Lecture 4 – Software Development Methodologies

SE 217 - Principles of Software Engineering

Categories of Methodologies (Focus on process or data)

- Process Centered
- Data Centered
- Object-Oriented

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Categories of Methodologies (Sequence of SDLC phases)

- Structured Design (dominant 1980s)
  - Waterfall Development
  - Parallel Development

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Categories of Methodologies (Sequence of SDLC phases)

- Rapid Application Development (popular in 1990s)
  - Phased
  - Prototyping
  - Throwaway Prototyping

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Categories of Methodologies (Sequence of SDLC phases)

- Agile Development (emphasize on iterative development)
  - eXtreme Programming
  - Scrum

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Plan-Driven and Agile Processes

- Plan-driven (also called structured, heavyweight) processes are processes where all of the process activities are planned in advance and progress is measured against this plan

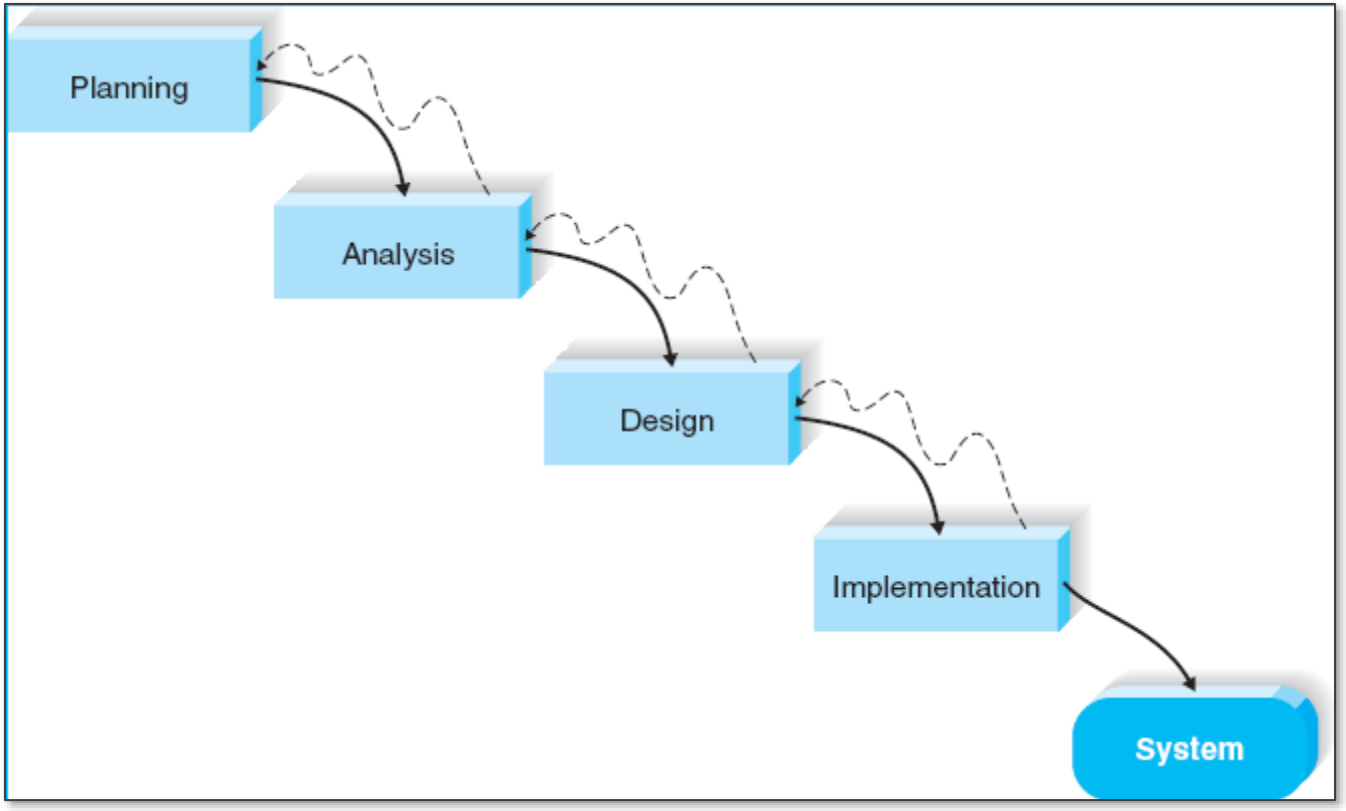
Plan-Driven and Agile Processes

- In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements

Plan-Driven and Agile Processes

- In practice, most practical processes include elements of both plan-driven and agile approaches
- There are no right or wrong software processes

Software Development Life Cycle Models



Waterfall Development

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Waterfall Development

- Also called "one shot" or "once through"
- Advantages
  - Identifies system requirements long before programming begins
  - Minimizes changes to the requirements as the project proceeds

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Waterfall Development

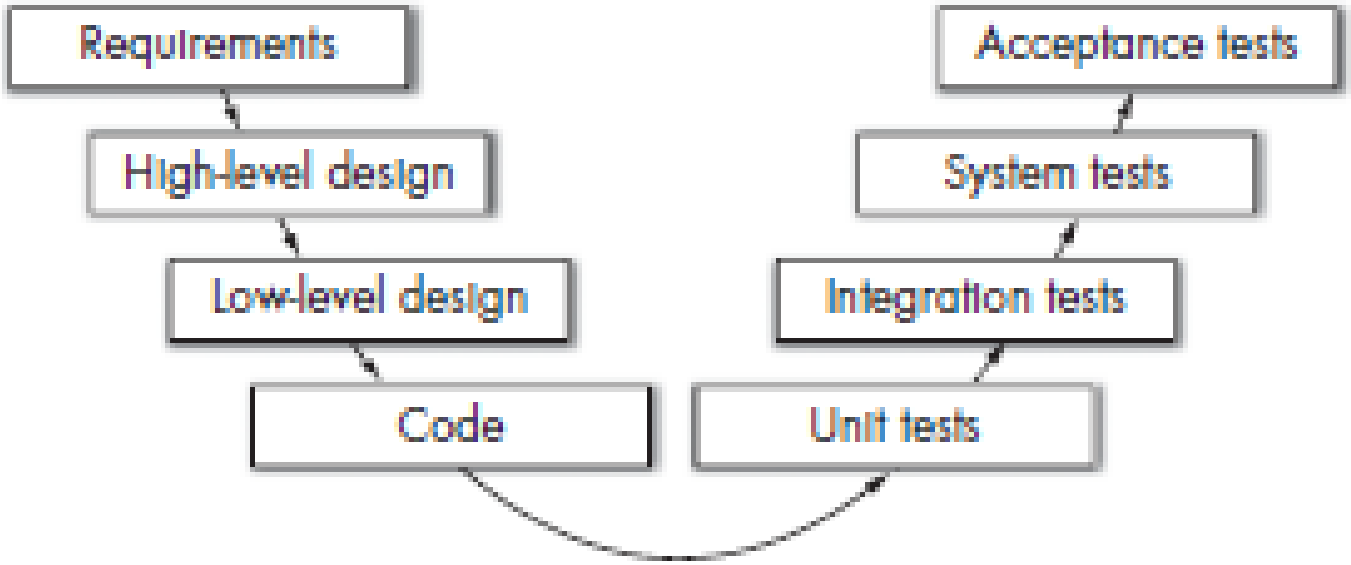
- Disadvantages
  - Design must be completely specified before programming begins
  - A long time elapses between the system proposal and the system delivery

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Waterfall Development

- 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

V Model



Code craft: the practice of writing excellent code / Pete Goodliffe

V Model

- On the left, we see the development phases leading up to software construction:
  - The planning, design, and implementation work

V Model

- The right-hand stream governs testing and approval
- Each level of test work is measured against the specification generated from the corresponding left-hand phase

V Model

- The testing phases (in the right branch) can begin in parallel to the development work (the left branch), and are given an equal importance

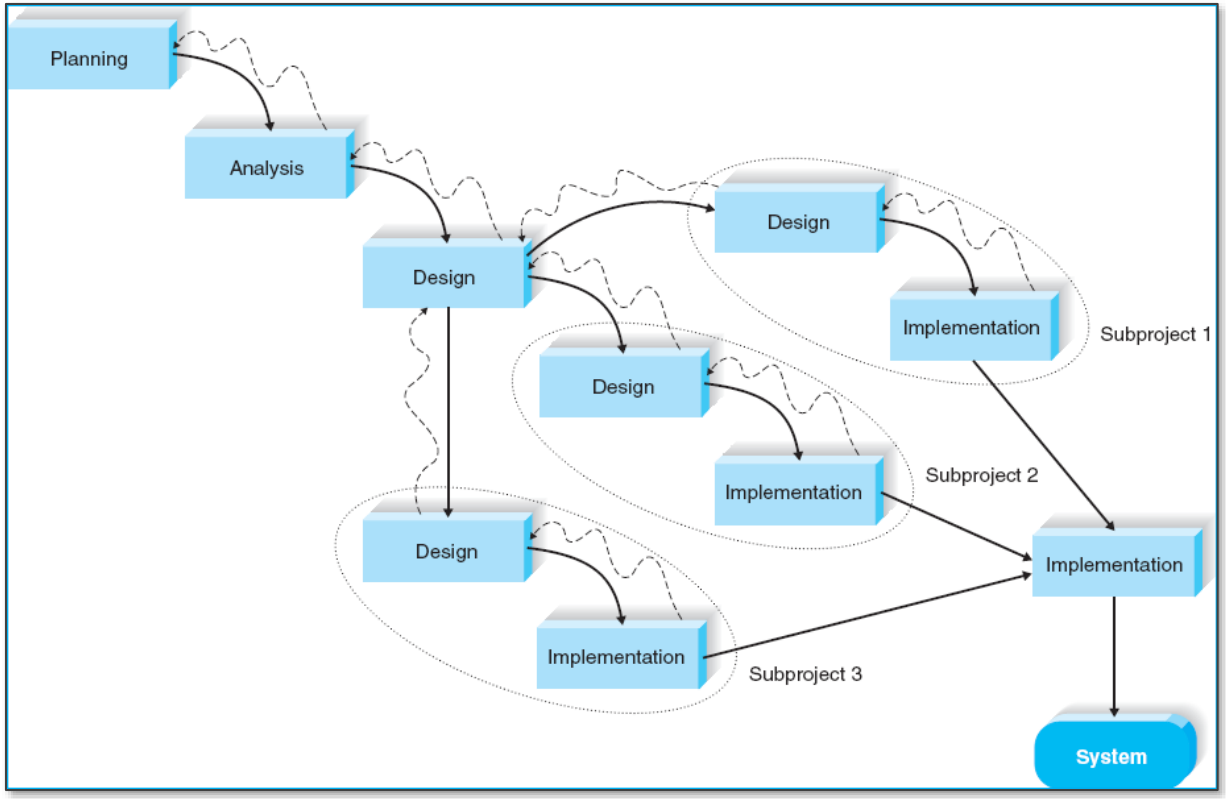
Code craft: the practice of writing excellent code / Pete Goodliffe

V Model

- In the Real World, testing and bug fixing often take up more than half of a project's total time

Code craft: the practice of writing excellent code / Pete Goodliffe

Software Life Development Cycle Models



Parallel Development

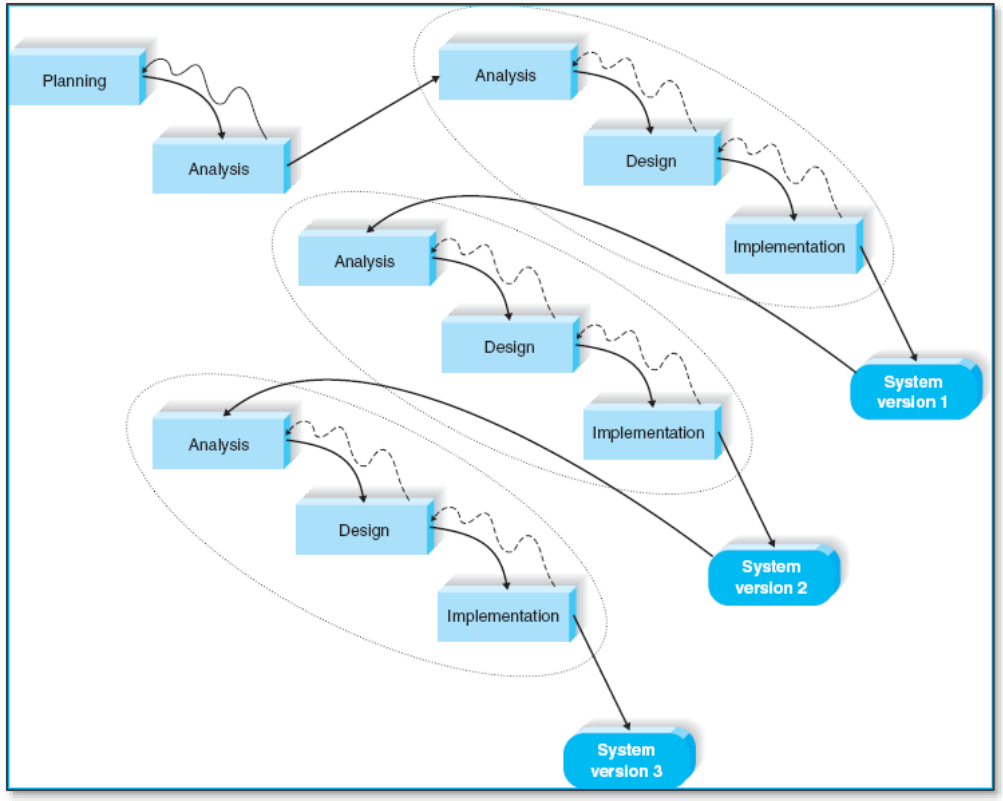
From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Parallel Development

- Advantages
  - It can reduce the schedule time to deliver a system
- Disadvantages
  - Disadvantages for waterfall model applies
  - Sometimes subprojects are not completely independent; design decisions in one subproject may affect the other
  - May require significant integration effort at the end of the project

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Software Development Life Cycle Models



Phased Development

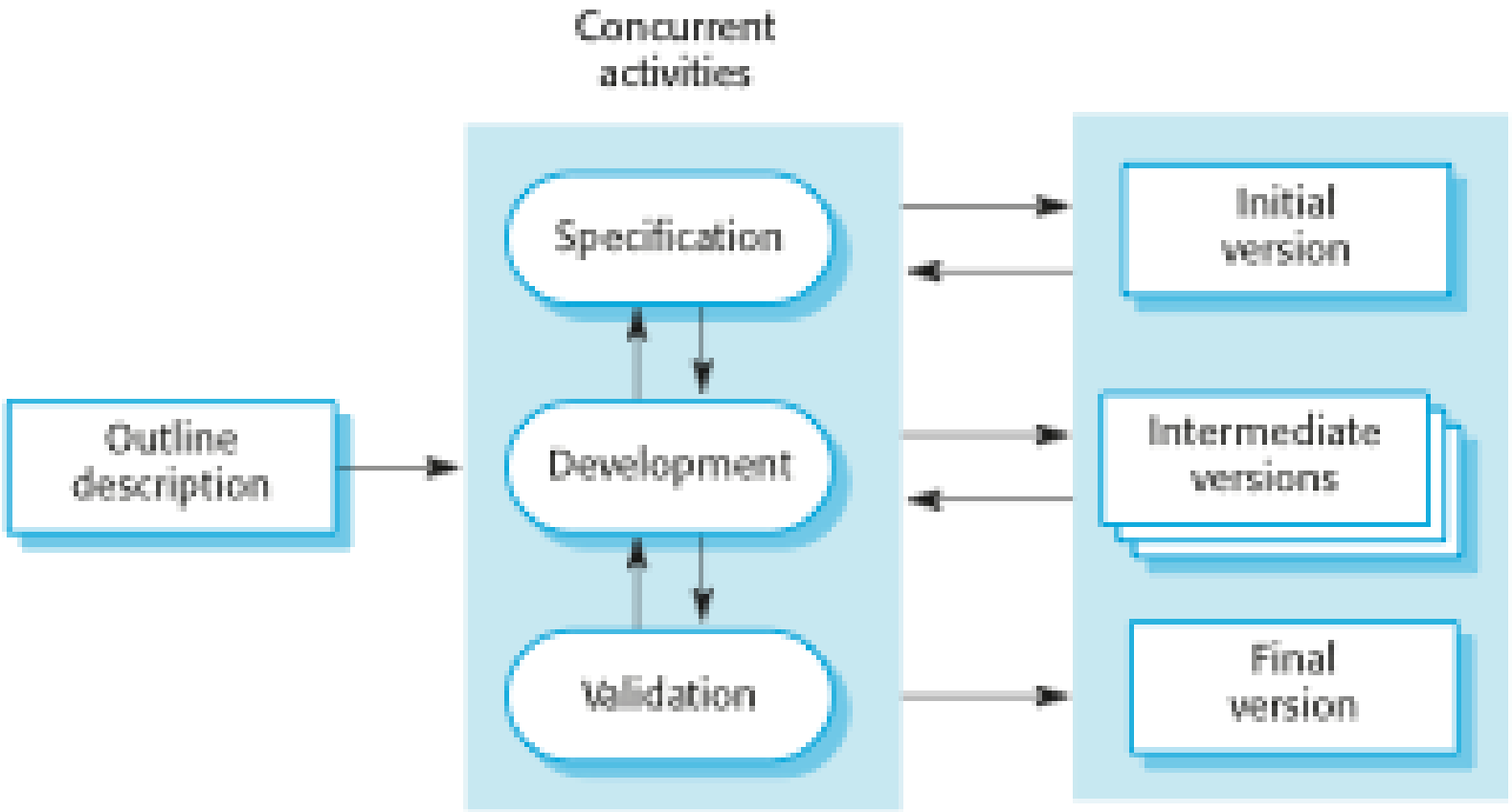
From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Phased Development

- Advantages
  - Quickly gets a useful system into the hands of users
- Disadvantages
  - Users begin to work with incomplete systems – it is important to decide on features of the first version

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Incremental Development



Incremental Development Benefits

- The cost of accommodating changing customer requirements is reduced
  - The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model

Incremental Development Benefits

- It is easier to get customer feedback on the development work that has been done
  - Customers can comment on demonstrations of the software and see how much has been implemented

Incremental Development Benefits

- More rapid delivery and deployment of useful software to the customer is possible
  - Customers are able to use and gain value from the software earlier than is possible with a waterfall process

Incremental Development Problems

- The process is not visible
  - Managers need regular deliverables to measure progress
  - If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system

Incremental Development Problems

- System structure tends to degrade as new increments are added
  - Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure
  - Incorporating further software changes becomes increasingly difficult and costly

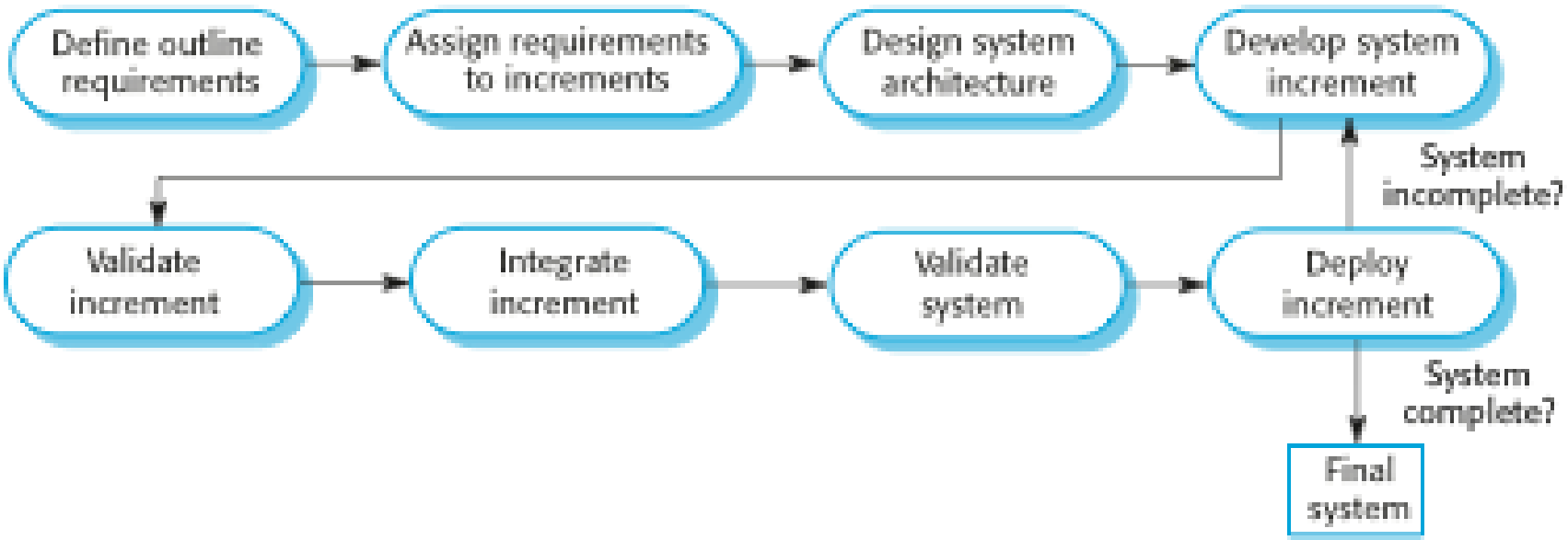
Incremental Development and Delivery

- Incremental development
  - Develop the system in increments and evaluate each increment before proceeding to the development of the next increment;
  - Normal approach used in agile methods;
  - Evaluation done by user/customer proxy

Incremental Development and Delivery

- Incremental delivery
  - Deploy an increment for use by end-users;
  - More realistic evaluation about practical use of software;
  - Difficult to implement for replacement systems as increments have less functionality than the system being replaced

Incremental Delivery



Incremental Delivery Advantages

- Customer value can be delivered with each increment so system functionality is available earlier
- Early increments act as a prototype to help elicit requirements for later increments

Incremental Delivery Advantages

- Lower risk of overall project failure
- The highest priority system services tend to receive the most testing

Incremental Delivery Problems

- Most systems require a set of basic facilities that are used by different parts of the system
  - As requirements are not defined in detail until an increment is to be implemented, it can be hard to identify common facilities that are needed by all increments

Incremental Delivery Problems

- The essence of iterative processes is that the specification is developed in conjunction with the software
  - However, this conflicts with the procurement model of many organizations, where the complete system specification is part of the system development contract

Software Prototyping

- A prototype is an initial version of a system used to demonstrate concepts and try out design options

Software Prototyping

- A prototype can be used in:
  - The requirements engineering process to help with requirements elicitation and validation;
  - In design processes to explore options and develop a UI design;
  - In the testing process to run back-to-back tests

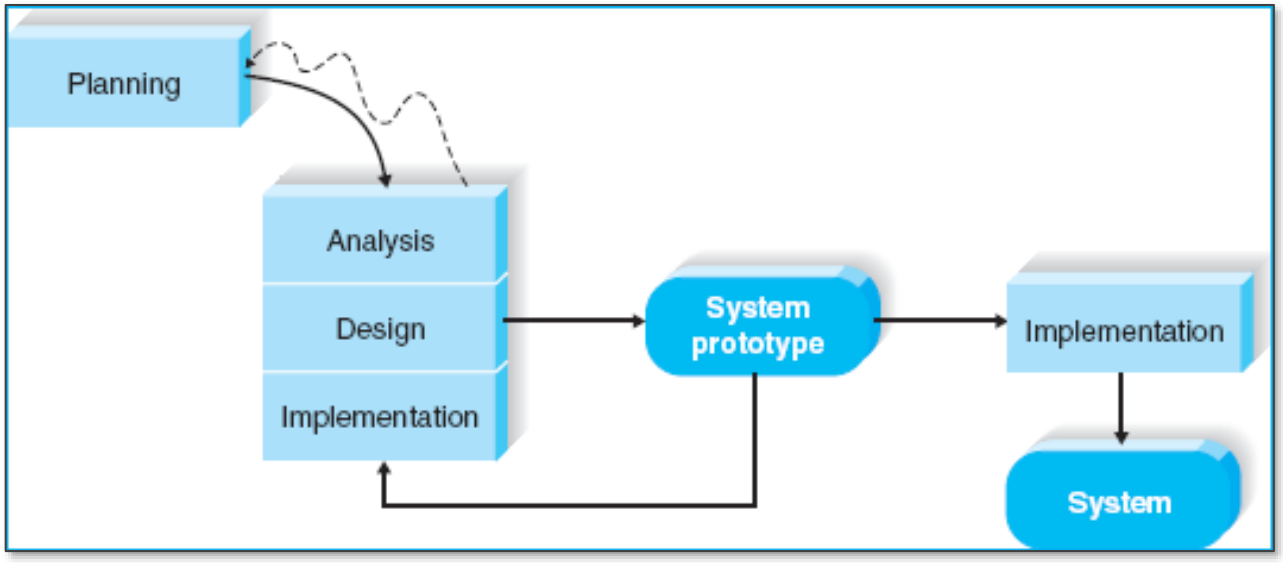
Benefits of Prototyping

- Improved system usability
- A closer match to users' real needs
- Improved design quality
- Improved maintainability
- Reduced development effort

Prototype Development

- May be based on rapid prototyping languages or tools
- May involve leaving out functionality
  - Prototype should focus on areas of the product that are not well-understood;
  - Error checking and recovery may not be included in the prototype;
  - Focus on functional rather than non-functional requirements such as reliability and security

Software Life Cycle Models



System Prototyping

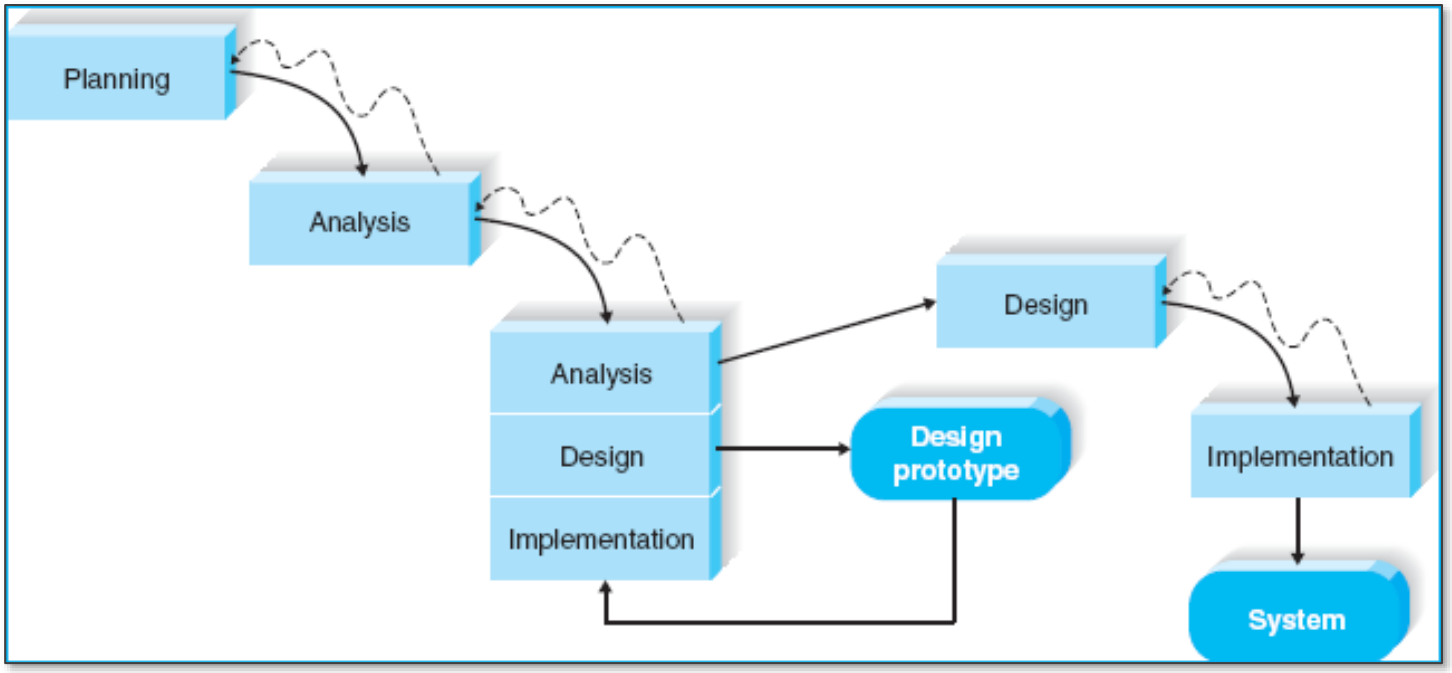
From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

System Prototyping

- Advantages
  - Very quickly provides a system that users can interact
- Disadvantages
  - Fast-paced system releases challenge attempts to conduct careful, methodical analysis
    - Can cause problems in complex system development because fundamental issues and problems are not recognized until into the development process

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Software Life Cycle Models



Throwaway Prototyping

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Throw-away Prototypes

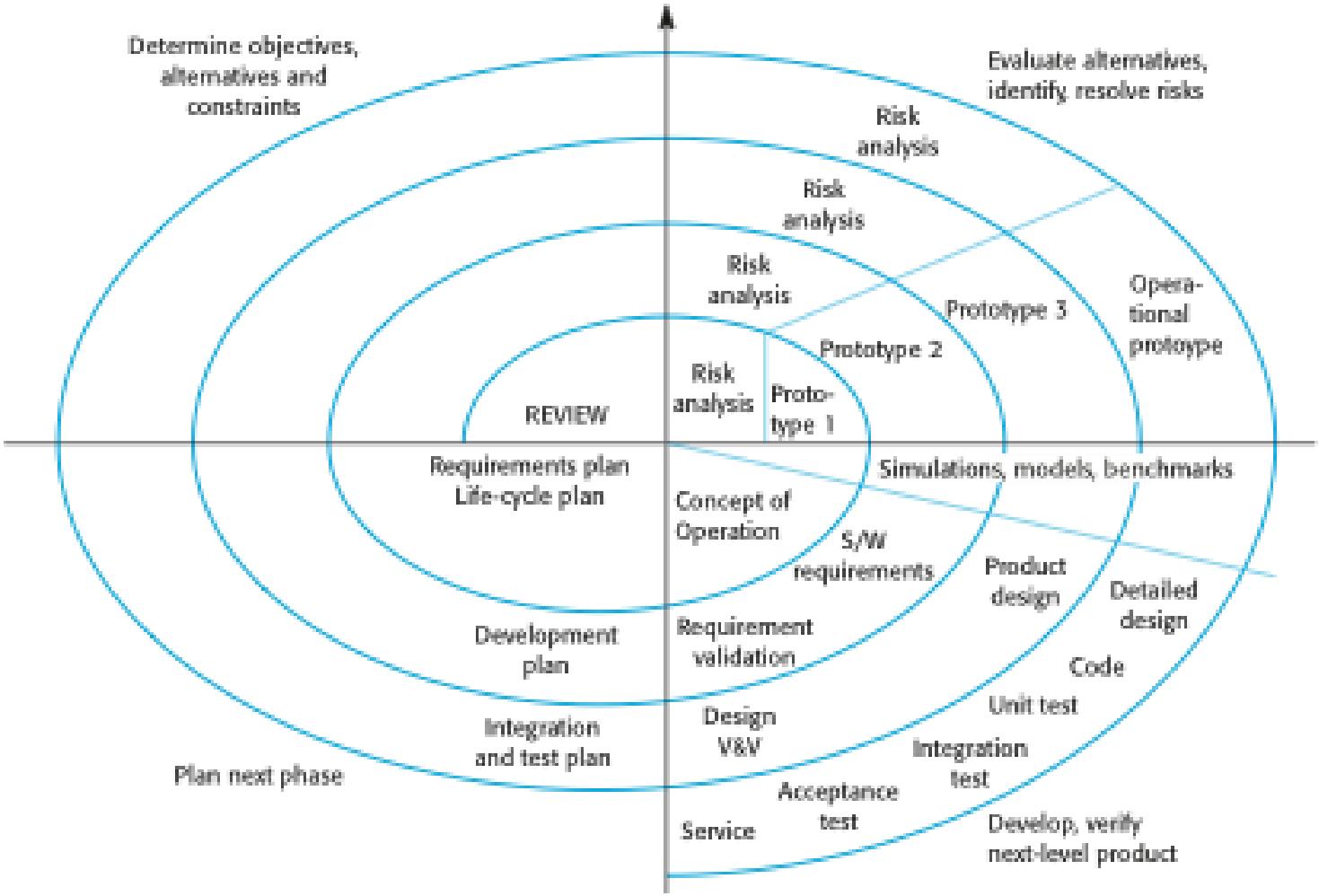
- Prototypes should be discarded after development as they are not a good basis for a production system:
  - It may be impossible to tune the system to meet non-functional requirements;
  - Prototypes are normally undocumented;
  - The prototype structure is usually degraded through rapid change;
  - The prototype probably will not meet normal organizational quality standards

Throwaway Prototyping

- Advantages
  - Balance the benefits of careful analysis and design phases with the advantages of using prototypes
- Disadvantages
  - It may take longer to deliver the final system compared to prototyping-based methodologies

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Boehm's Spiral Model of the Software Process



Boehm's 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

Boehm's Spiral Model

- 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

Spiral Model Sectors

- 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 Usage

- Spiral model has been very influential in helping people think about iteration in software processes and introducing the risk-driven approach to development

Spiral Model Usage

- In practice, however, the model is rarely used as published for practical software development

Object-Oriented Analysis & Design

- Attempt to balance emphasis on data and process
- Uses Unified Modeling Language (UML)

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Object-Oriented Analysis & Design

- Characteristics of OOAD
  - Use-case Driven
    - Use cases are primary modeling tools
  - Architecture Centric
    - Functional, static and dynamic views
  - Iterative and Incremental
    - Continuous testing and refinement

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Object-Oriented Analysis & Design

- Enables
  - To break a complex system into smaller modules
  - To work on the modules individually
  - To piece the modules easily back together to form the system

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Object-Oriented Analysis & Design

- Modularity makes system development easier to grasp, share among members of the project team and to communicate to users
- Enables to create reusable pieces
- Many people believe "object-think" is more natural

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

The Unified Process

- A specific methodology that maps out when and how to use the various UML techniques for object-oriented analysis and design
- "Three amigos"
  - Grady Booch, Ivar Jacobsen and James Rumbaugh

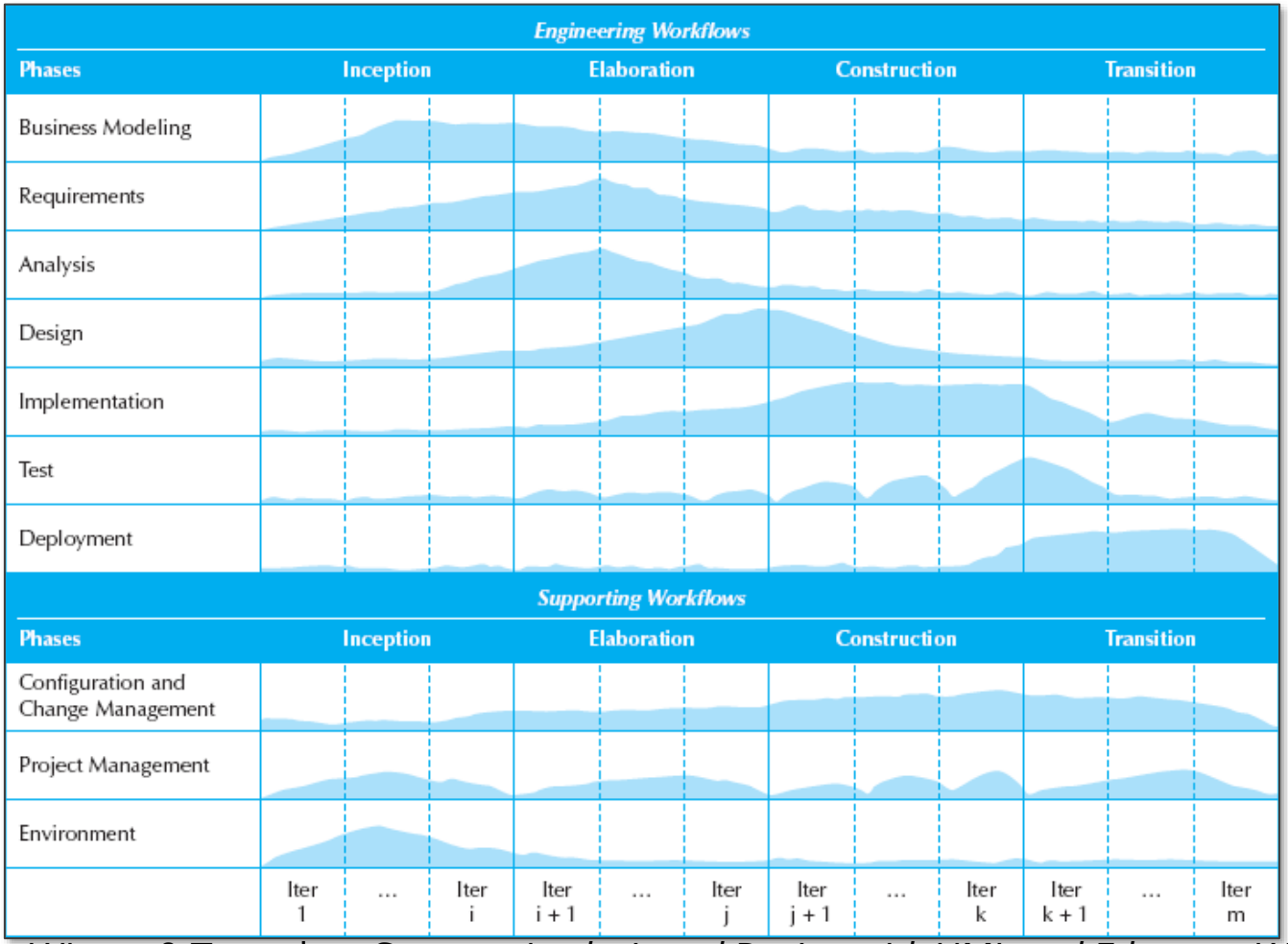
From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

The Unified Process

- A two-dimensional process consisting of phases and flows
  - Phases describe how the system evolves over time
  - Workflows are collections of tasks that occur throughout the lifecycle, but vary in intensity

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

The Unified Process



From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Unified Process Phases

- Inception
  - Similar to planning phase
- Elaboration
  - Analysis and design workflows are the primary focus
- Construction
  - Focuses heavily on programming
- Transition
  - Primary focus is testing and deployment workflows

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Static Workflows in the Rational Unified Process

Workflow	Description
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.

Static Workflows in the Rational Unified Process

Workflow	Description
Testing	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and change management	This supporting workflow managed changes to the system
Project management	This supporting workflow manages the system development
Environment	This workflow is concerned with making appropriate software tools available to the software development team.

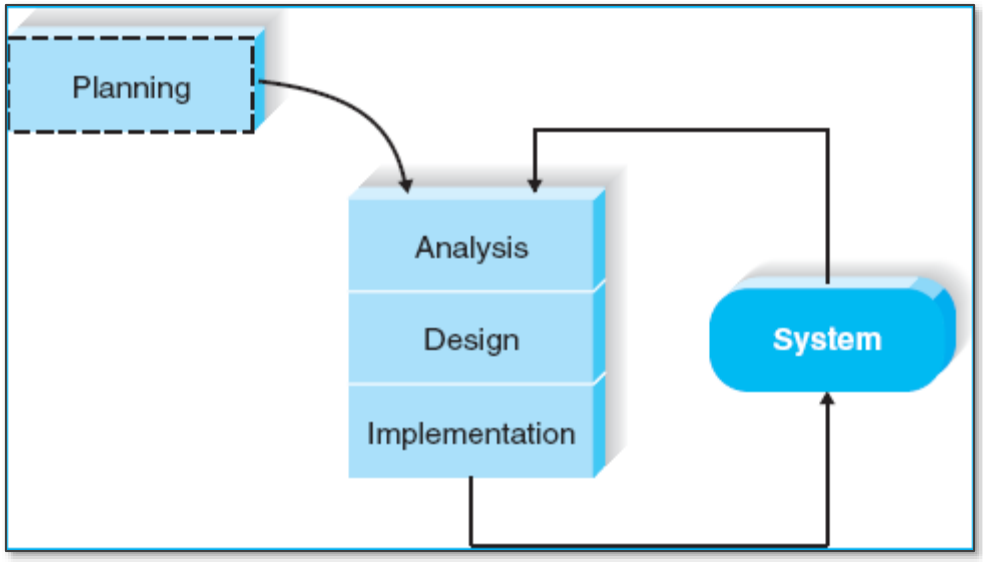
Supporting Workflows

- Project management
- Configuration and change management
- Environment
- Operations and support*
- Infrastructure management*

* Part of the *enhanced* unified process

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Software Life Cycle Models



Extreme Programming

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Extreme Programming

- Key principles it uses
  - Continuous testing
  - Simple coding performed by pairs of developers
  - Close interaction with end users

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Extreme Programming

- Advantages
  - Good for small projects
- Disadvantages
  - Not suitable for large projects or teams that are not "jelled"

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*

Selecting the Right Methodology

Usefulness for	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	Extreme Programming
Unclear user requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
Unfamiliar technology	Poor	Poor	Good	Poor	Excellent	Poor
Complex systems	Good	Good	Good	Poor	Excellent	Poor
Reliable systems	Good	Good	Good	Poor	Excellent	Good
Short time schedule	Poor	Good	Excellent	Excellent	Good	Excellent
Schedule visibility	Poor	Poor	Excellent	Excellent	Good	Good

From: Dennis, Wixom & Tegarden, *Systems Analysis and Design with UML, 3rd Ed, 2009, Wiley*