(11224) INTRODUCTION TO SOFTWARE ENGINEERING

Lecture 2: Software Process

Shereen Fouad

Motivation

Why do software projects fail?

Motivation

Why do software projects fail?

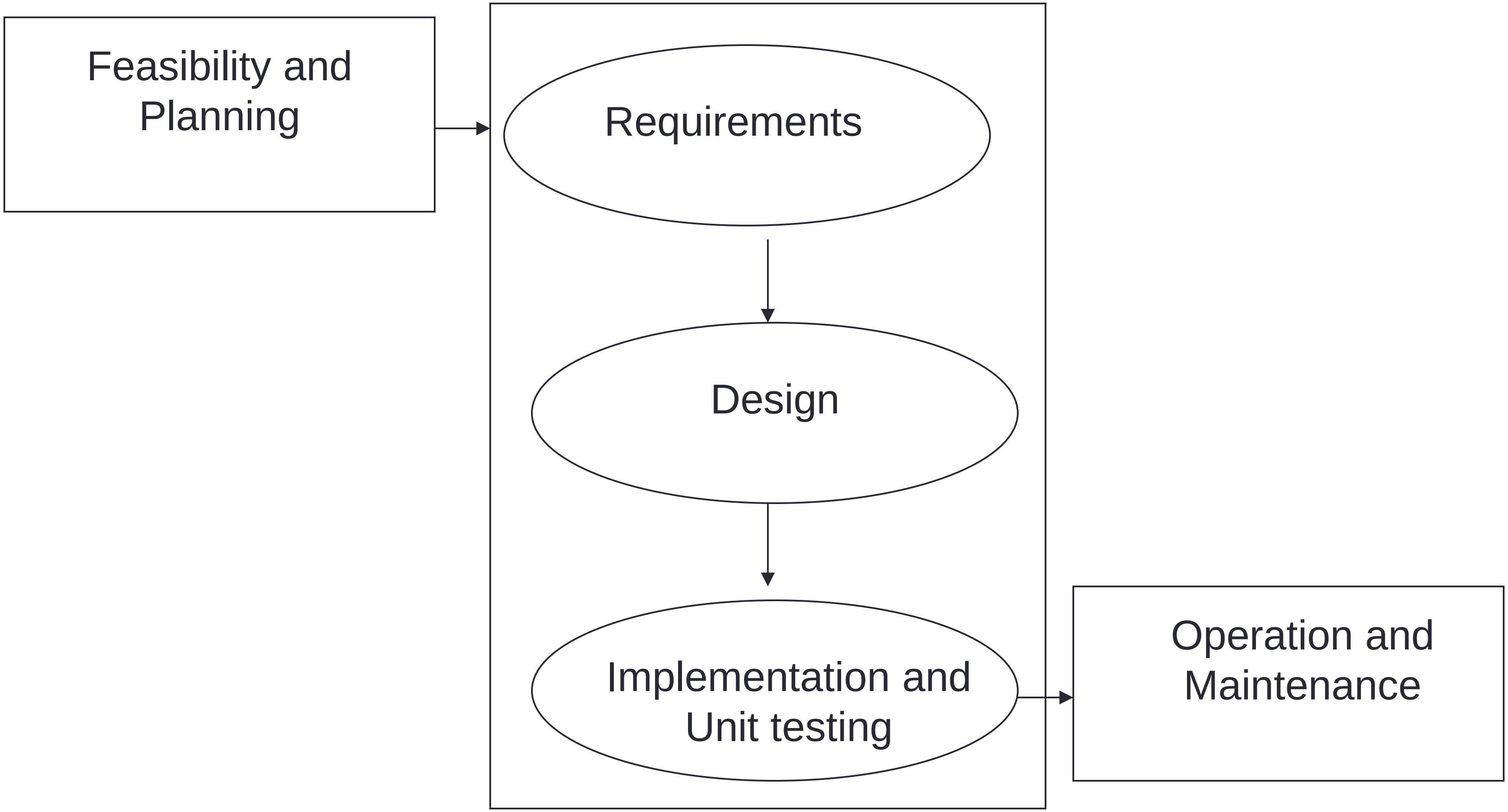
- Unrealistic project goals
- Inaccurate estimates of needed resources
- Bad planning
- Unmanaged risks
- Poor communication
- Inadequate estimation resources
- Unclear milestones
- **Poor process oversight**
- **Inconsistency**

… *so many reasons!!!*

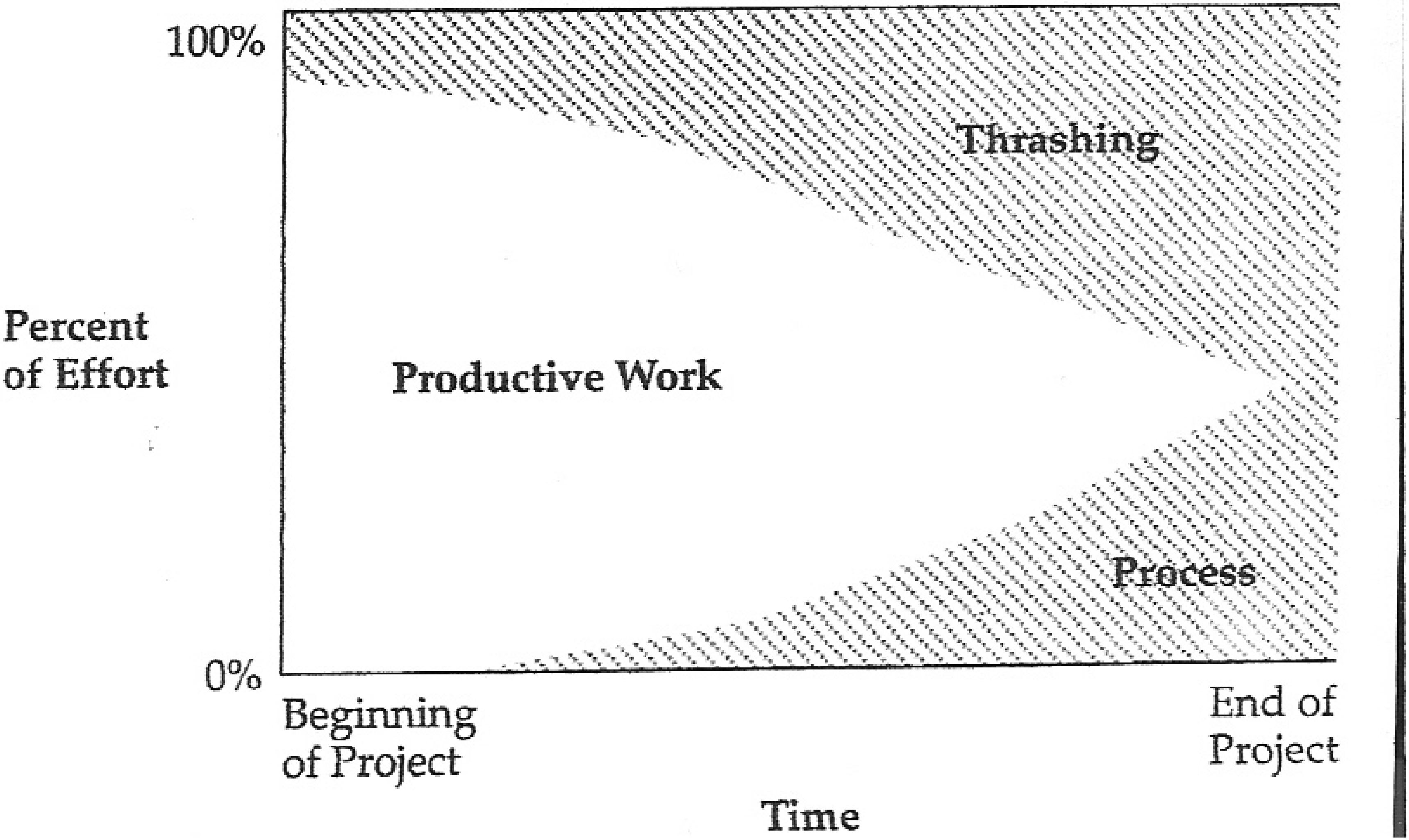
Software Processes

- Structured set of activities to develop software systems
- Defines what task to do and how to do them.
- Processes are at the very centre of software engineering
- Successful software development projects all need to address similar issues.
- This creates a number of process steps that must be part of all software projects within them

Basic Process Steps in all Software Development

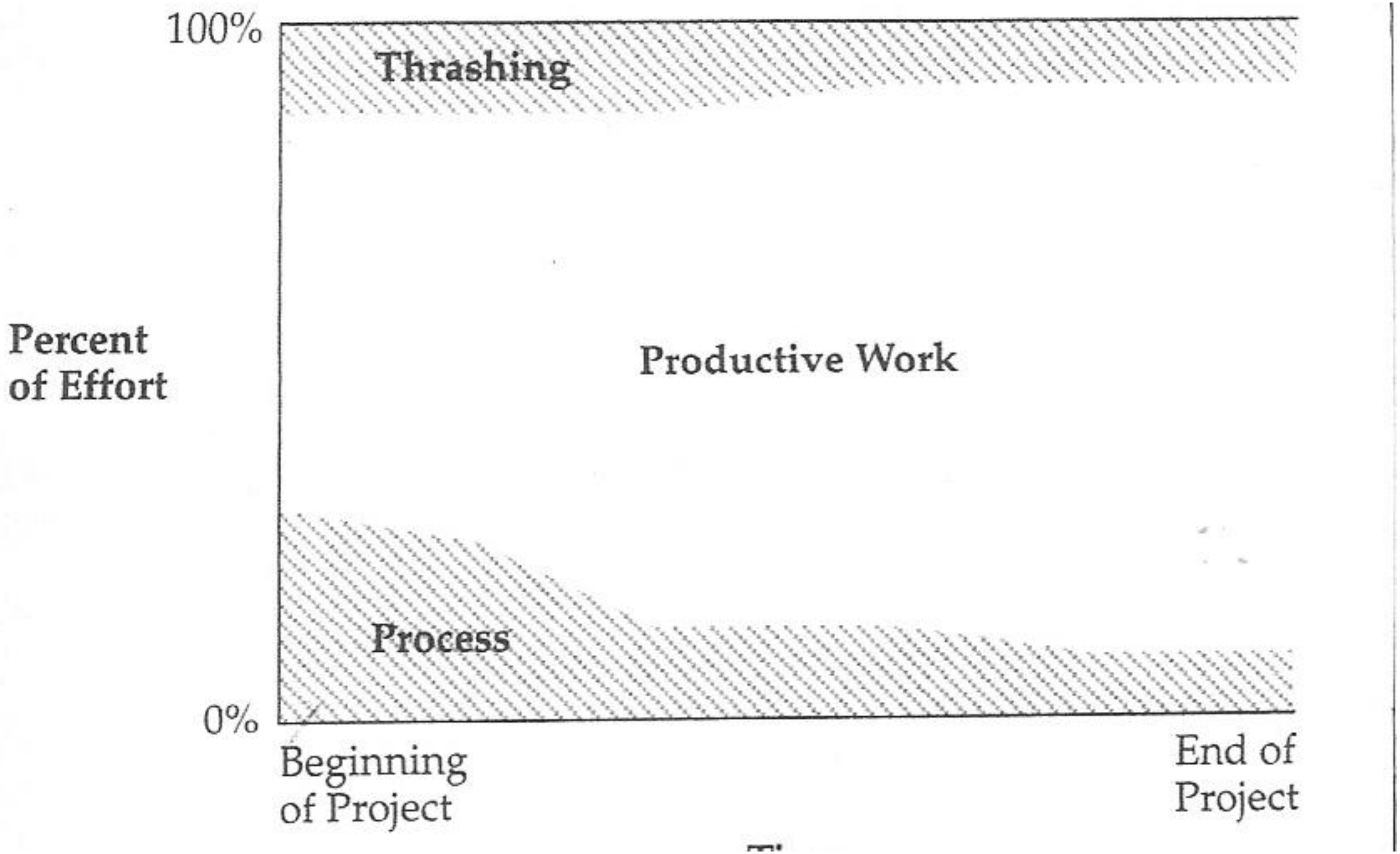


Project With Little Attention to Process



Source: http://www.stevemcconnell.com/articles/art09.htm

Project with Good Attention to Process



Source: http://www.stevemcconnell.com/articles/art09.htm

Software Process Model

- Abstract representation of a process.
- It presents a description of a process from some particular perspective.
- Many different processes might lead to completion of a software development project.
- Software products are very varied, therefore, there is no standard process model
- **Choose a process model that**
  - *maximizes quality and*
  - *minimizes time*
  - *minimizes cost*
  - *matches the kind of project we are undertaking,*

Elements of Software Process

- Development process, that specifies all the engineering activities
- Project management process, that specifies how to plan and control those activities
- Software configuration control process, that manages the large collection of items (code files, requirements documents, documentation, designs etc.) that are created, updated and deleted during the progress of the project.

Software Process Model

- Waterfall
- Spiral
- Iterative and incremental
- RAD (Rapid Application Development)
- Agile
- XP (Extreme Programming)
- Scrum

- And many others …

Software Process Model

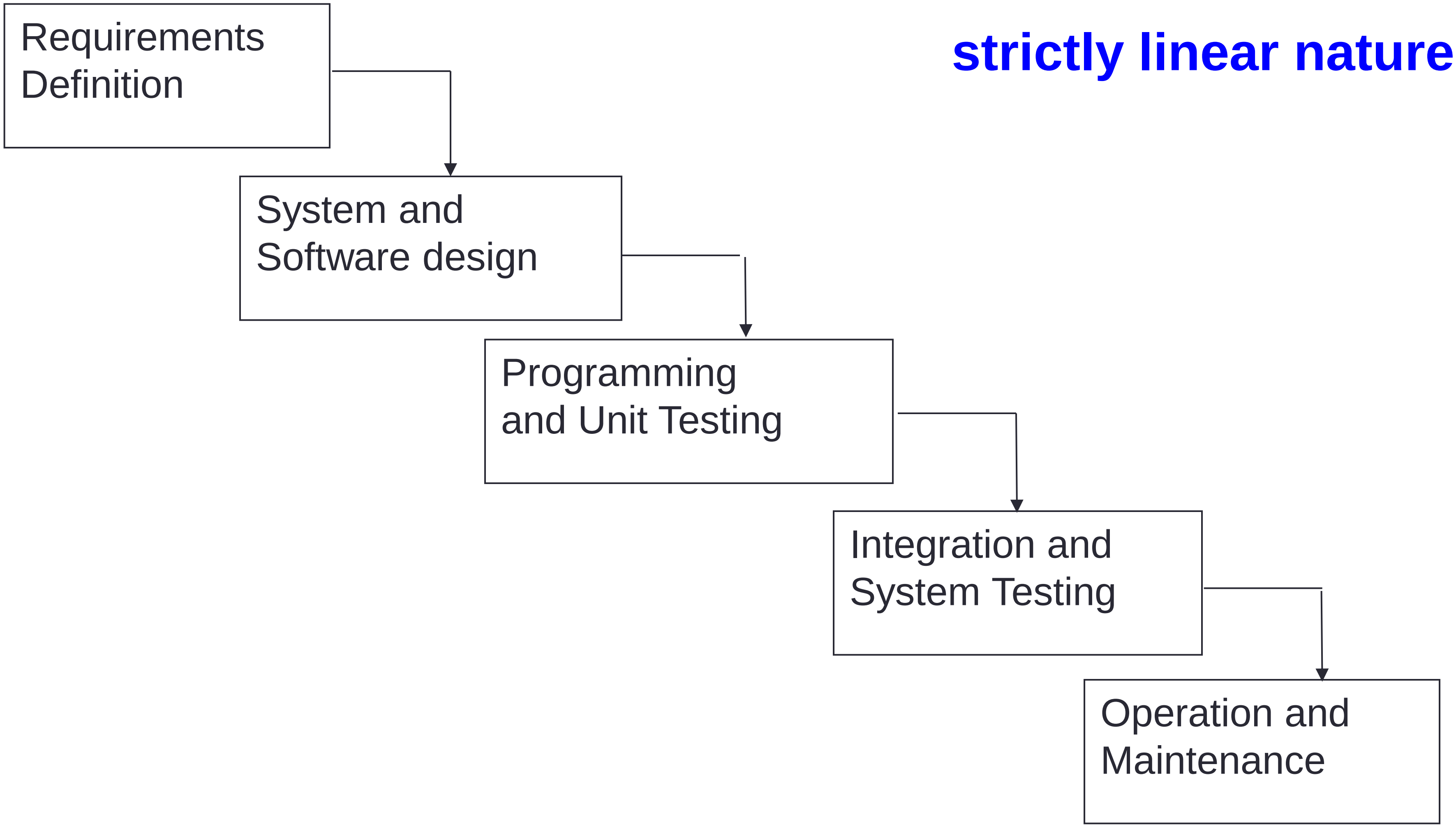
- **Waterfall**
- Spiral
- Iterative and incremental
- RAD (Rapid Application Development)
- Agile
  - XP (Extreme Programming)
  - Scrum

- And many others …

Waterfall Model

- The simplest and most venerable process model
- There are separate identified phases in the waterfall model
- A phase has to be complete before moving onto the next phase.
- This organizes the phases in a strict linear order with feedback loops
- There are a number of variations and adaptations of this process model.

Waterfall Model


Requirements Definition

System and Software design

Programming and Unit Testing

Integration and System Testing

Operation and Maintenance

strictly linear nature !

Waterfall Model

- **A pure sequential model is impossible**
- Examples:
  - A feasibility study cannot create a proposed budget and schedule without a preliminary study of the requirements and a tentative design.
  - Detailed design or implementation usually reveals gaps in the requirements specification.

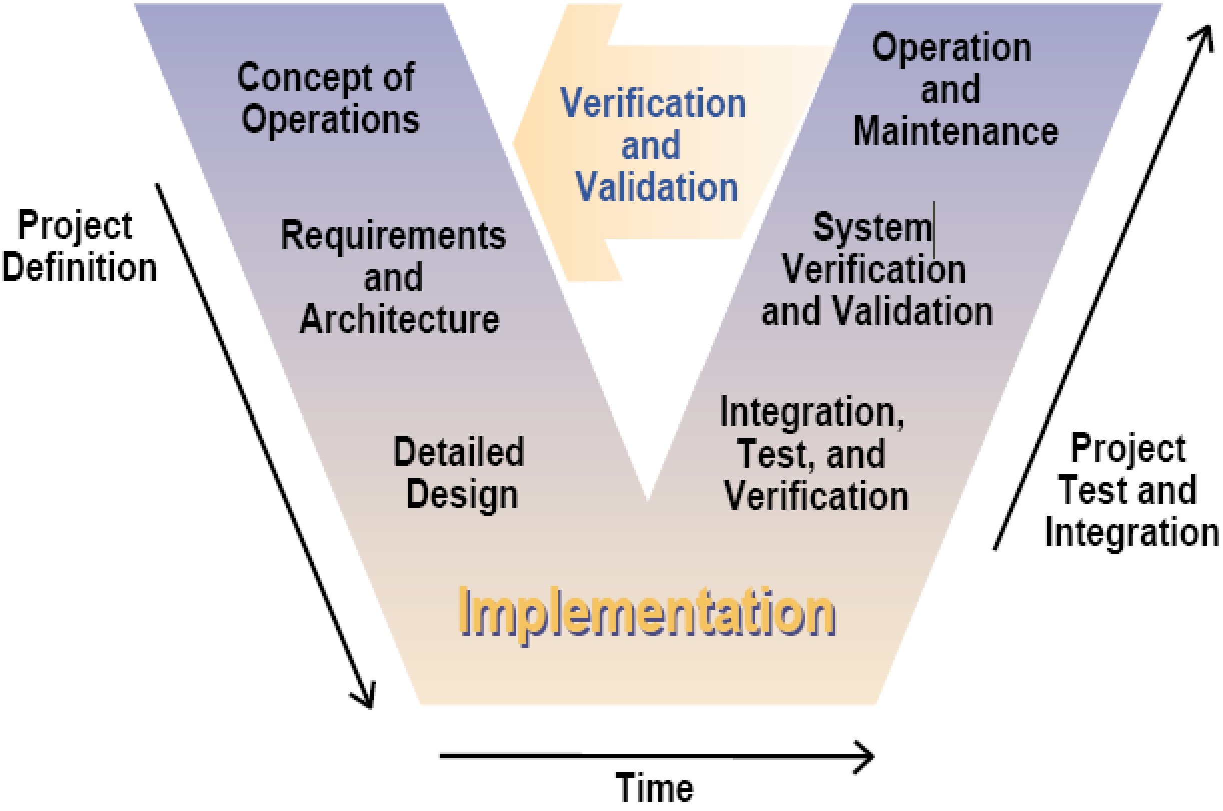
Waterfall is a Document Driven Process

- Documents are produced during each phase and form the input to the following phase.
- Documents must be accurately maintained
- A phase is not complete until its documents have been approved (signed-off).
- If you are not producing end-of-phase documents, you are not doing Waterfall!

Verification and Validation

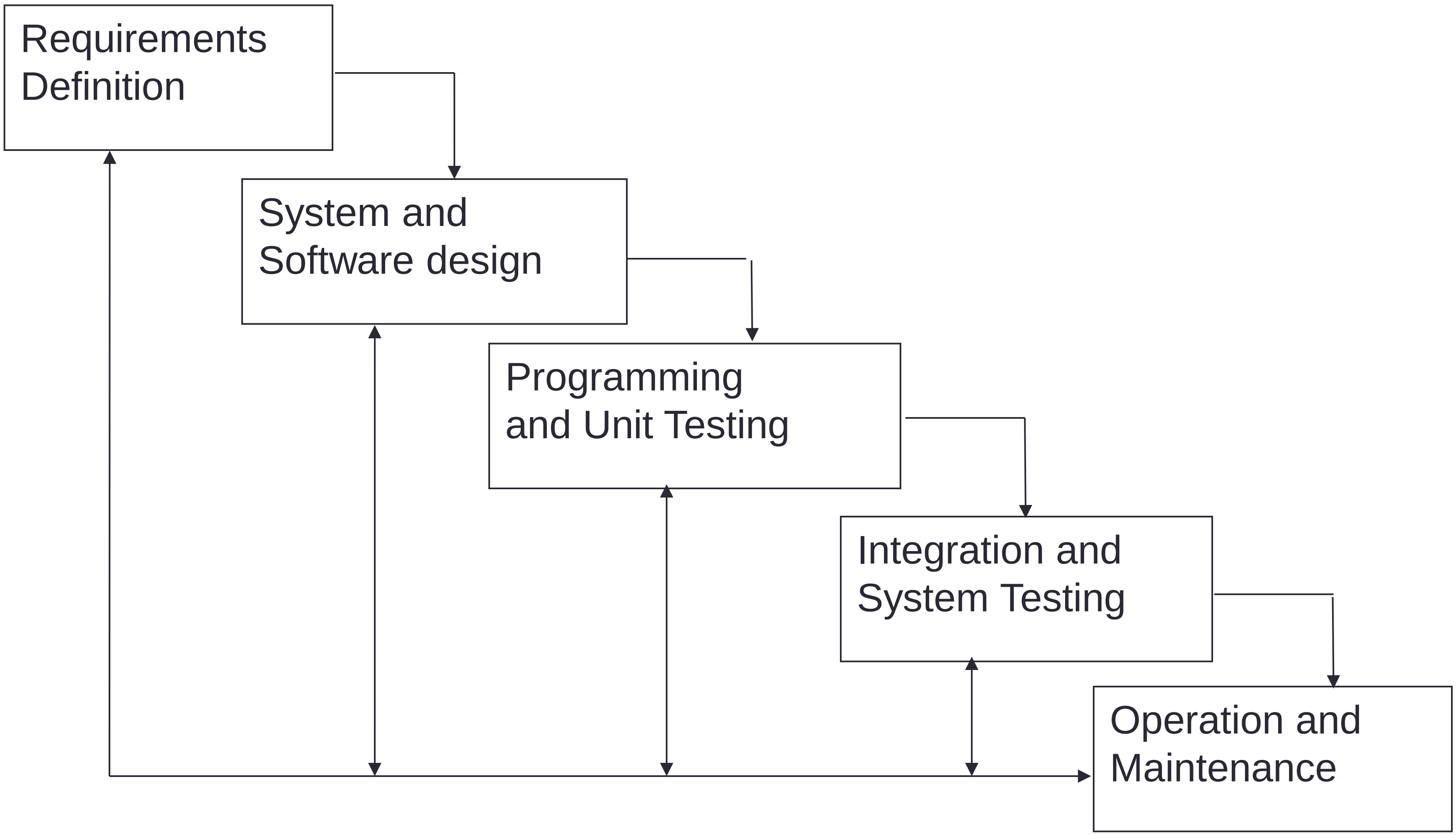
- High confidence in the correctness of the result of a phase is achieved before moving on to the next phase.
- Apply verification and validation activities at the end of each phase.
- **Verification:**
  - The process of evaluating the output of a phase to determine whether the products satisfy the conditions imposed at the start of that phase.
- **Validation:**
  - The process of evaluating the output of a phase to determine whether the products satisfy the requirements of the project.

Verification and Validation


Concept of Operations
Requirements and Architecture
Detailed Design
Verification and Validation
Operation and Maintenance
System Verification and Validation
Integration, Test, and Verification
Implementation
Project Definition
Project Test and Integration
Time

"Systems Engineering Process II" by Leon Osborne, Jeffrey Brummond, Robert Hart, Mohsen (Moe) Zarean Ph.D., P.E, Steven Conger

Feedback in the Waterfall Model



Requirements Definition

System and Software design

Programming and Unit Testing

Integration and System Testing

Operation and Maintenance

Feedback in the Waterfall Model

- It is a heavyweight task
- If many feedback loops are necessary then the whole project is at risk of going over budget, over time or even failing altogether.

Disadvantages of Waterfall Model

- Requirements are frozen before design.
- Target hardware and environment must normally be chosen at the same time as the requirements are frozen.
- The process leads to a high risk “all or nothing” situation
  - There is no useful system until the very end of the project, and the client can not see what will be delivered until the very end.
- It encourages requirements bloat
- requirements analysis phase is the only opportunity that
  - clients have to influence what the end product will contain,

When Waterfall is severely inappropriate?

When Waterfall is severely inappropriate?

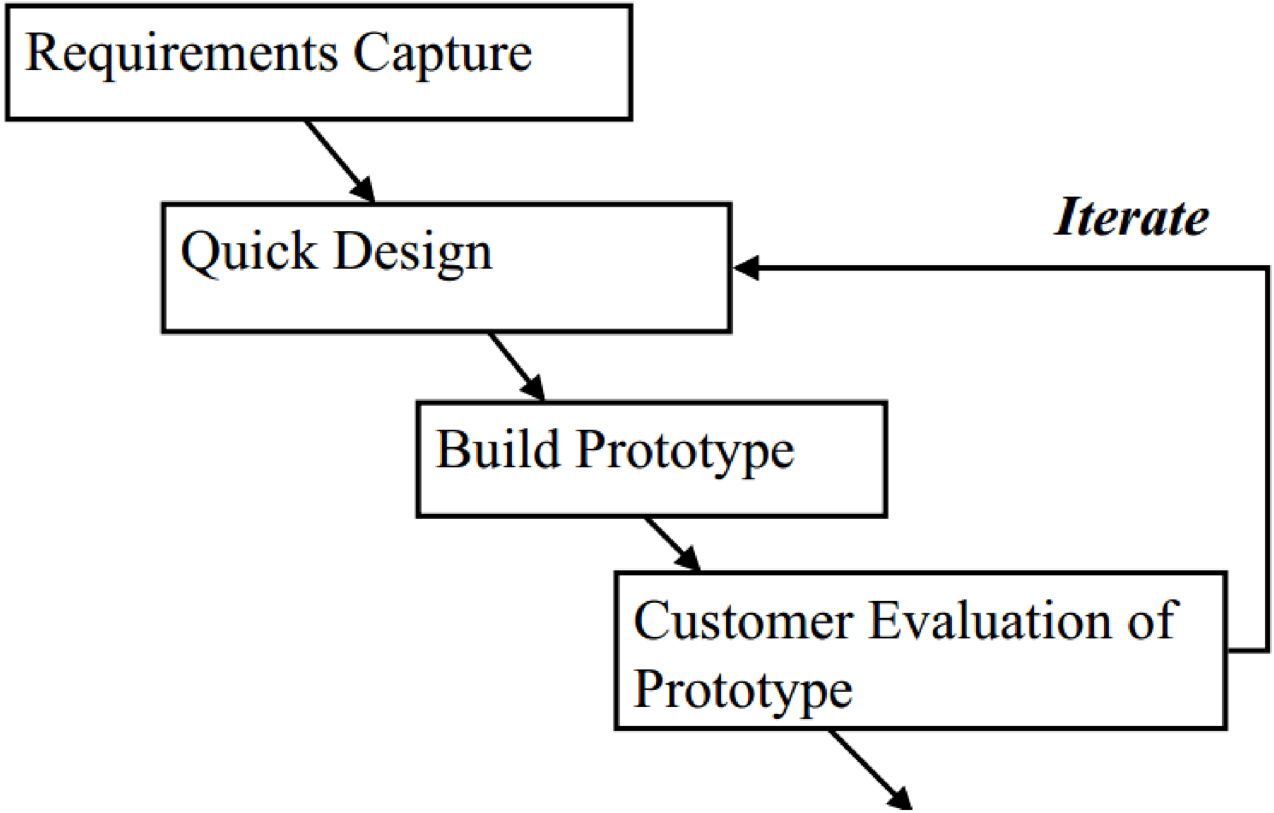
- if one can expect significant changes to requirements during a project.
- if the requirements are hard to elicit, for example, if the clients are not clear what they need or if the application area is a new one that is not well understood.

When Waterfall is Appropriate to Use?

When Waterfall is Appropriate to Use?

- 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.
  - In those circumstances, the plan-driven nature of the waterfall model helps coordinate the work.

Waterfall Variation: Rapid Prototyping during Requirements Analysis



Requirements Capture

Quick Design

Iterate

Build Prototype

Customer Evaluation of Prototype

The Rapid Prototype Workflow

process continues with the usual, unchanged Waterfall phases.

Waterfall Variation: Rapid Prototyping during Requirements Analysis

- **Advantages:**
- Get, at minimal cost, some idea about the requirements
- Early feedback to the clients
  - so that they can determine if the direction that the requirements are heading towards matches their understanding of what they need.
- Reduce the risk associated with projects where the requirements are hard to elicit.

Waterfall Variation: Rapid Prototyping during Requirements Analysis

- **Disadvantages:**
- It does not help with projects where the requirements change during the rest of the project.
- An unstable/badly implemented prototype often becomes the final product.
- Requires extensive customer collaboration

Software Process Model

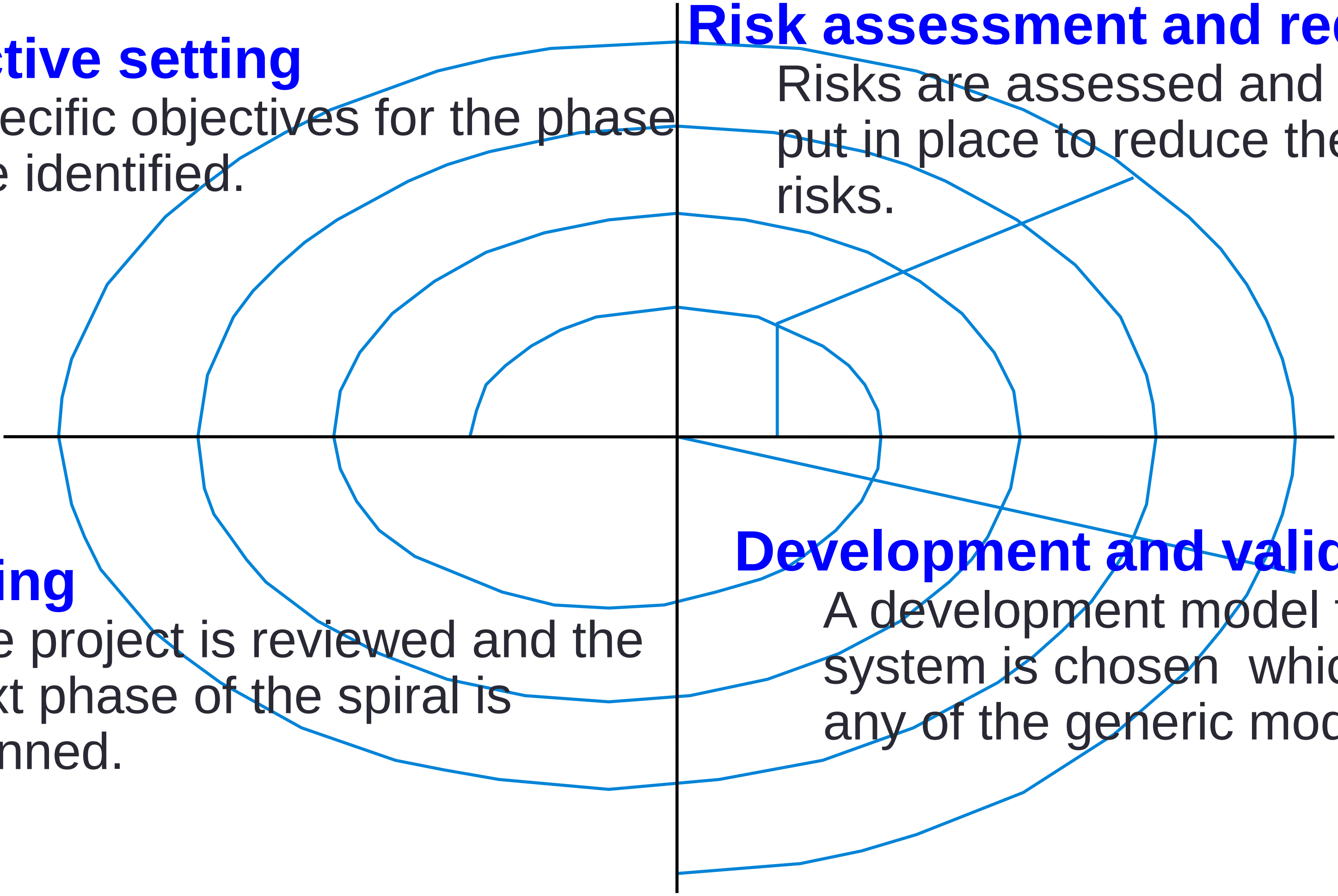
- Waterfall
- **Spiral**
- Iterative and incremental
- RAD
- Agile
- XP
- Scrum

- And many others …

Spiral Model

- 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 e.g. project inception in the innermost loop, requirements capture in the next loop, system design after that, etc.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
- Risks are explicitly assessed and resolved throughout the process.

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.

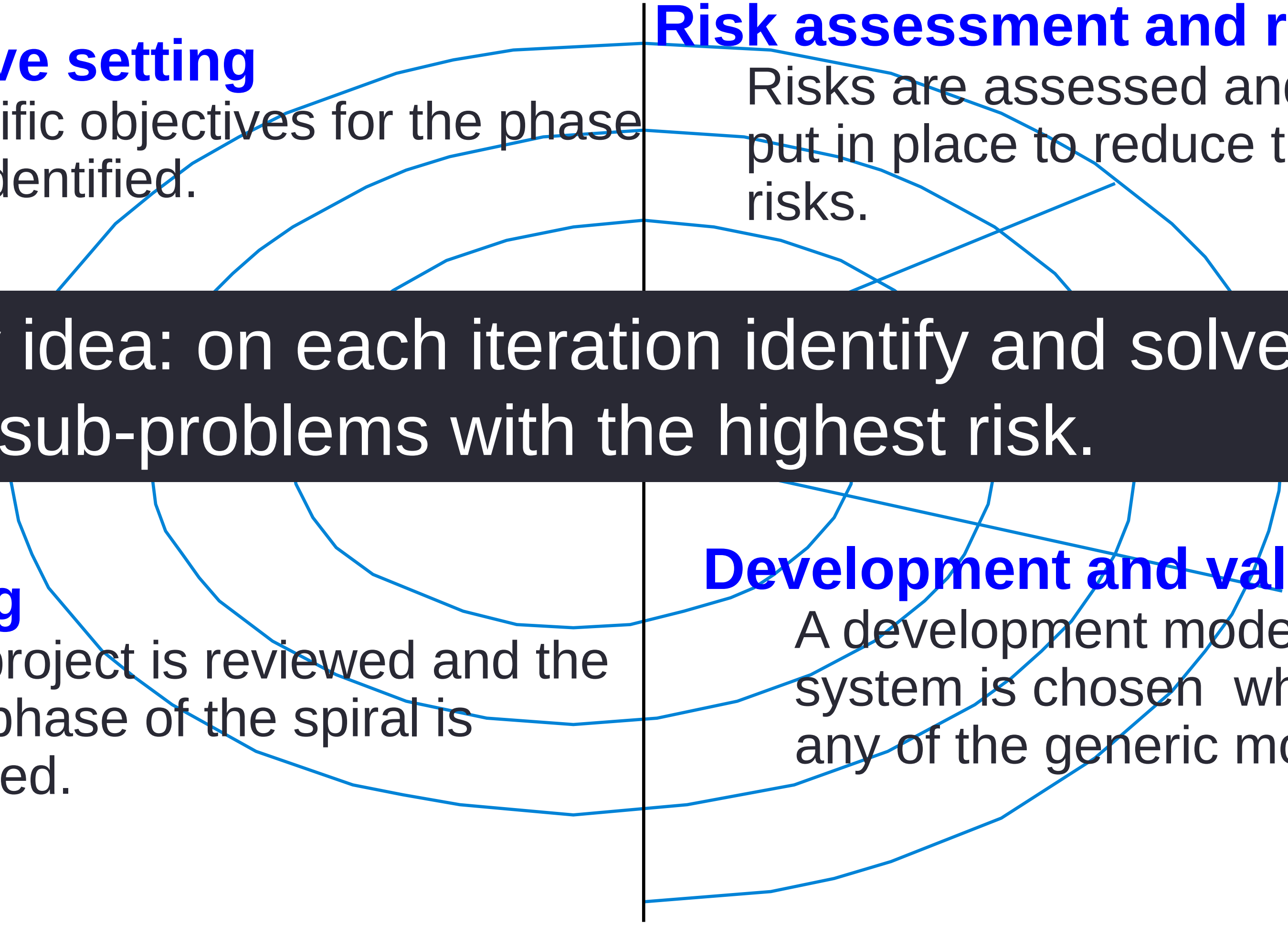
Planning

The project is reviewed and the next phase of the spiral is planned.

Development and validation

A development model for the system is chosen which can be any of the generic models.

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.

Key idea: on each iteration identify and solve the sub-problems with the highest risk.

Planning

The project is reviewed and the next phase of the spiral is planned.

Development and validation

A development model for the system is chosen which can be any of the generic models.

Spiral Model

- **Advantages:**
- Realistic in terms of handling the unclear requirements
- Comprehensive model decreases risk
- Good project visibility.
- **Disadvantages:**
- Needs technical expertise in risk analysis and risk management to work well.
- Model is poorly understood by nontechnical management, hence not so widely used

Quiz !

- One can choose Waterfall Model if the project development schedule is tight.
  A) True
  B) False

- The process of checking whether we are building the right product is

A) Validation

B) Verification

C) Testing

- Which of the following life cycle model can be chosen if the development team has less experience on similar projects?
  a) Spiral
  b) Waterfall
  c) Iterative Enhancement Model

Software Process Model

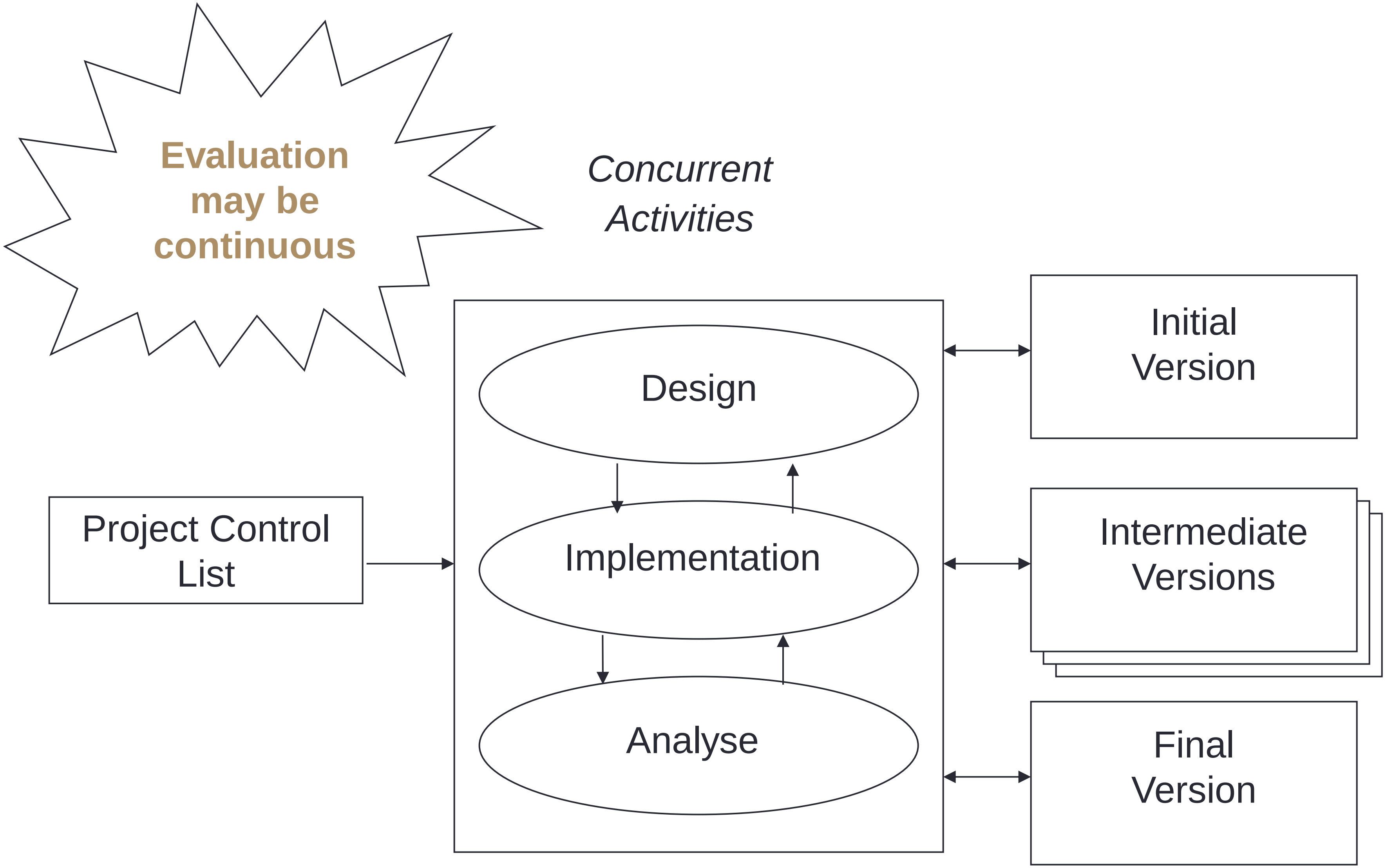
- Waterfall
- Spiral
- **Iterative and incremental**
- RAD
- Agile
- XP
- Scrum

- And many others …

Iterative and Incremental Models

- Software should be developed in increments.
- The result of each increment should be a working, demonstrable system with a bit more functionality than the result of the previous increment.
- Each increment involves taking the next task from the list and dealing with that task in 3 phases:
  - Design the implementation for this task.
  - Implement this task.
  - Analyse the result and update the project control list.

Iterative and Incremental Models



Iterative Models

- **Advantages**:
- Supports the problems of changing requirements well,
  - as the analysis phase of one iteration can identify design or requirements defects and add further tasks to redesign or reemployment parts of the system or to add new functionality to deal with newly discovered requirements.
- provides plenty of feedback to the client about what can be expected from the final product during the course of the project.
- **Disadvantages**:
- Include the extra costs of having to redesign or reimplement existing parts of the system,

References

- A number of slides in this talk is based on:
  - Alan P. Sexton hand-outs (Introduction to Software Engineering. The University of Birmingham. Spring Semester 2014)
  - SOFTWARE ENGINEERING 9 Ed. by Ian Sommerville

Thank You ☺