Waterfalls, V's and Spirals: Standard SE Methodologies

Dr. Julie Greensmith

Objectives

- To introduce three of the major software process models:
  - Waterfall methods
  - Spiral Methodologies
  - V- model
- To describe the pros and cons of each model
- To show how these models are used in practice

The software process

- The software process is a structured set of activities required to develop a software system as pictured as a linear activity:

- A software process model is an abstract representation of a process.
- It presents a description of a process from some particular perspective
- Different methodologies utilise these stages in different orders and in different combinations

Lots of methodologies!

- Many methodologies are used in SE
- Which one you use is often dictated by:
  - the project manager
  - the complexity of the problem
  - organisational policy
- *Methodology:* a standard process or collection of methods used within a specific discipline, to provide a consistent and unified development process

Why do we need standards?

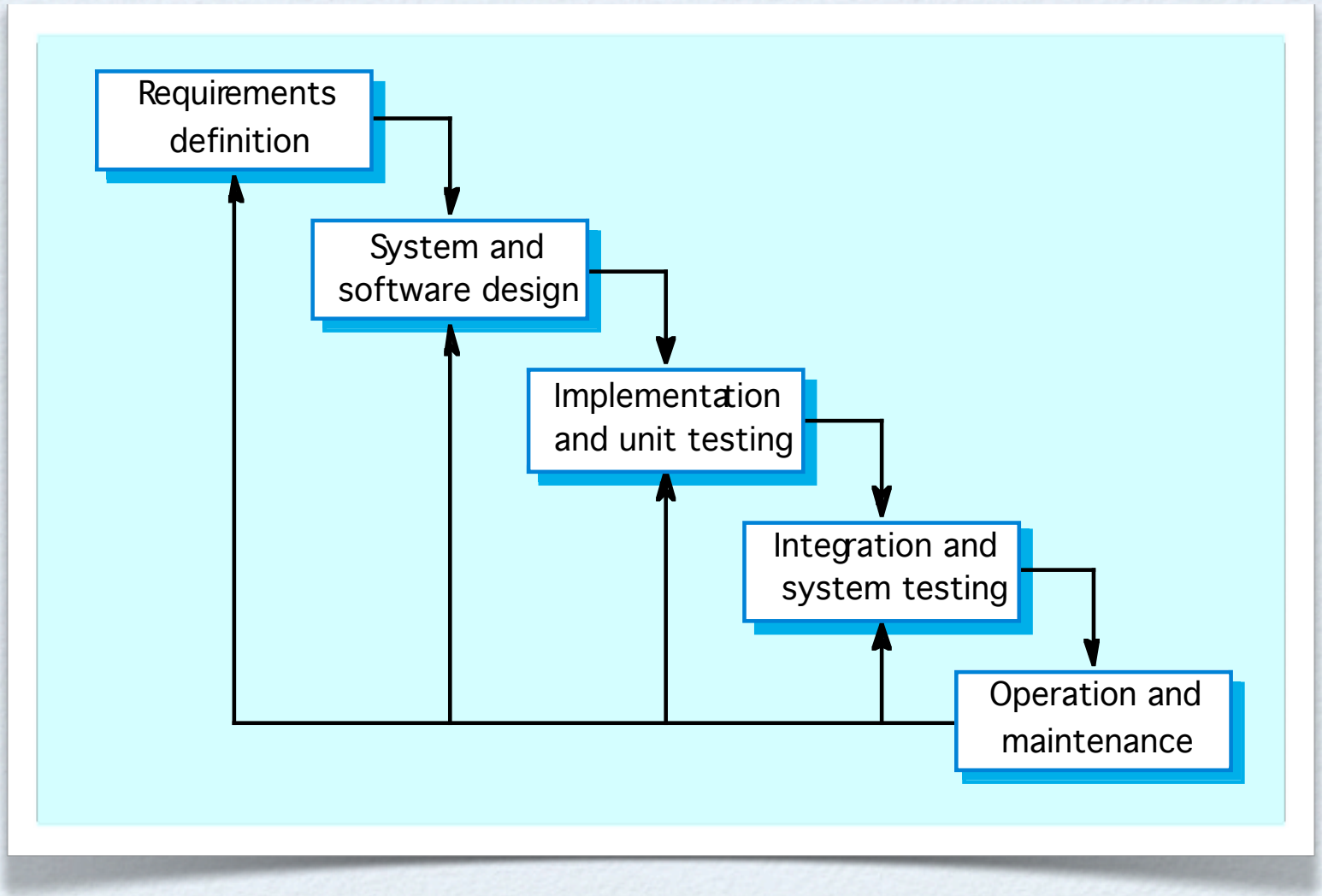
- *Surely the nature of a project should dictate how we go about developing the software?*
- This is not necessarily the case - these methods are important so that we have common frameworks in which to develop software
  - use common language
  - clear order in which components can be developed
  - makes it obvious where testing should be done
- Present in many industries not just in SE

The Waterfall Model

Generic software process models

- The waterfall model
  - Separate and distinct phases of specification and development
- Evolutionary development
  - Specification, development and validation are interleaved
- Component-based software engineering
  - The system is assembled from existing components
- There are many variants of these models e.g. formal development where a waterfall-like process is used but the specification is a formal specification that is refined through several stages to an implementable design


Requirements
Design
Implementation
Integration
Deployment
Maintenance


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Waterfall model phases

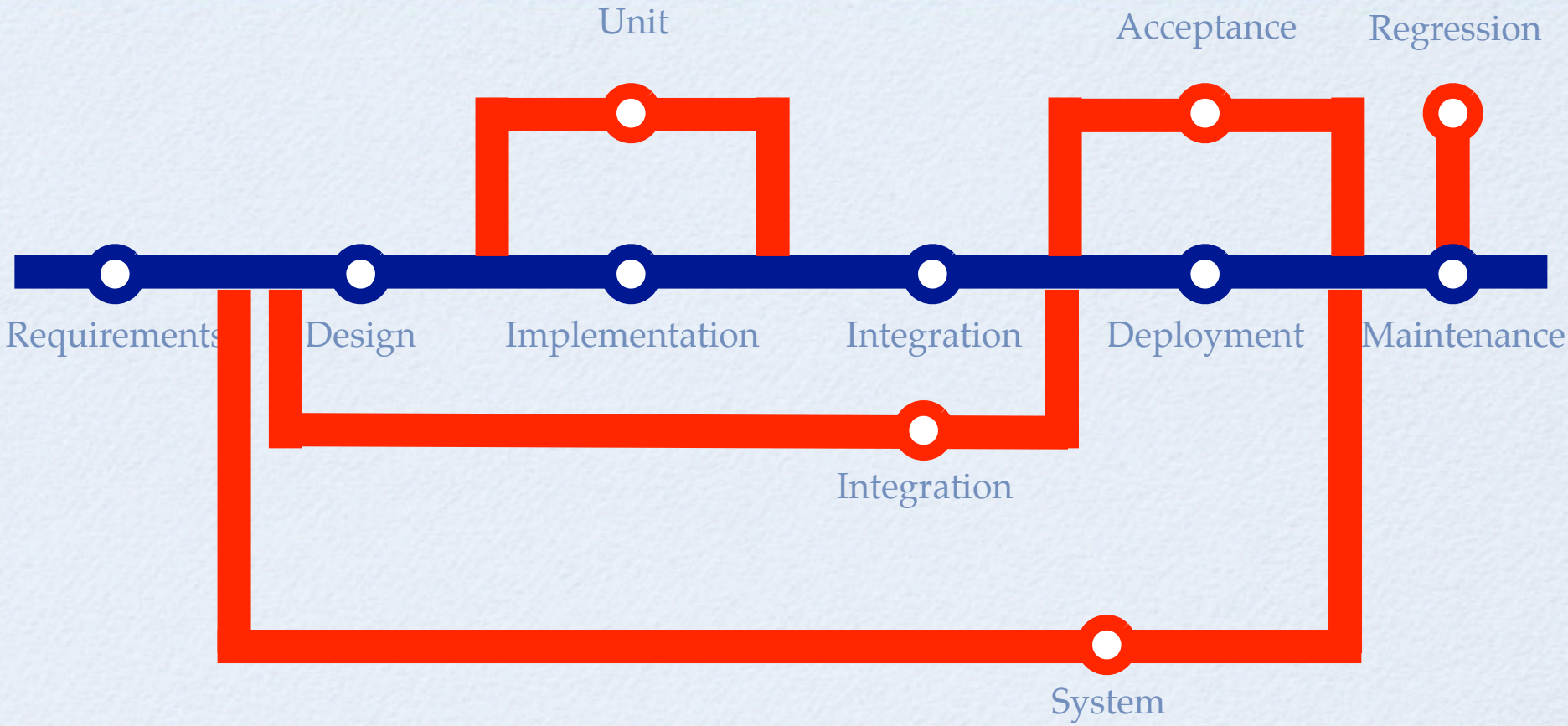
- The standard components of a software lifecycle
  - sometimes interaction between phases
  - generally cascades downwards, hence the name waterfall
- You cannot easily accommodate change!
  - One phase has to be complete before moving onto the next phase
  - Royce published the technique as an exemplar as to how *not* to develop software!

Waterfall model problems

- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing client requirements
- Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process
- Few business systems have stable requirements
- The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites including construction

Where does testing fit in?

Where does testing fit in?



Projects which use waterfalls

- Military and space development projects use waterfall methods
  - BAE systems
  - Aerospace development companies
  - Engineering heritage and mistrust of new and novel techniques
  - Do you remember Arianne-5??
- The NHS Information System - *is it running yet??*

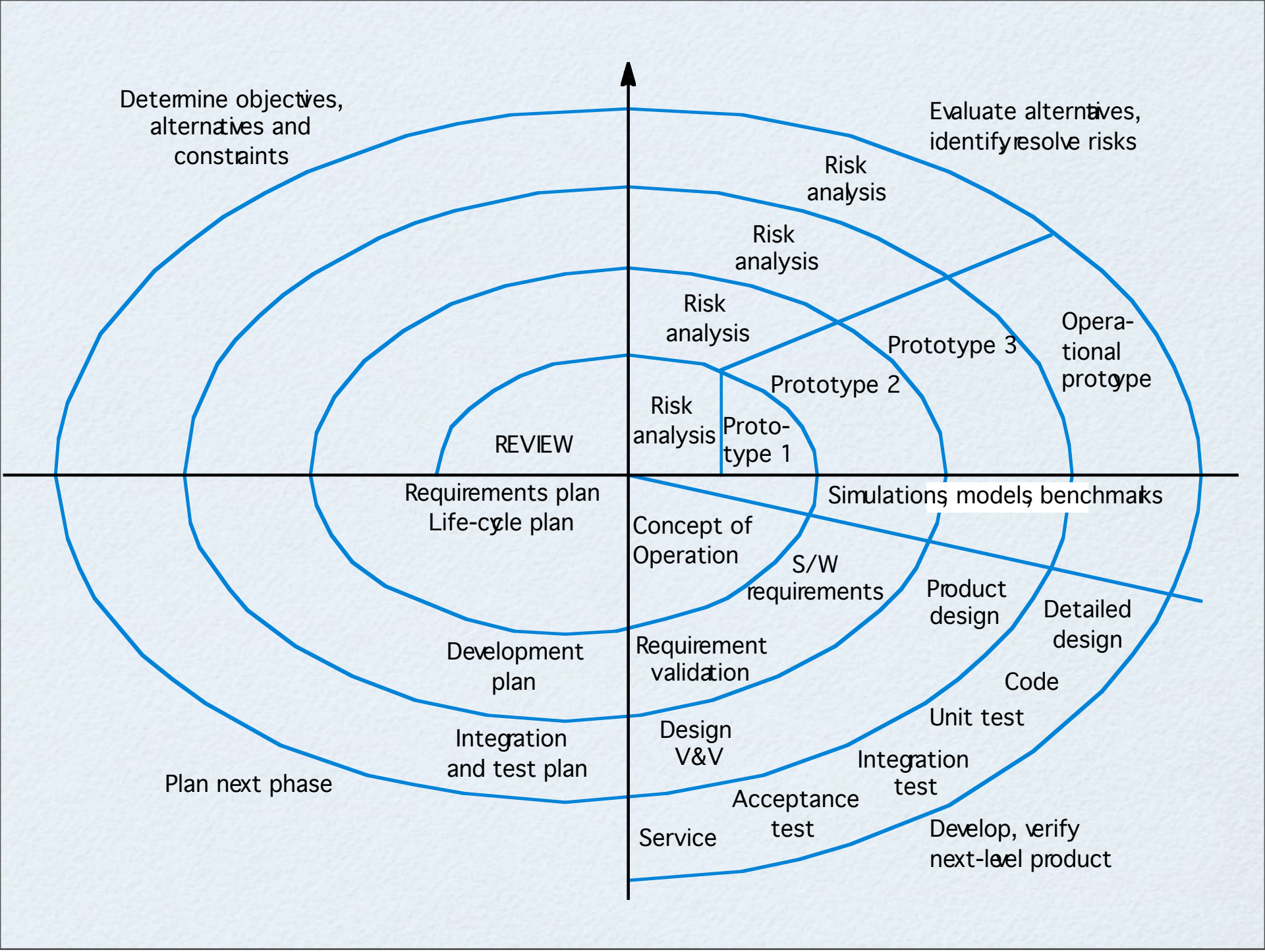
Variants on the Waterfall Model

- What we have described is known as the "pure" waterfall model
- Two commonly known variants:
  - Royce's model which links back one loop from implementation back to requirements
  - Big Design Up Front (BDUF) model where 60% of effort is spent in perfecting designs and not in heavily investing in prototyping
    - Experian use BDUF - clients "who know best" like it

Spiral Methodologies

Spiral development -Boehm

- Process is represented as a spiral rather than as a sequence of activities with backtracking.
- Each loop in the spiral represents a phase in the process and is a mini-waterfall
- No fixed phases or phase order such as specification or design
  - loops in the spiral are chosen depending on what is required, they are lovely and flexible!!
- Risks are explicitly assessed and resolved throughout the process.


Determine objectives, alternatives and constraints
Evaluate alternatives, identify, resolve risks
Risk analysis
Risk analysis
Risk analysis
Risk analysis
Prototype 3
Opera-tional prototype
Prototype 2
Proto-type 1
REVIEW
Requirements plan
Life-cycle plan
Concept of Operation
Simulations, models, benchmarks
S/W requirements
Product design
Detailed design
Development plan
Requirement validation
Code
Unit test
Integration and test plan
Design V&V
Integration test
Plan next phase
Acceptance test
Service
Develop, verify next-level product

Spiral model sectors

- Objective setting
  - Specific objectives for the phase are identified.
- Risk assessment and reduction
  - Risks are assessed and activities put in place to reduce the key risks.
- Development and validation
  - A development model for the system is chosen which can be any of the generic models.
- Planning
  - The project is reviewed and the next phase of the spiral is planned.

Spiral Model Stages

- Can be whatever is needed to get the job done
  - dictated by what you are trying to develop
  - depends what your resources are in terms of both people, time and money
- Can take a much more fine grained and less hierarchical view of the development process
  - risk analysis
  - specify where the prototypes are used
  - precise with testing points in development cycle

Spiral Model Pros

- Estimates of time and cost are easier to see as problems become apparent quite early on
  - each rotation of the spiral is a mini-version of the whole project
- It is robust to coping with an evolving requirements specification or changes in underlying technology
- The development team can get stuck in sooner, meaning less idle people in the project team

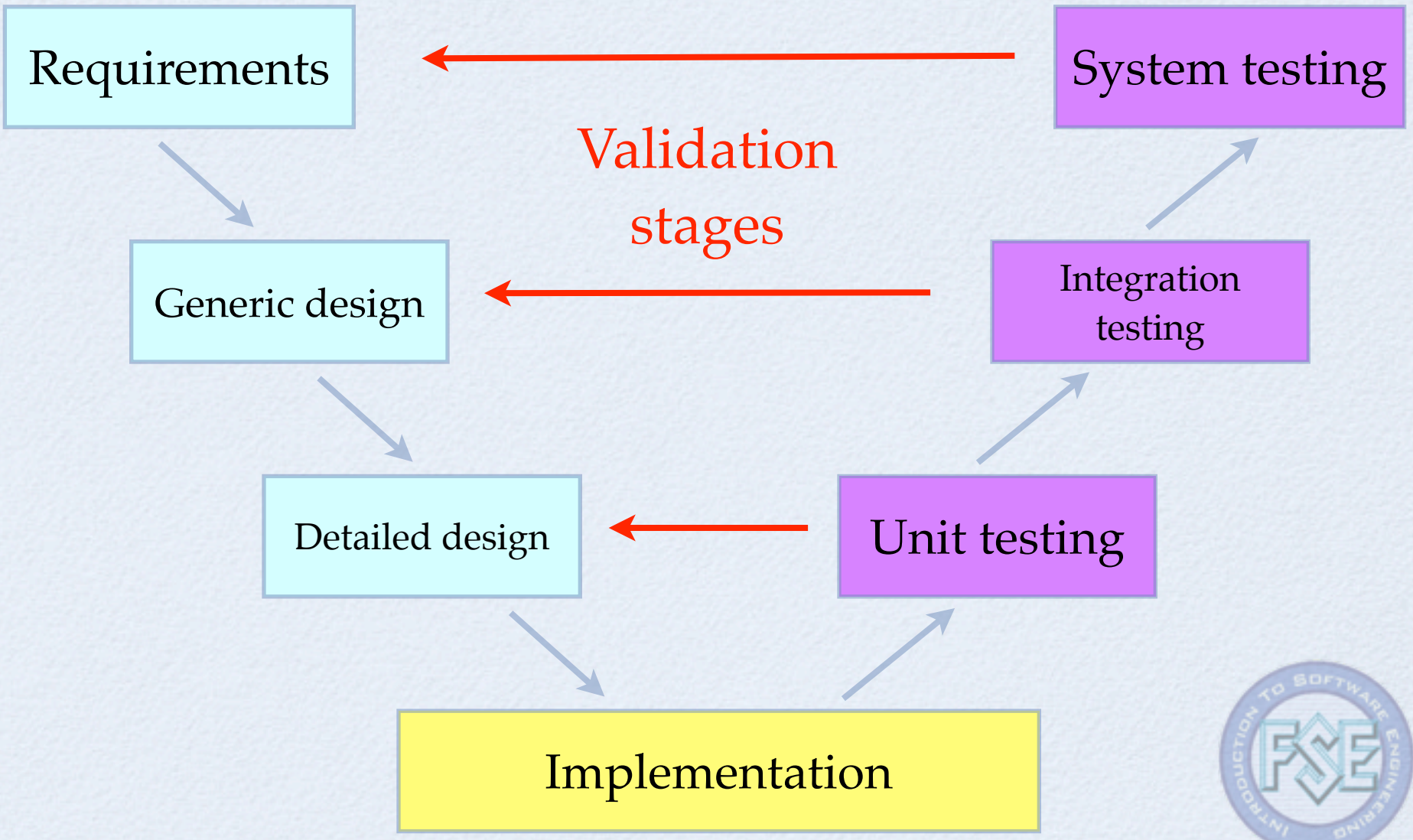
Spiral Model Cons

- Limited re-use as spirals tailored per project
- Spirals applied to different problems in a different way
- Hard to see the overall design up front
  - managers get nervous with things like this
- Hard to keep on track in terms of time and budget as its not explicit in the original spiral

V-Models

V- Development

- An implementation sits underneath interacting layers of design/planning with integration and testing
- Is less flexible but more structured than the spiral methods
- Can account for more changes than a pure waterfall model
- Popular with developers as implementation is seen as the keystone of this approach


Requirements
System testing
Validation
stages
Generic design
Integration
testing
Detailed design
Unit testing
Implementation

Pros and cons of V-model

- In theory, used by Thales for aerospace systems
- Pros:
  - Each phase has specific deliverables
  - Simple and Easy to use
- Cons:
  - Very Rigid similar to the Waterfall method
  - As there is a fixed implementation phase there is little scope for prototype development

The Truth....

...Choice is not always yours

- Which approach is dictated by the organisational culture of your institution
- Smaller staff numbers -> more iterative based models
- For an undergraduate project with a *single developer* a waterfall based approach may be appropriate
- *The right methodology depends on the context of the software development environment*

Summary

- To introduce three of the major software process models:
  - Waterfall methods
  - Spiral Methodologies
  - V- model
- To describe the pros and cons of each model
- To show how these models are used in practice