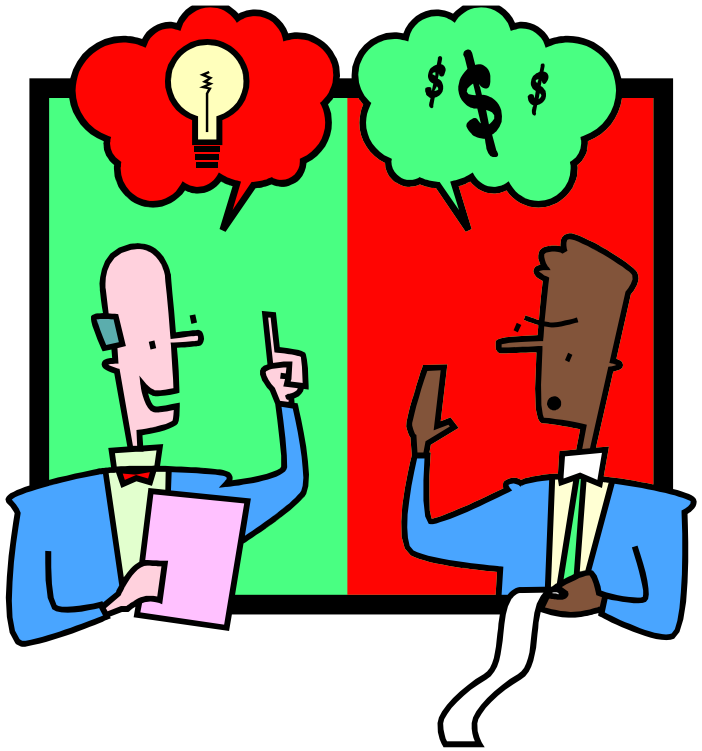
- Requirements are well documented, clear and fixed
- Product definition is stable
- Technology is well understood
- No ambiguous requirements
- The project is short

Where Is Waterfall Suitable?



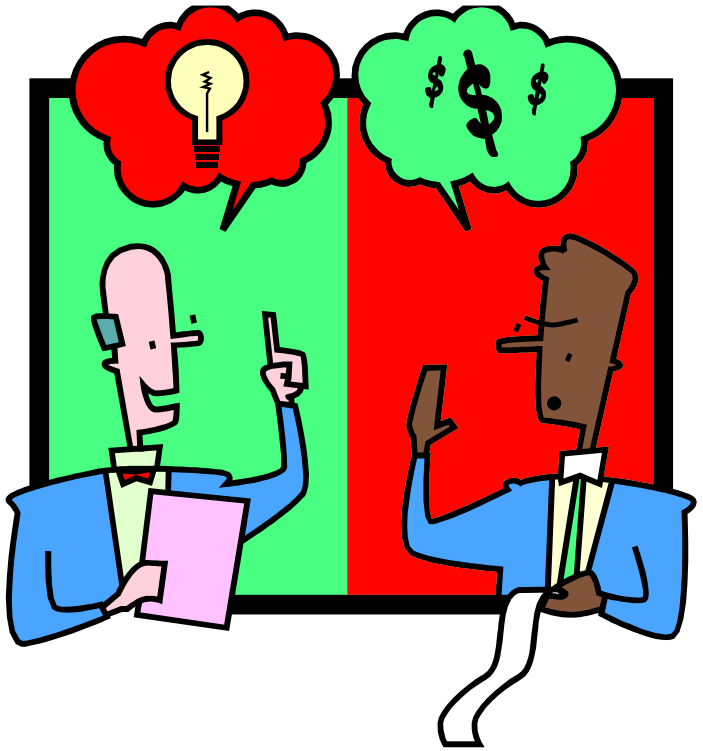
- Requirements are well documented, clear and fixed
- Product definition is stable
- Technology is well understood
- No ambiguous requirements
- The project is short
- Suitable resources available

Advantages and Disadvantages



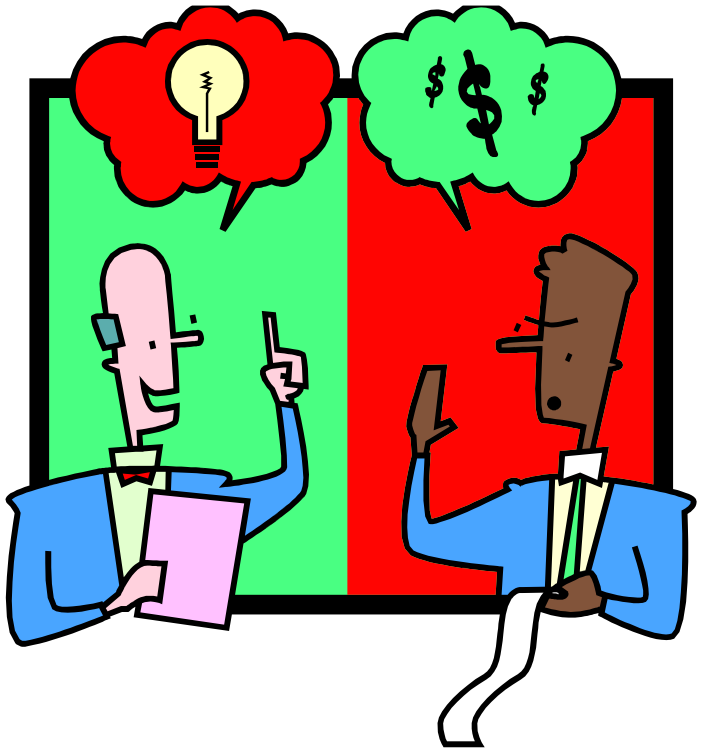
- Advantages

Advantages and Disadvantages



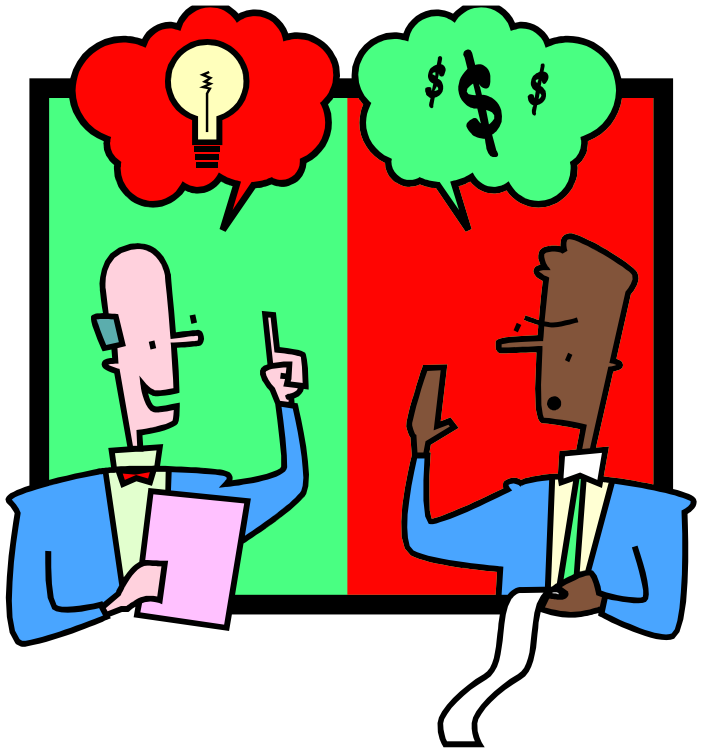
- Advantages
  - Easier scheduling and control

Advantages and Disadvantages



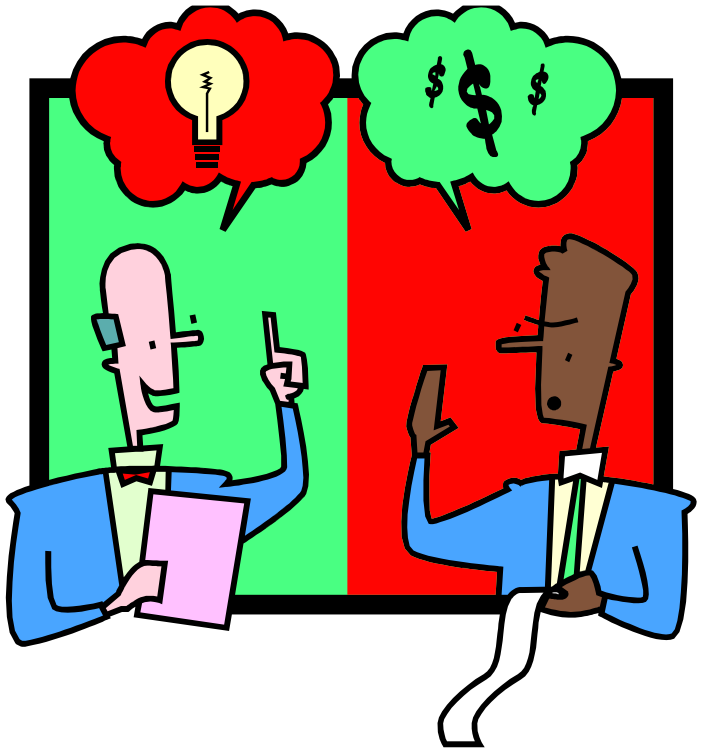
- Advantages
  - Easier scheduling and control
  - Departmentalization

Advantages and Disadvantages



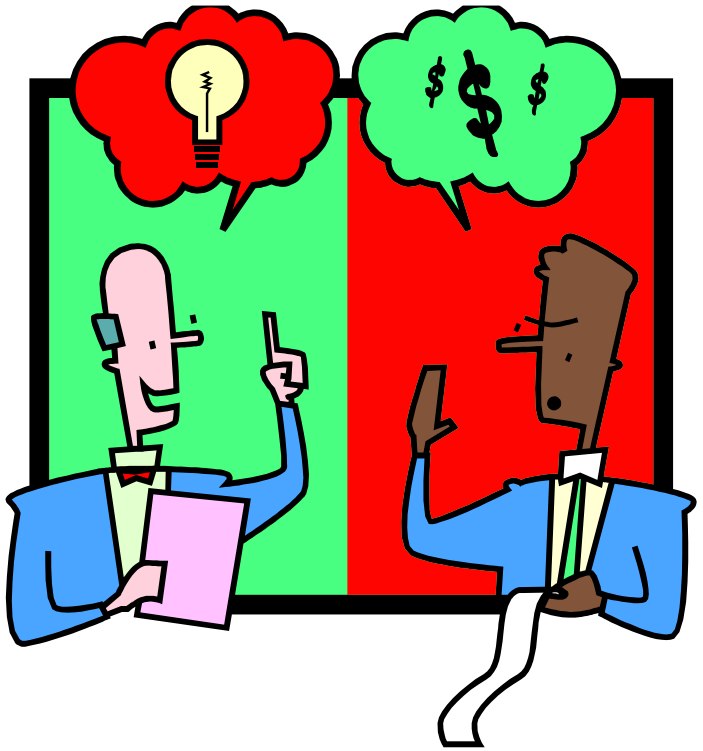
- Disadvantages

Advantages and Disadvantages



- Disadvantages
  - Does not allow for reflection or revision

Advantages and Disadvantages



- Disadvantages
  - Does not allow for reflection or revision
  - Once in testing stage, change is hard

Pros

Pros

- Simple and easy to understand

Pros

- Simple and easy to understand
- Easy to manage

Pros

- Simple and easy to understand
- Easy to manage
- Phases are completed one at a time

Pros

- Simple and easy to understand
- Easy to manage
- Phases are completed one at a time
- Works well for smaller projects

Pros

- Simple and easy to understand
- Easy to manage
- Phases are completed one at a time
- Works well for smaller projects
- Clearly defined stages

Pros

- Simple and easy to understand
- Easy to manage
- Phases are completed one at a time
- Works well for smaller projects
- Clearly defined stages
- Well understood milestones

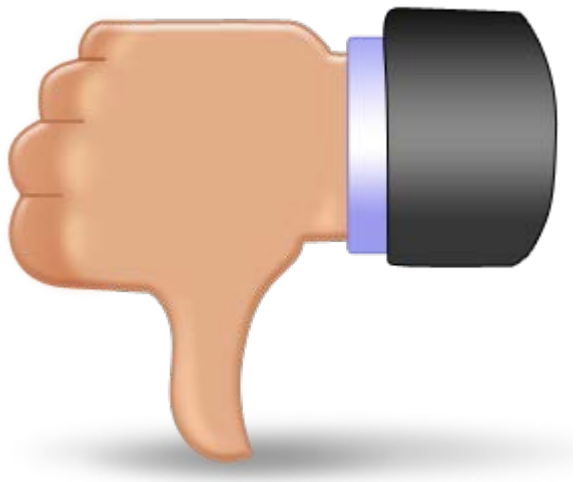
Pros

- Process and results are well documented

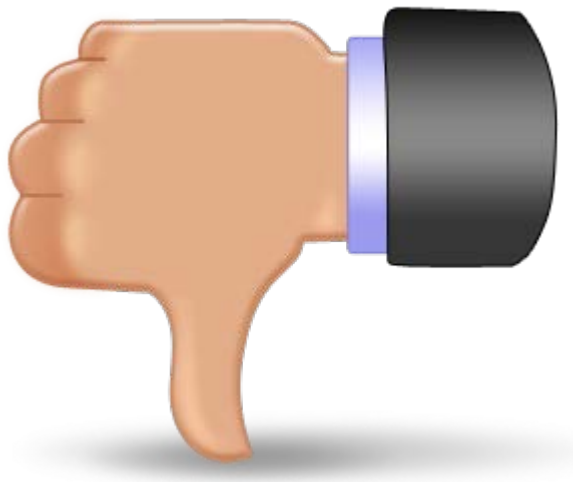
Pros

- Process and results are well documented
- Tasks are easy to arrange for a project manager

Cons

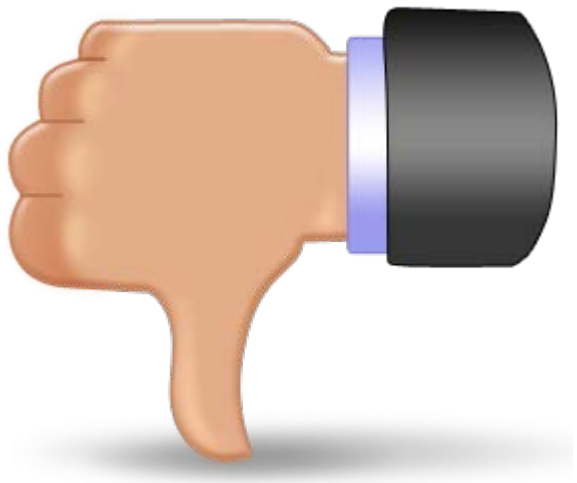


Cons



- No working software until late in the cycle

Cons

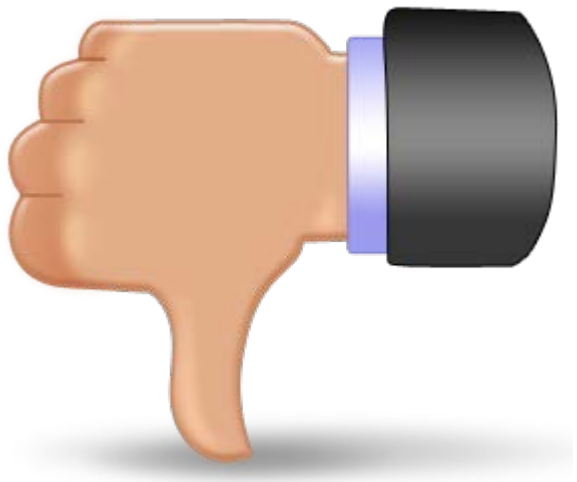


- No working software until late in the cycle
- High amounts of risk and uncertainty

Cons

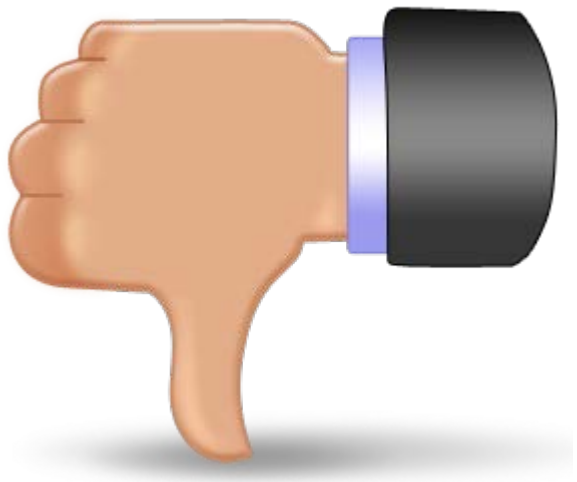
- No working software until late in the cycle
- High amounts of risk and uncertainty
- Not good for complex projects

Cons



- No working software until late in the cycle
- High amounts of risk and uncertainty
- Not good for complex projects
- Not good where change is expected

Cons



- Change in scope can end a project

Cons

- Change in scope can end a project
- Integration and delivery is done as a "Big Bang"

History of the V-Model

- Modified version of the waterfall model

History of the V-Model

- Modified version of the waterfall model
- Designed to be non-linear

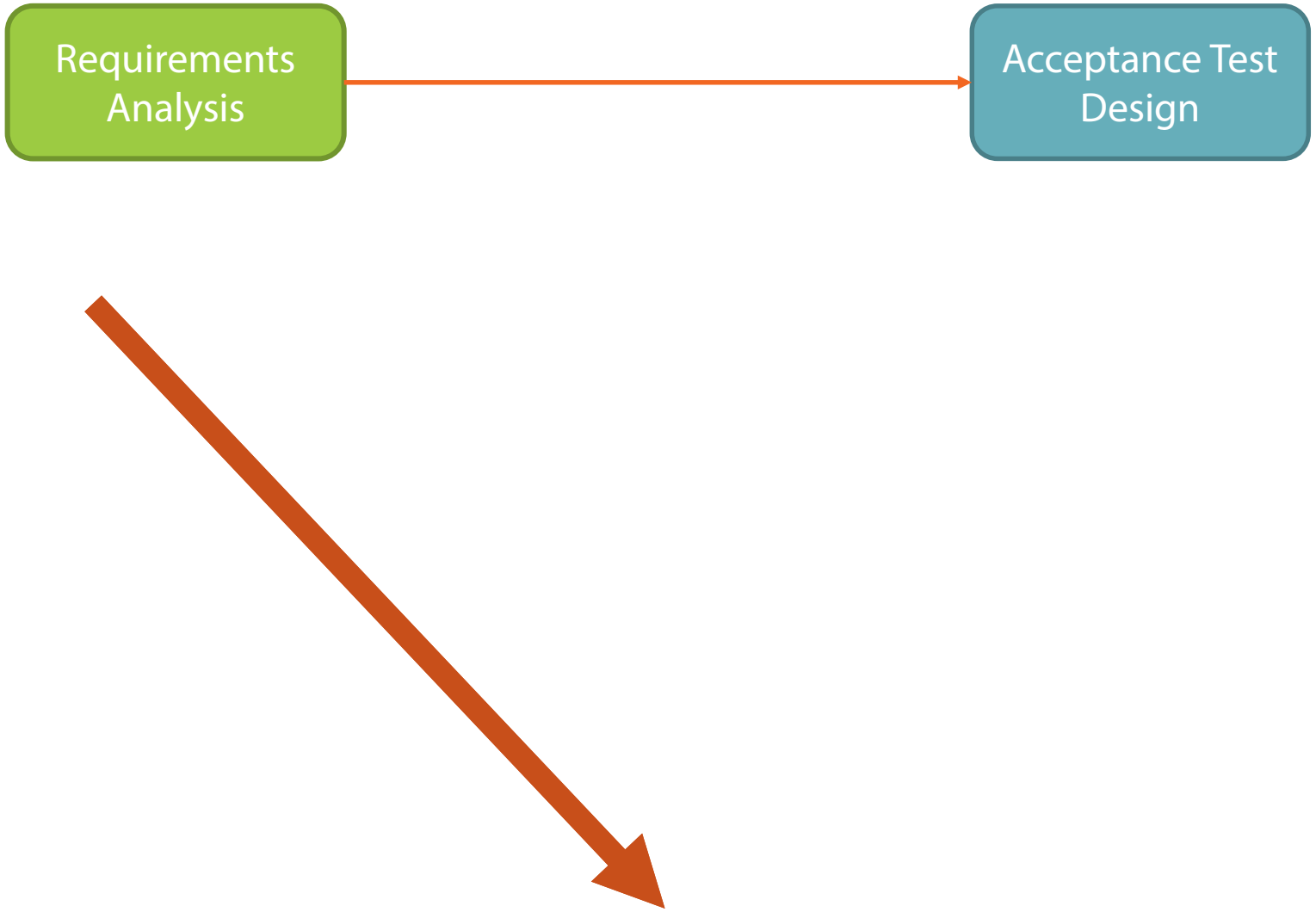
History of the V-Model

- 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

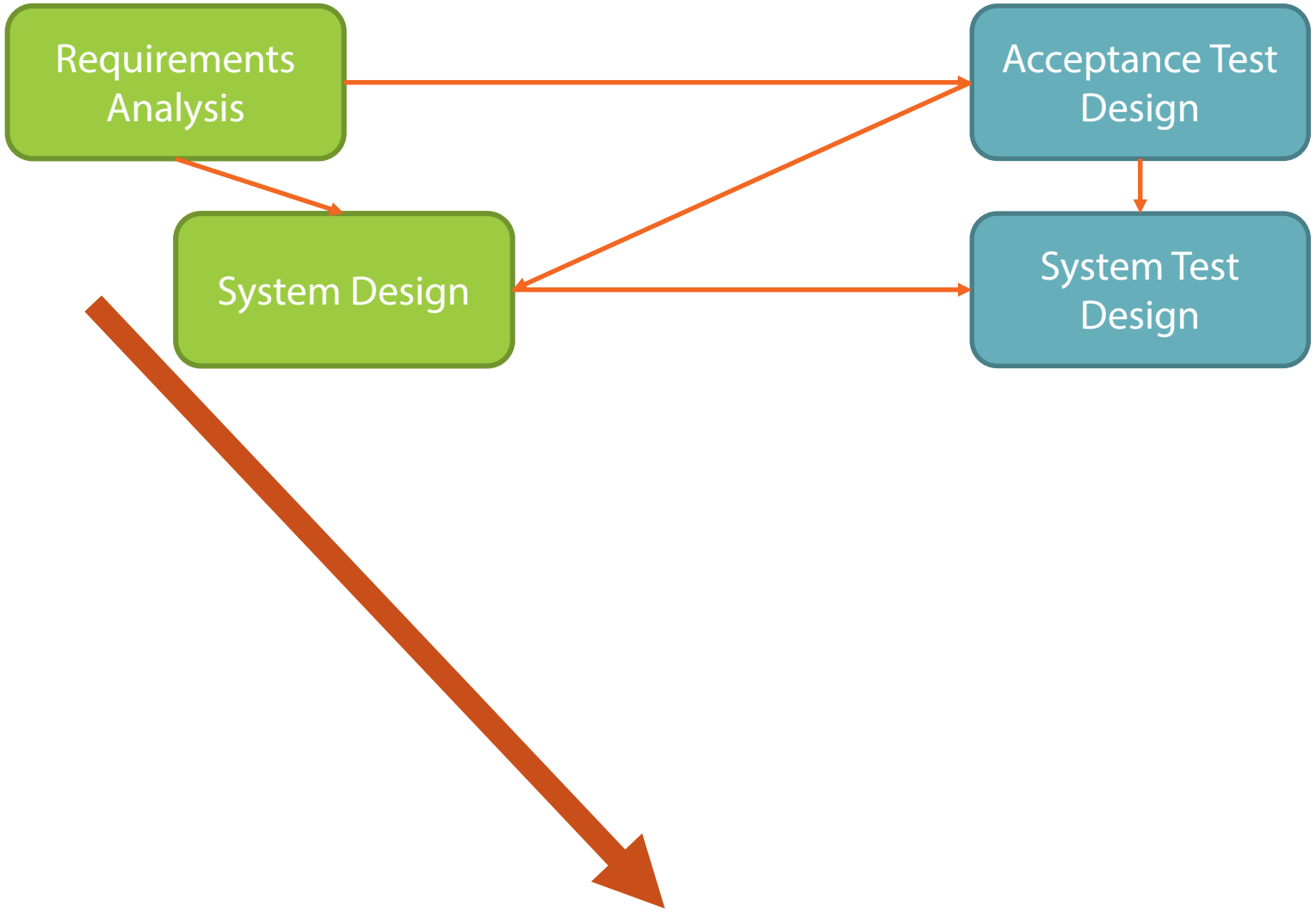
How Does the V-Model Work?

- Verification phases on one side and Validation phases on the other side

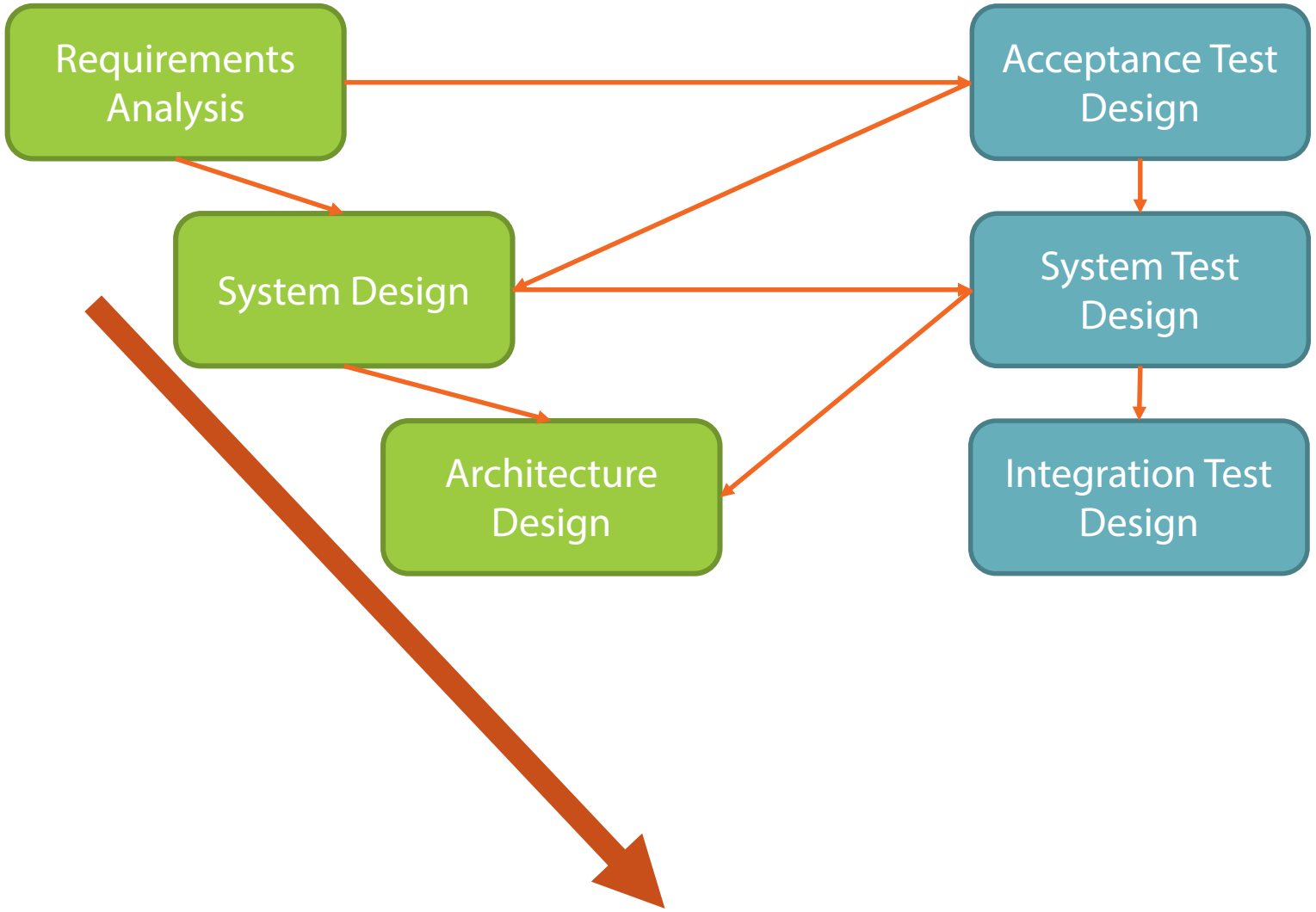
How Does the V-Model Work?



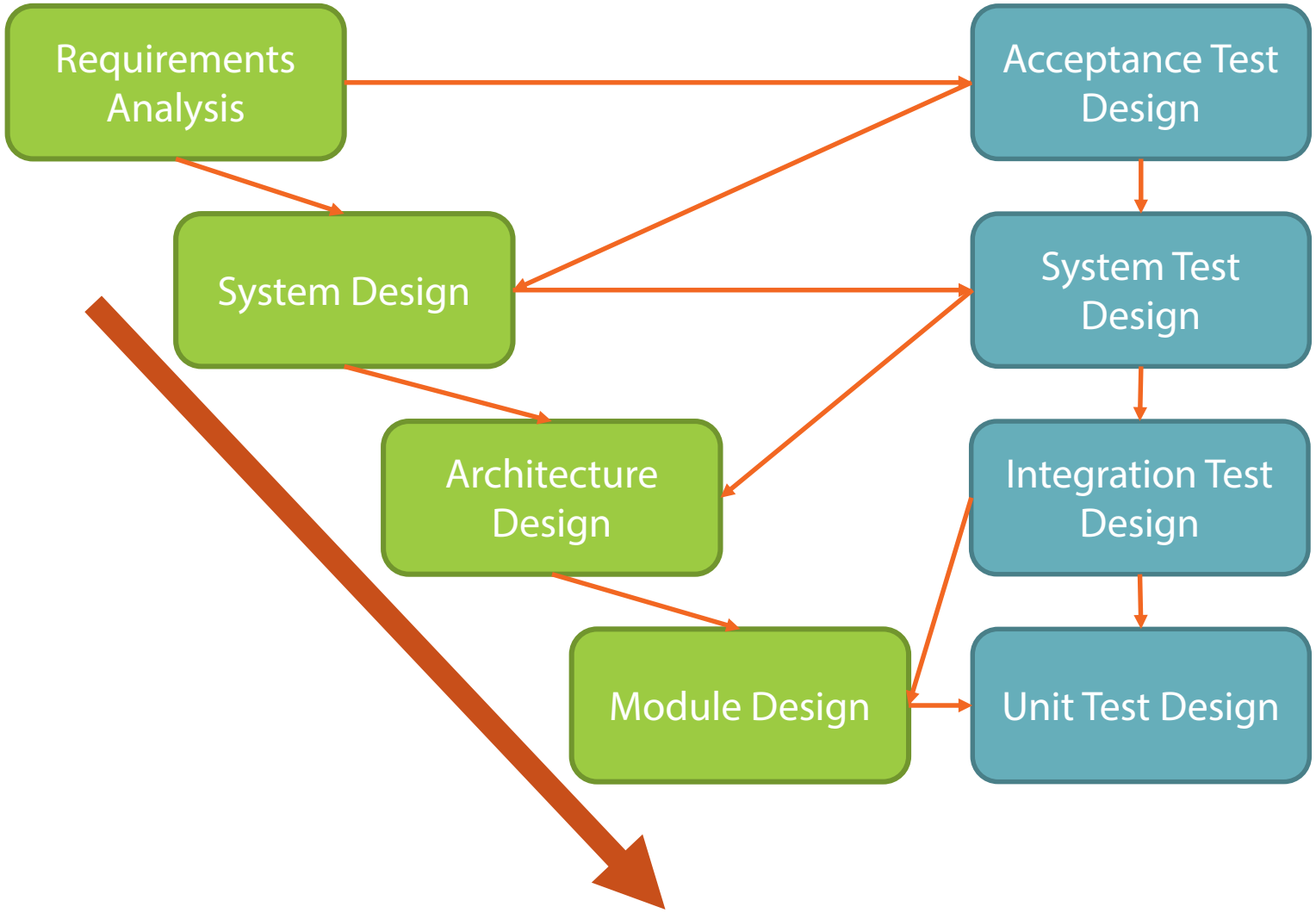
How Does the V-Model Work?



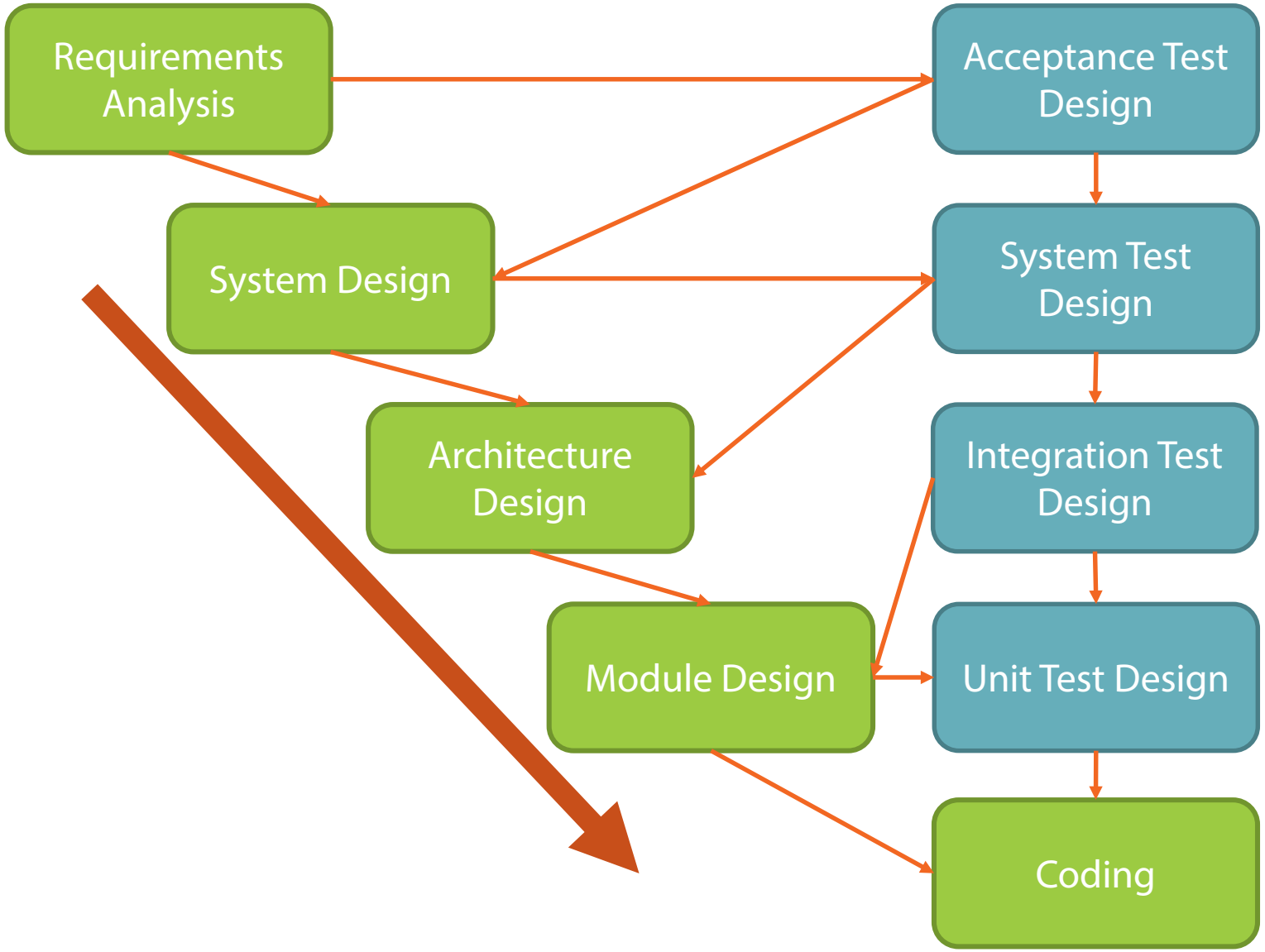
How Does the V-Model Work?



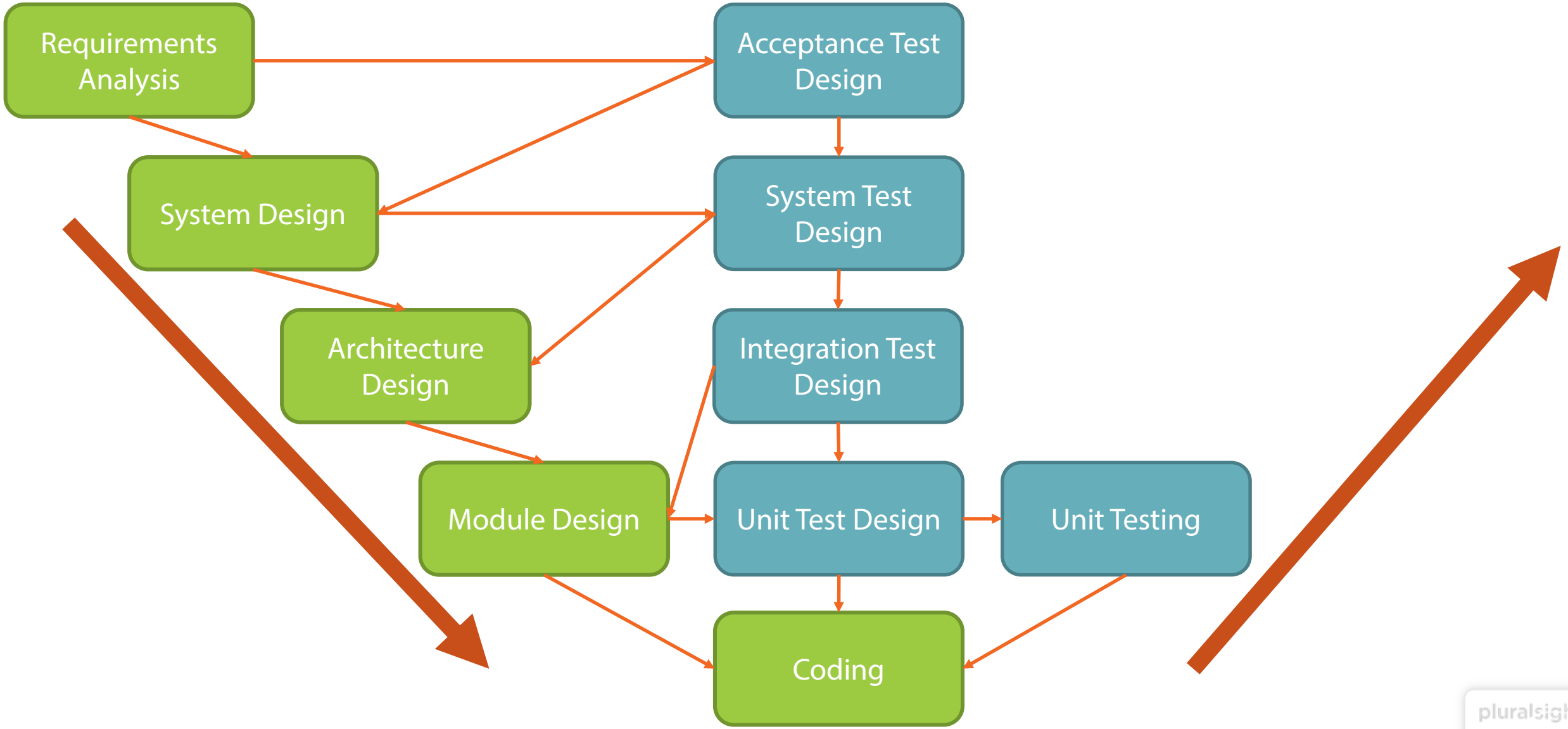
How Does the V-Model Work?



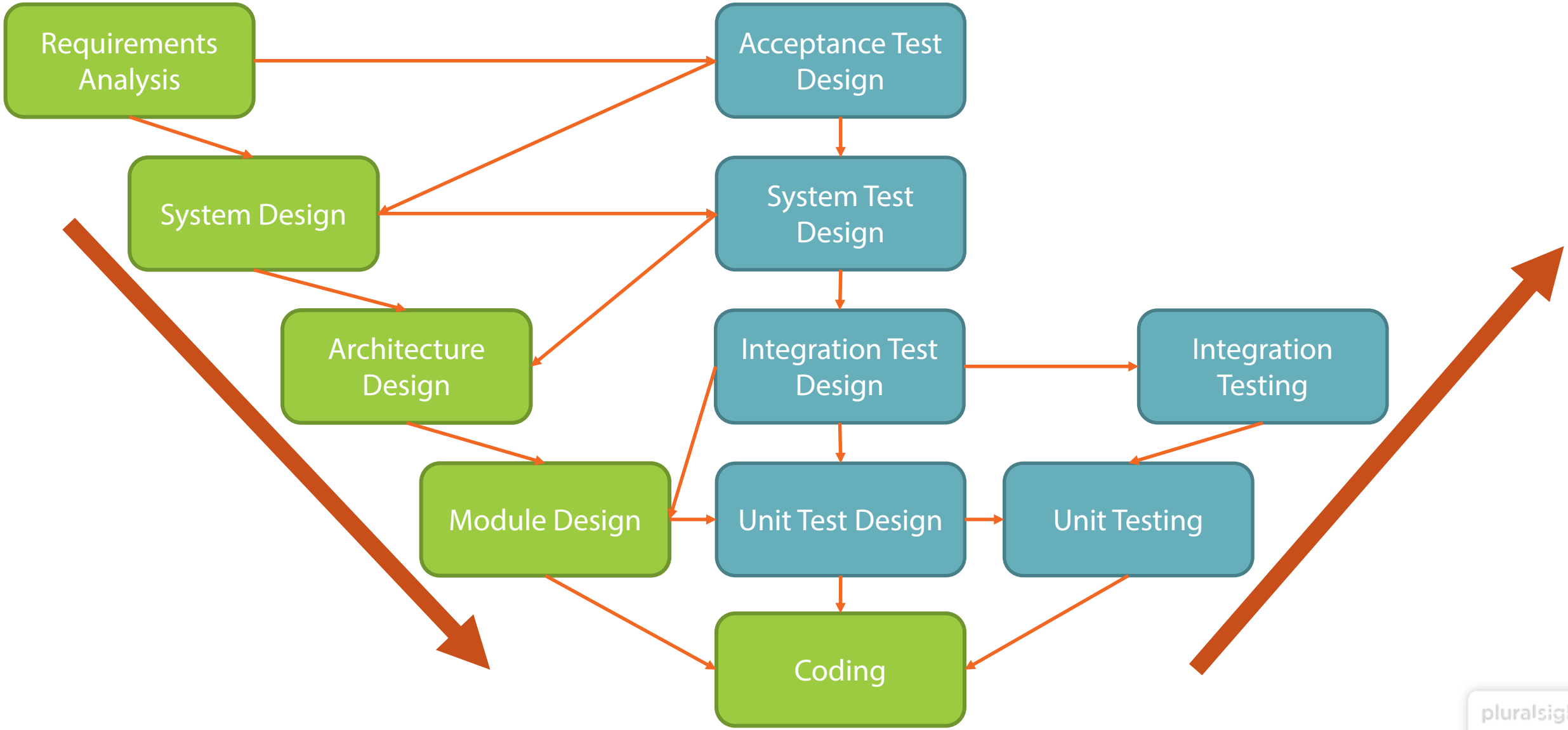
How Does the V-Model Work?



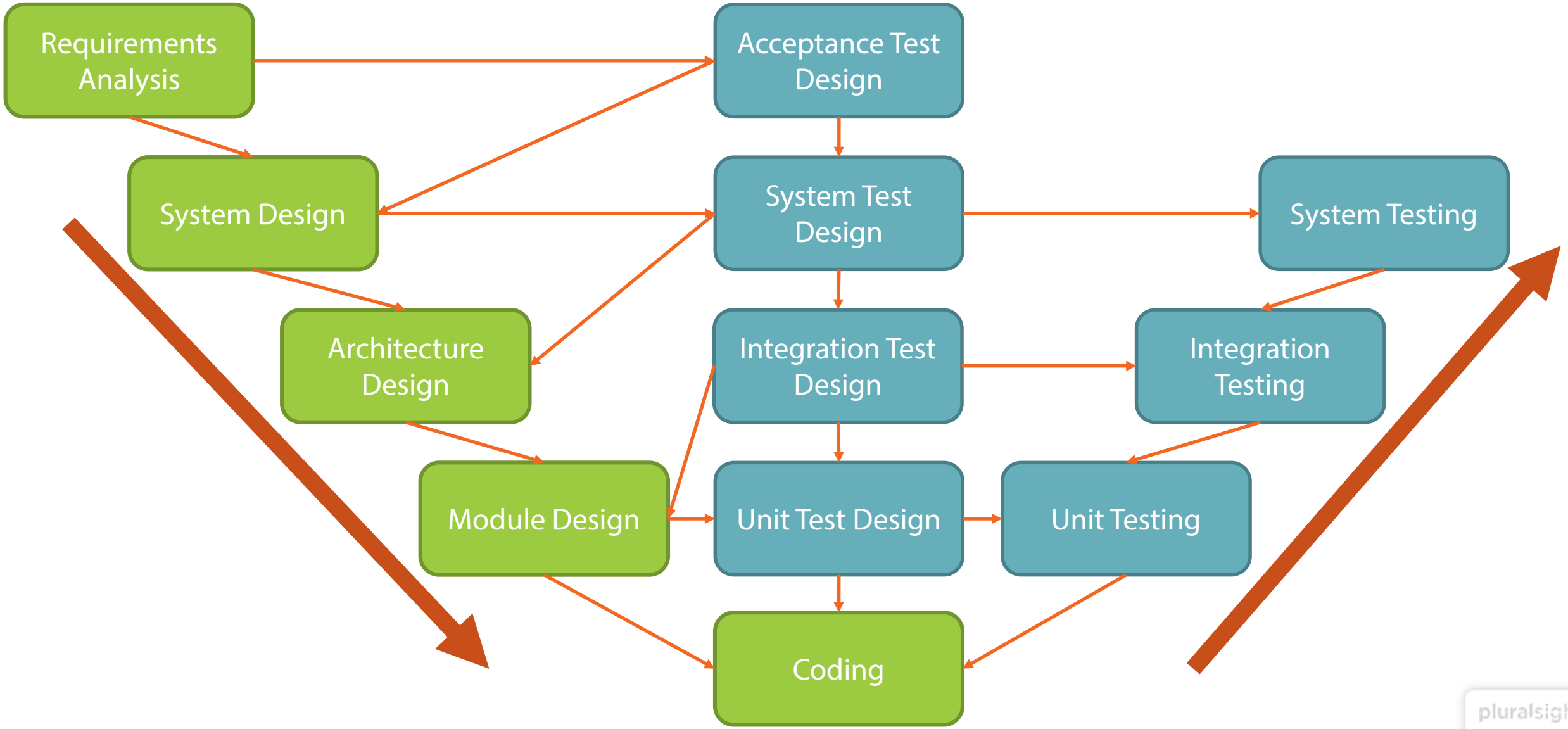
How Does the V-Model Work?



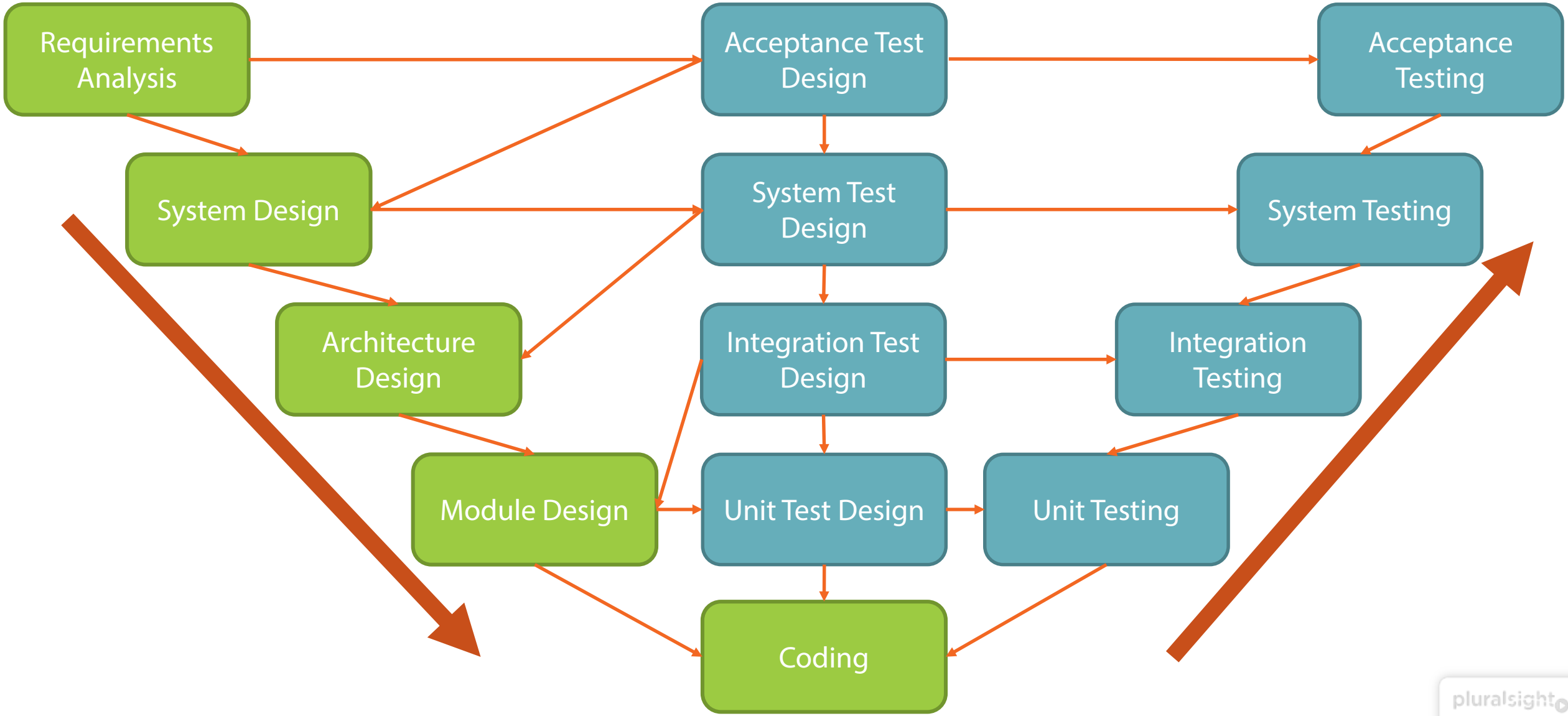
How Does the V-Model Work?



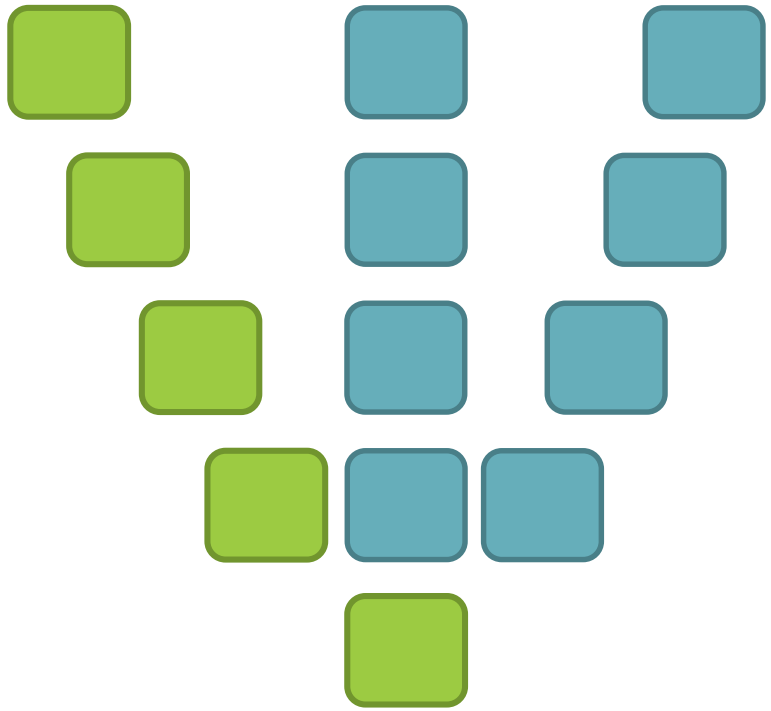
How Does the V-Model Work?



How Does the V-Model Work?

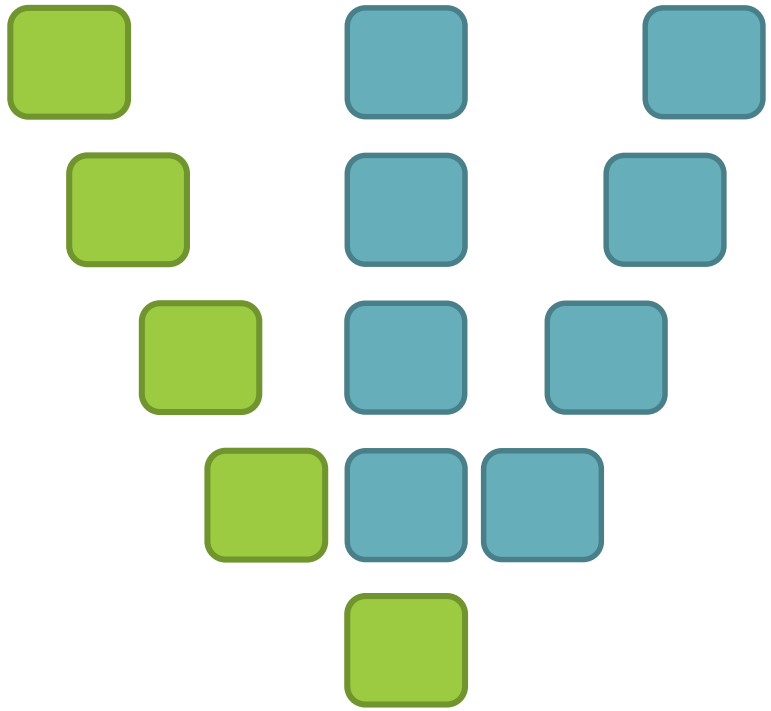


Where Is the V-Model Suitable?



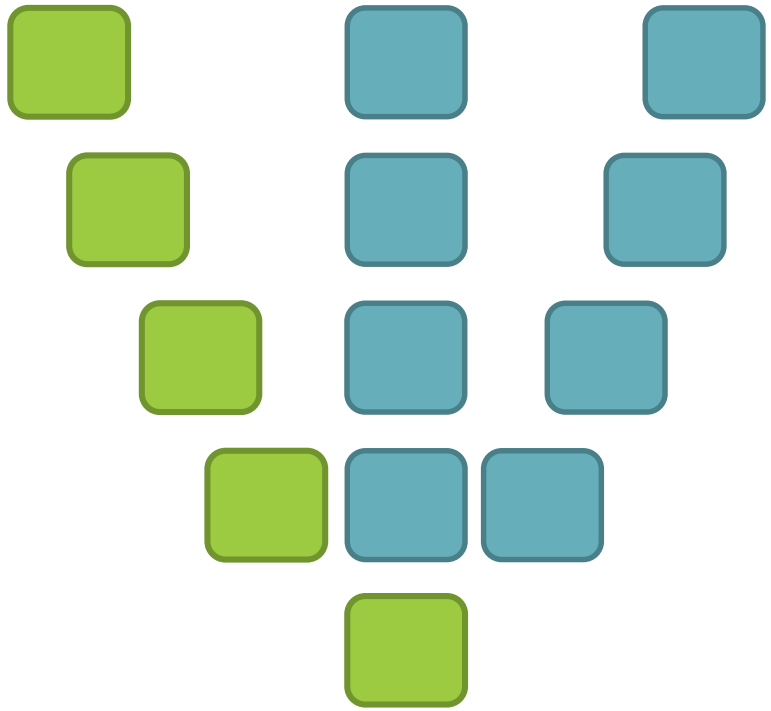
- Requirements are well defined, documented and fixed in place

Where Is the V-Model Suitable?



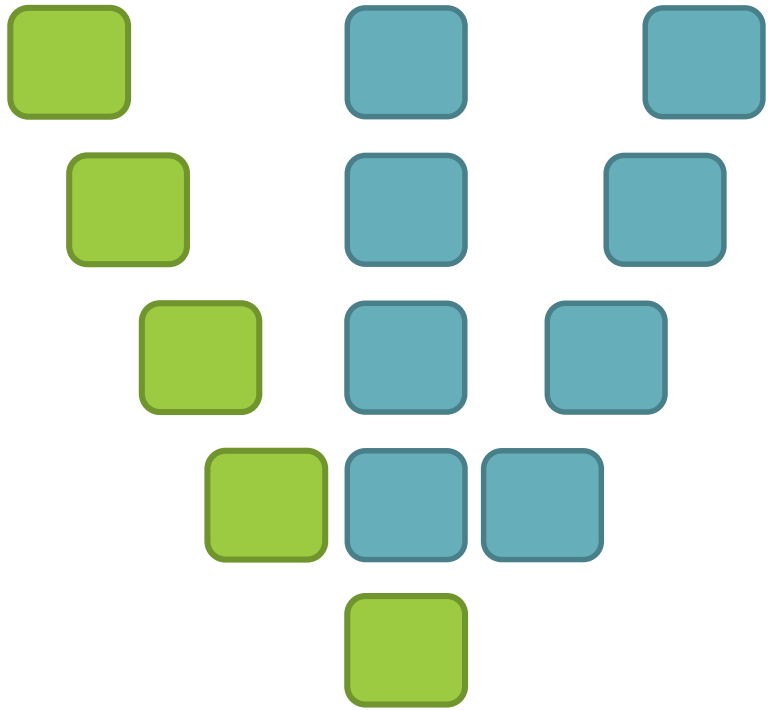
- Requirements are well defined, documented and fixed in place
- Product definition is stable

Where Is the V-Model Suitable?



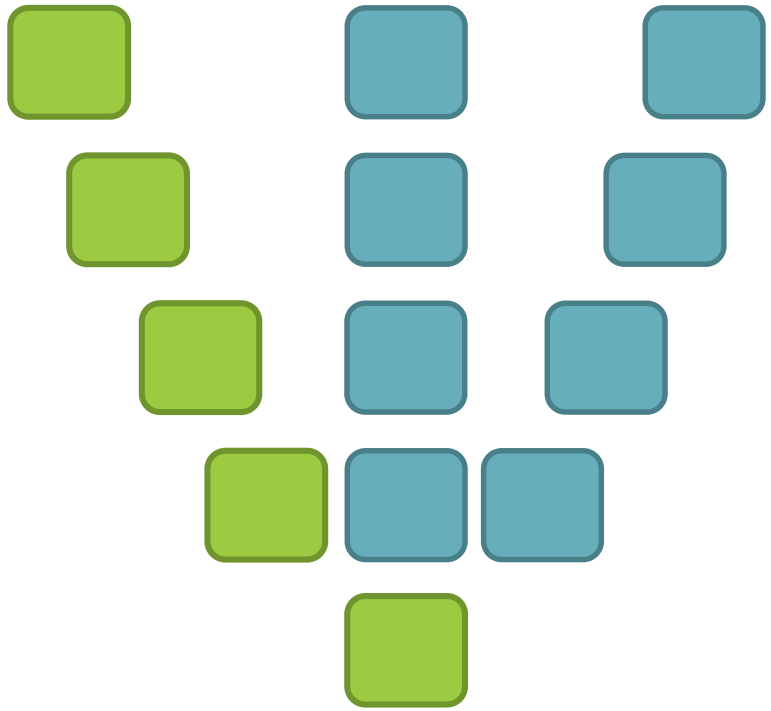
- Requirements are well defined, documented and fixed in place
- Product definition is stable
- Technology is well understood

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

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

Advantages and Disadvantages

- Advantages
  - Easy to understand and apply

Advantages and Disadvantages

- Advantages
  - Easy to understand and apply
  - Easy to manage

Advantages and Disadvantages

- Disadvantages
  - Model is not flexible to changes in requirements

Advantages and Disadvantages

- Disadvantages
  - Model is not flexible to changes in requirements
  - Requirements changes are disruptive and expensive

Pros

Pros

- Disciplined model where phases are complete one at a time

Pros

- Disciplined model where phases are complete one at a time
- Works well for small projects

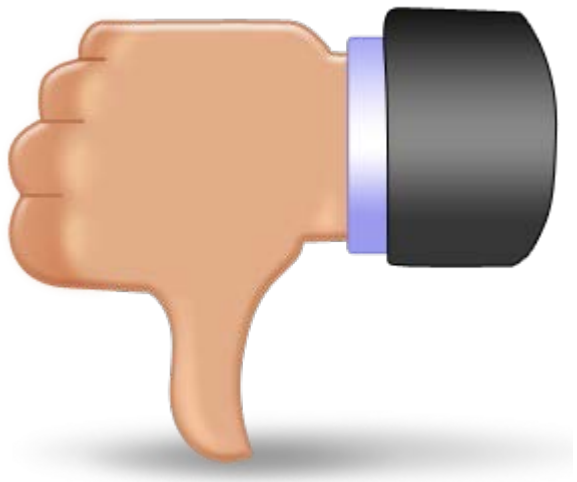
Pros

- Disciplined model where phases are complete one at a time
- Works well for small projects
- Easy to understand and use

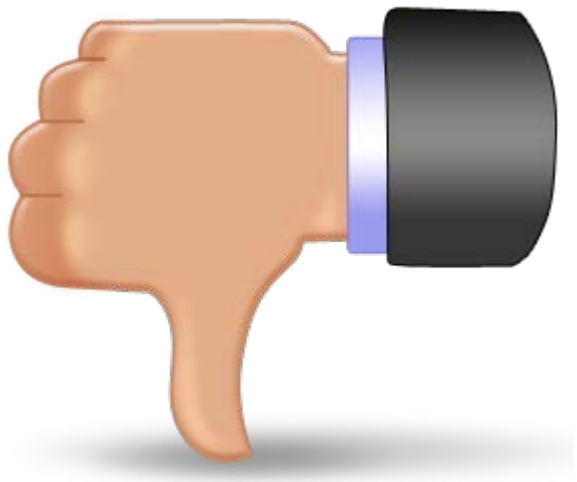
Pros

- Disciplined model where phases are complete one at a time
- Works well for small projects
- East to understand and use
- Easy to manage due to the rigidity of the model

Cons

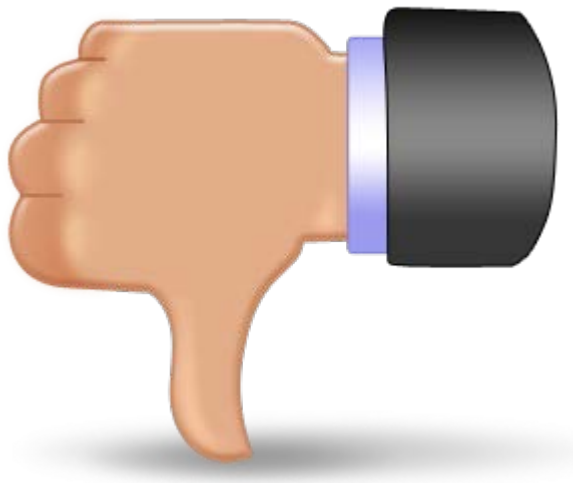


Cons



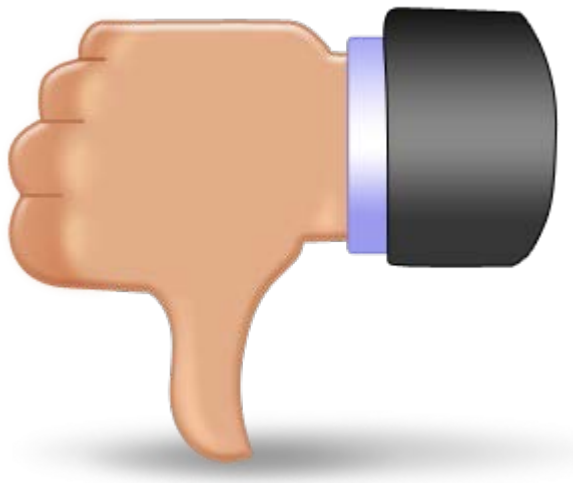
- High risk and uncertainty

Cons



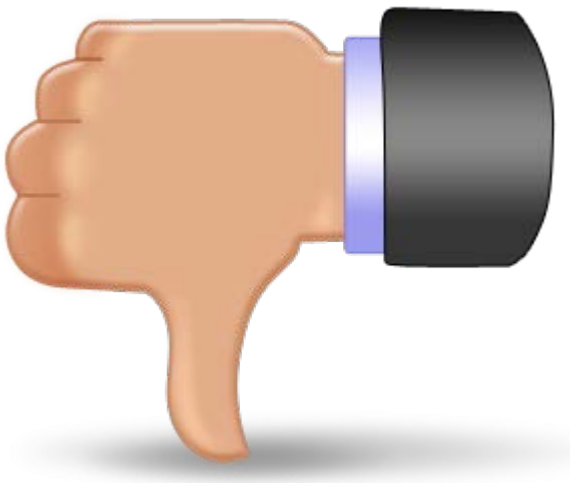
- High risk and uncertainty
- Not a good model for complex projects

Cons



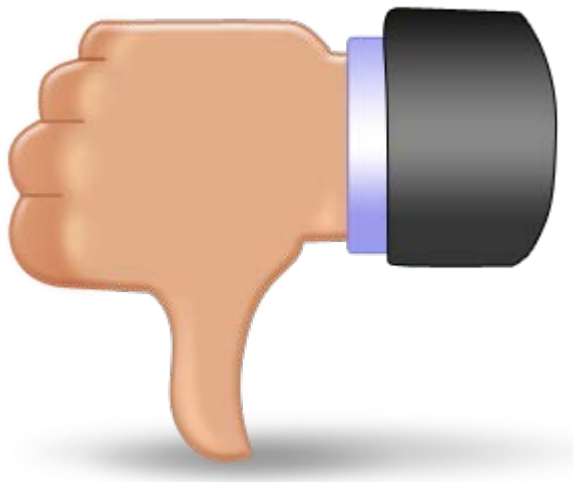
- High risk and uncertainty
- Not a good model for complex projects
- Not good for changes in requirements

Cons



- High risk and uncertainty
- Not a good model for complex projects
- Not good for changes in requirements
- Once in testing, difficult to go back

Cons



- No working software is produced until late in the day

Summary

- Waterfall and V-Model work in stages

Summary

- Waterfall and V-Model work in stages
- Each stages output is an input to the next stage

Summary

- Waterfall and V-Model work in stages
- Each stages output is an input to the next stage
- Changes in requirements are disruptive

Summary

- Waterfall and V-Model work in stages
- Each stages output is an input to the next stage
- Changes in requirements are disruptive
- Software delivered late in the process

Is There a Better Way?