

Software Development Process Models

Lesson 3

Notes compiled by
Melody Angelique C.
Rivera for CCS 6

References:

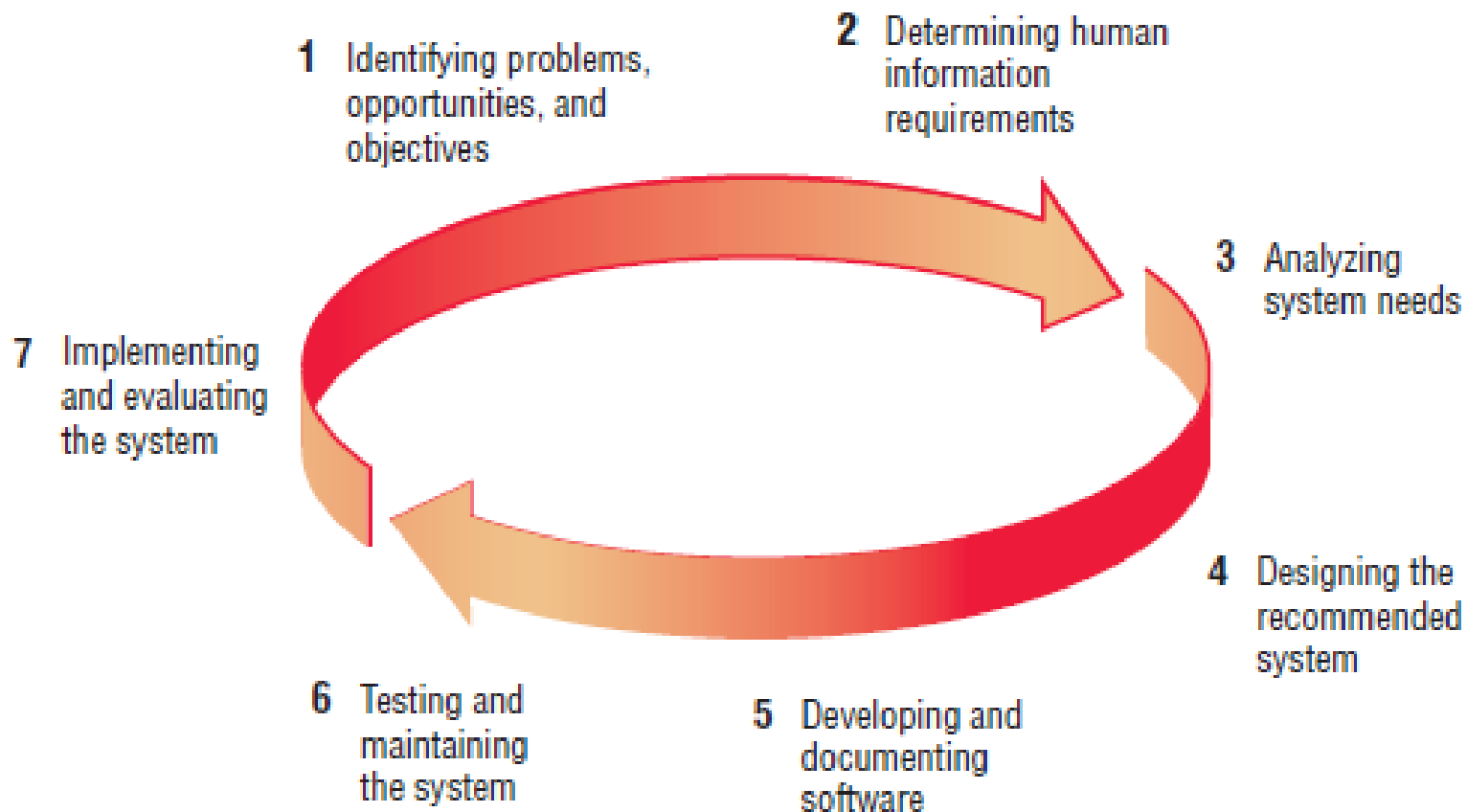
Roger S. Pressman, Software
Engineering 7th Edition

Ian Sommerville, Software
Engineering 9th Edition

Systems Development Life Cycle (SDLC)

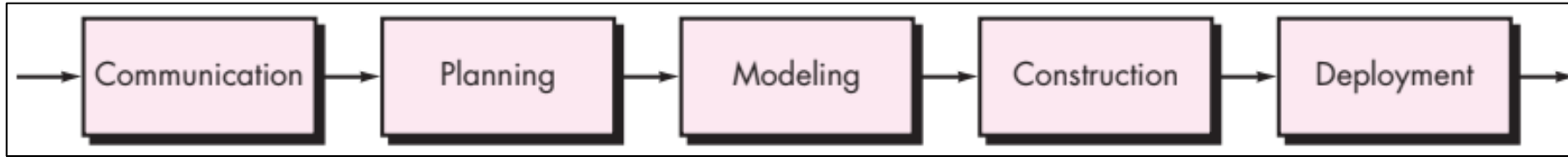
Process Flows

7 Phases of SDLC

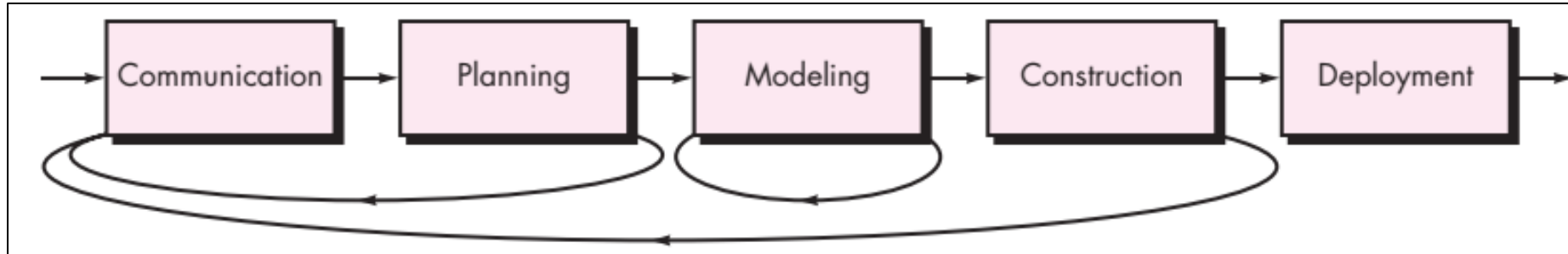


Kendall & Kendall,
Systems Analysis and Design, 8th Ed.

5 Phases of SDLC (1)

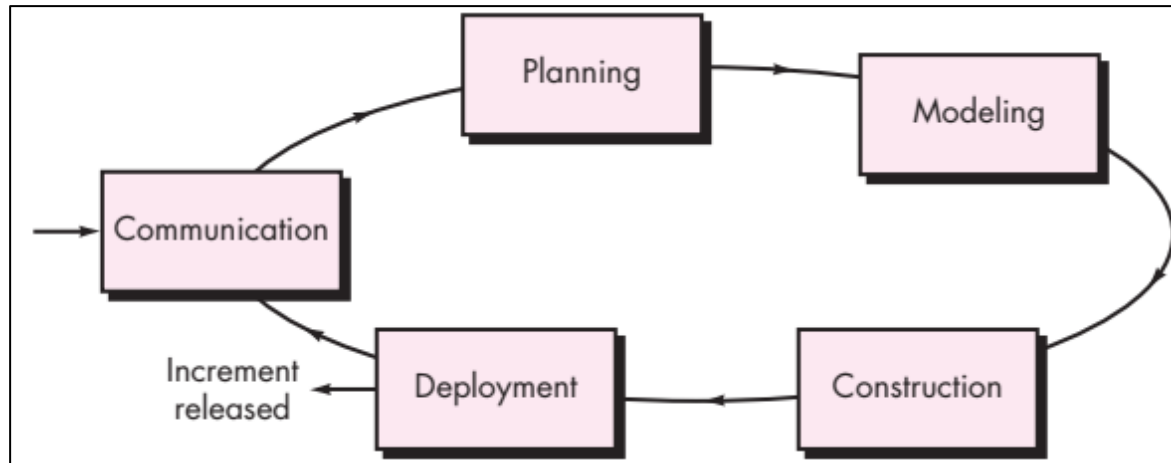


Linear Process Flow

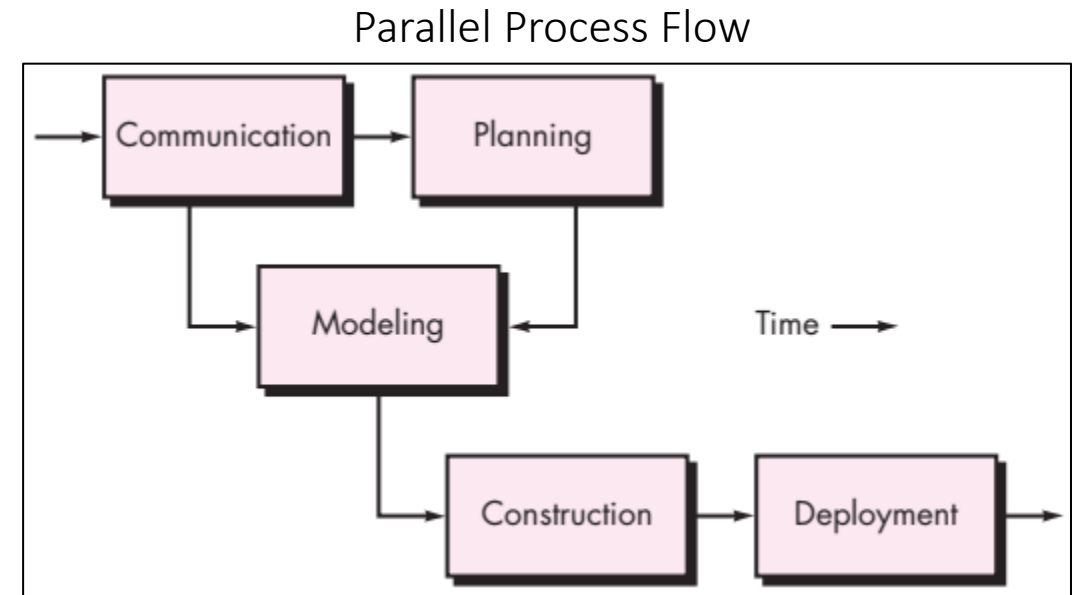


Iterative Process Flow

5 Phases of SDLC (2)



Evolutionary Process Flow

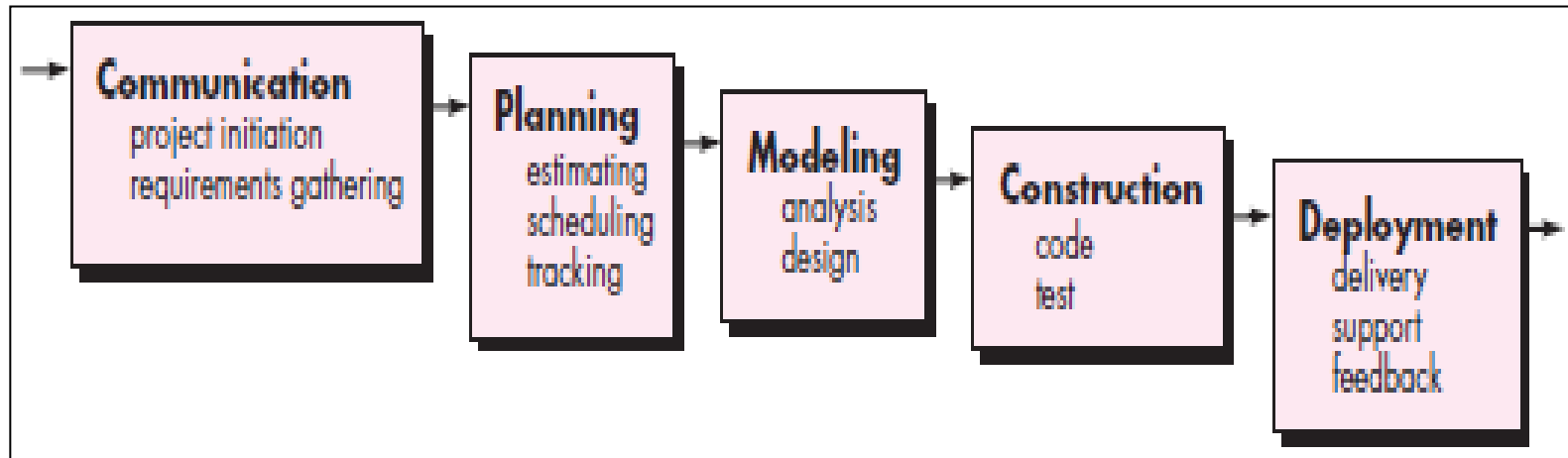


Sequential/Prescriptive Process Models

Waterfall Model

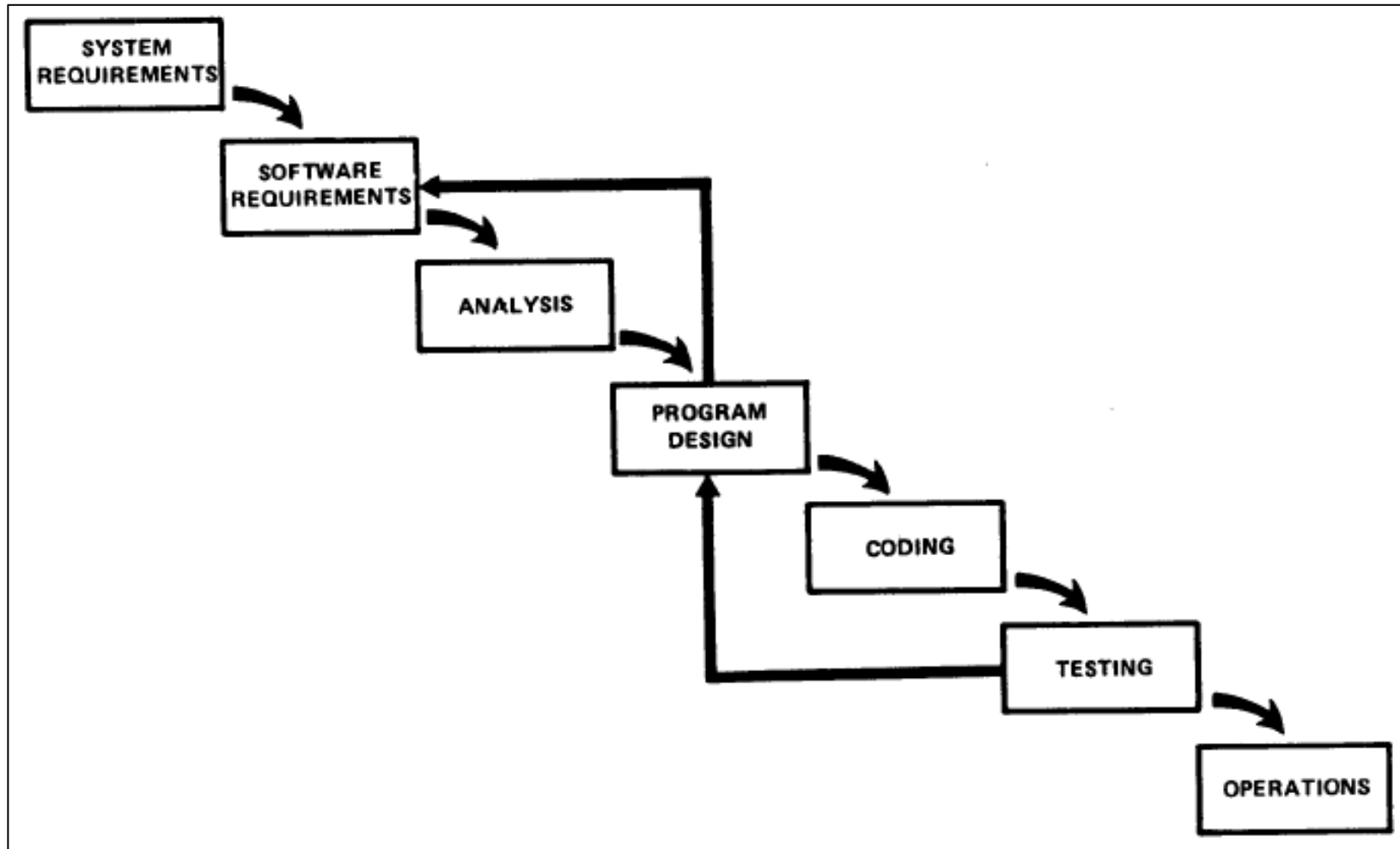
V-Model

Waterfall Model



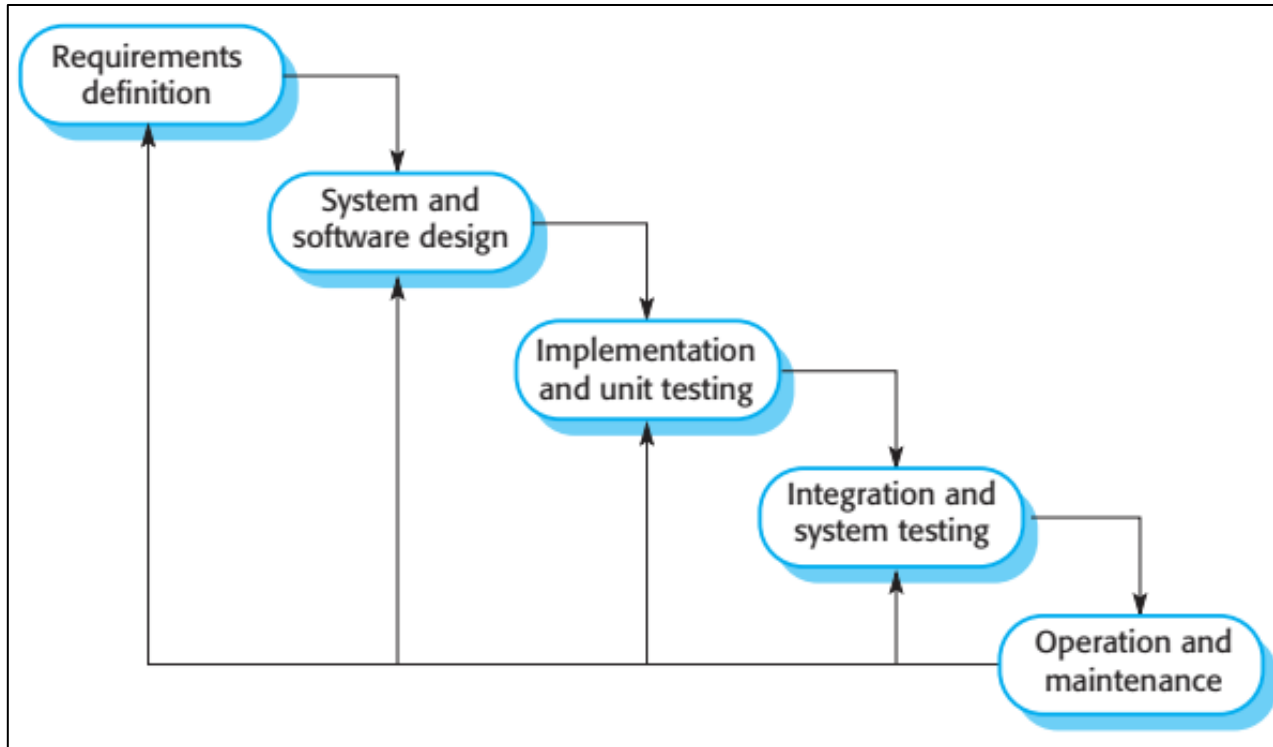
R. Pressman,
Software Engineering,
7th Ed.

- also referred to as a **linear-sequential life cycle model**
- sometimes called the *classic life cycle*, suggests a systematic sequential approach to software development
- begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment, culminating in ongoing support of the completed software
- each phase must be completed before the next phase can begin and there is no overlapping in the phases
- the outcome of one phase acts as the input for the next phase sequentially



The original
Waterfall model
(Royce, 1970)

By Winston W. Royce - IEEE WESCON, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=88210515>

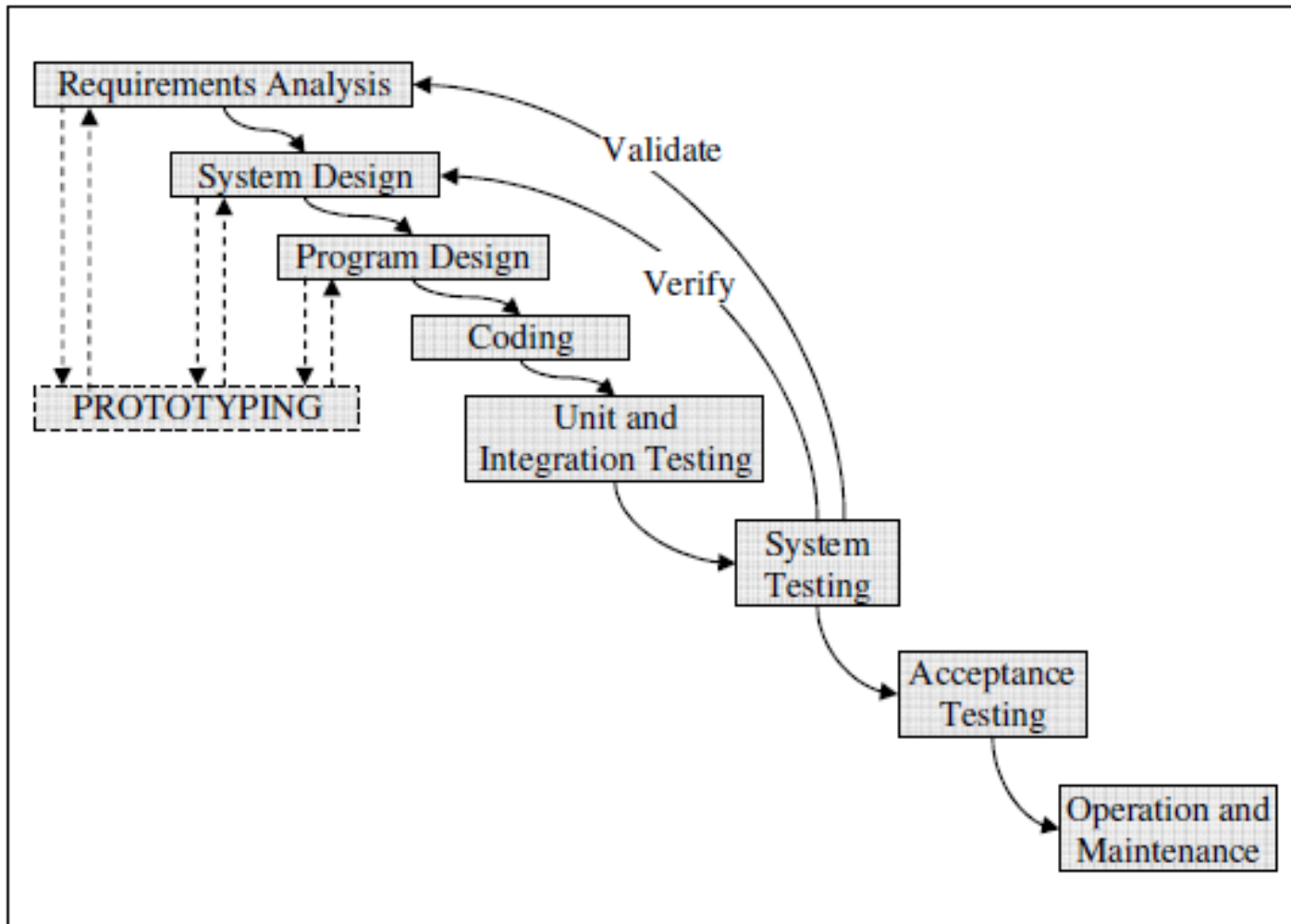


I. Sommerville, Software Engineering, 9th Ed.

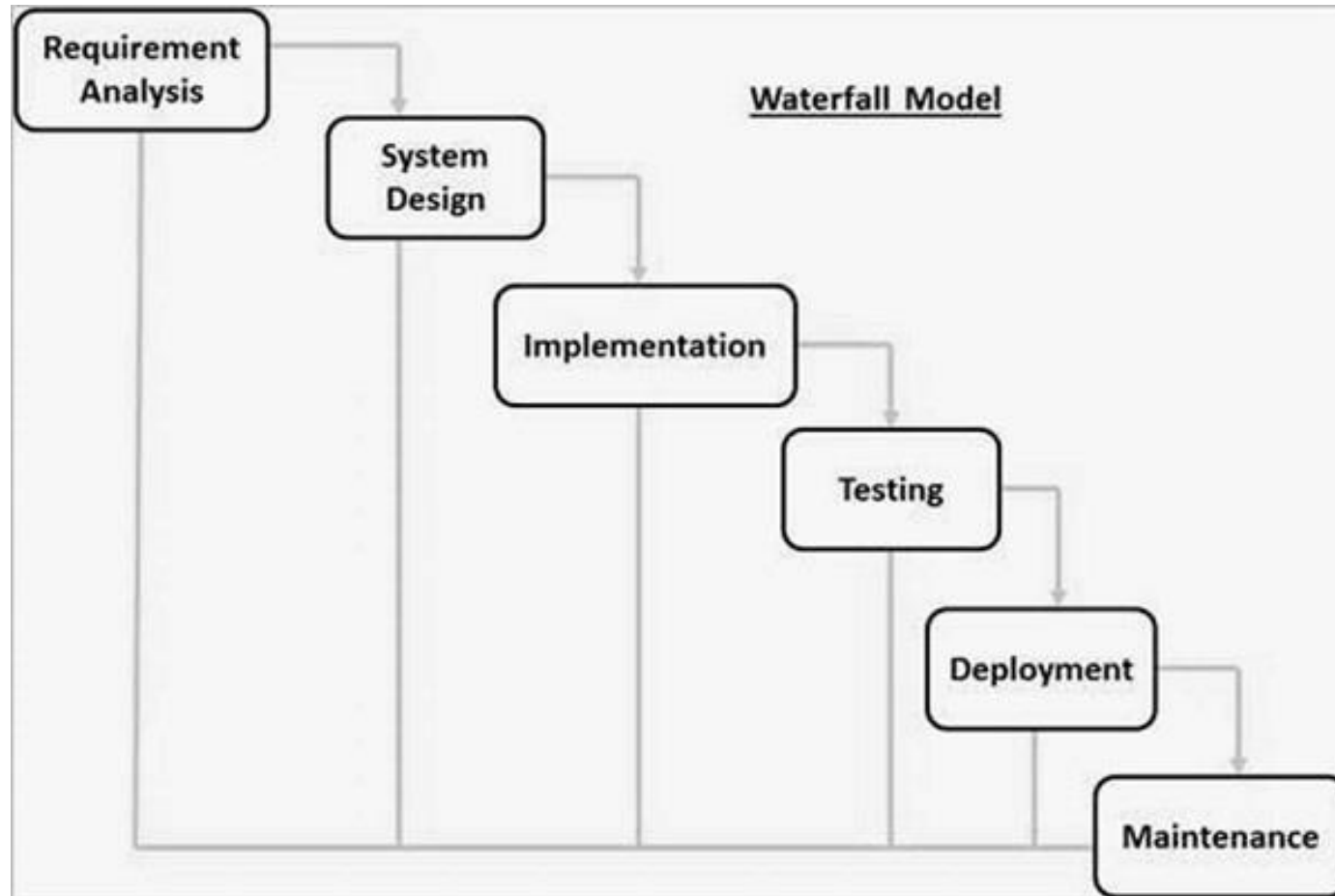
Iterative development and agile methods are better in situations where informal team communication is possible and software requirements change

The waterfall model is only appropriate for some types of system:

- **Embedded systems**
 - where the software has to interface with hardware systems
- **Critical systems**
 - where there is a need for extensive safety and security analysis of the software specification and design
- **Large software systems**
 - that are part of broader engineering systems developed by several partner companies



Waterfall Model with
Validation and Verification



https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Phases in the Waterfall Model

https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document

System Design – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Phases in the Waterfall Model (cont.)

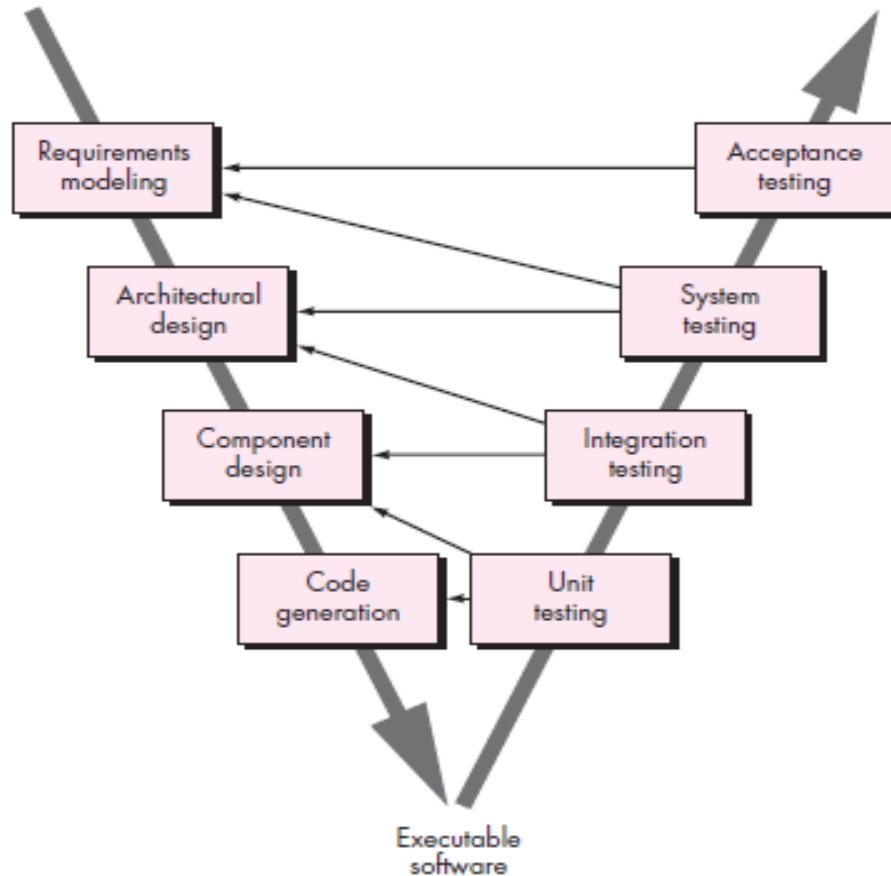
https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

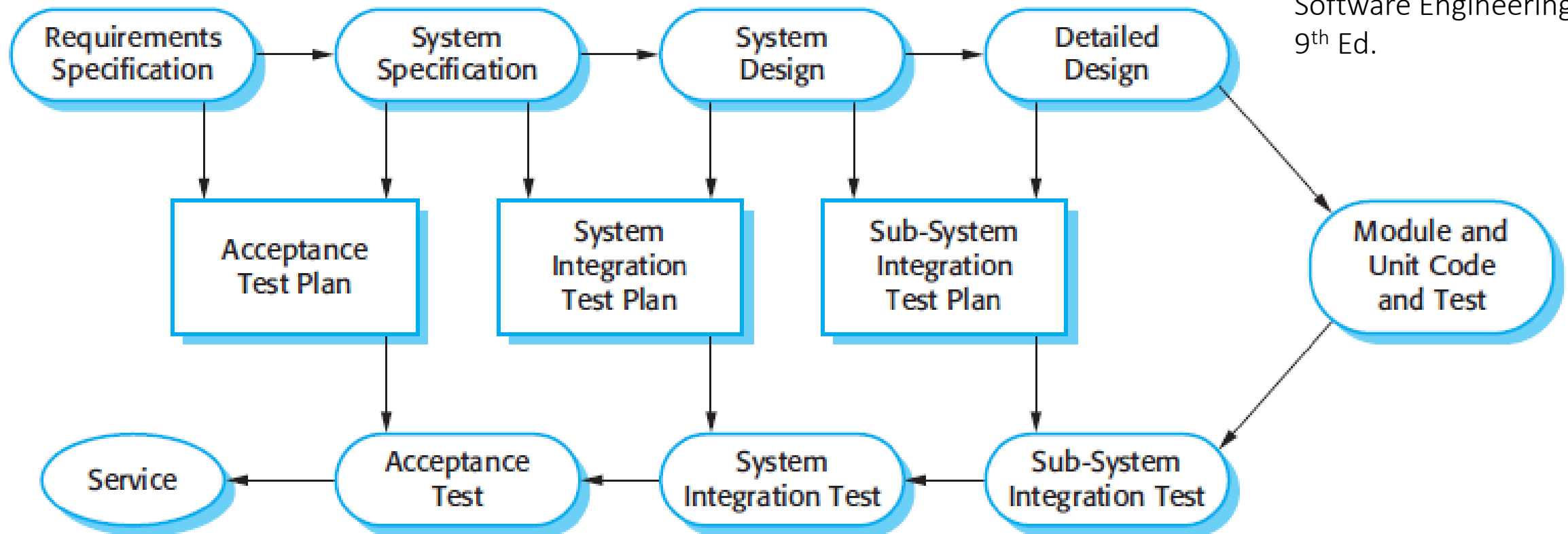
V-Model (1)

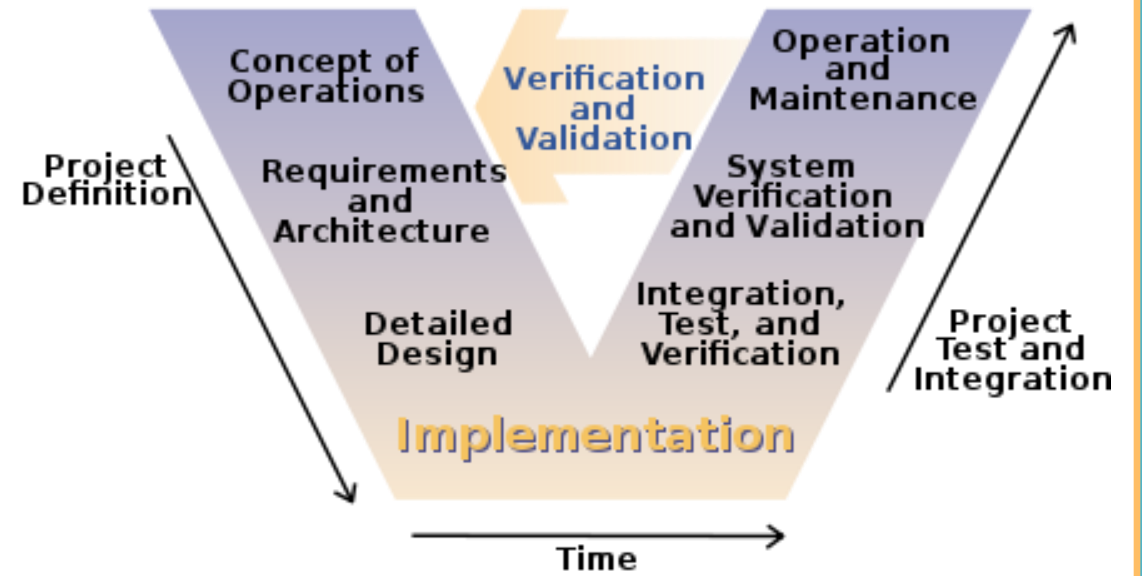
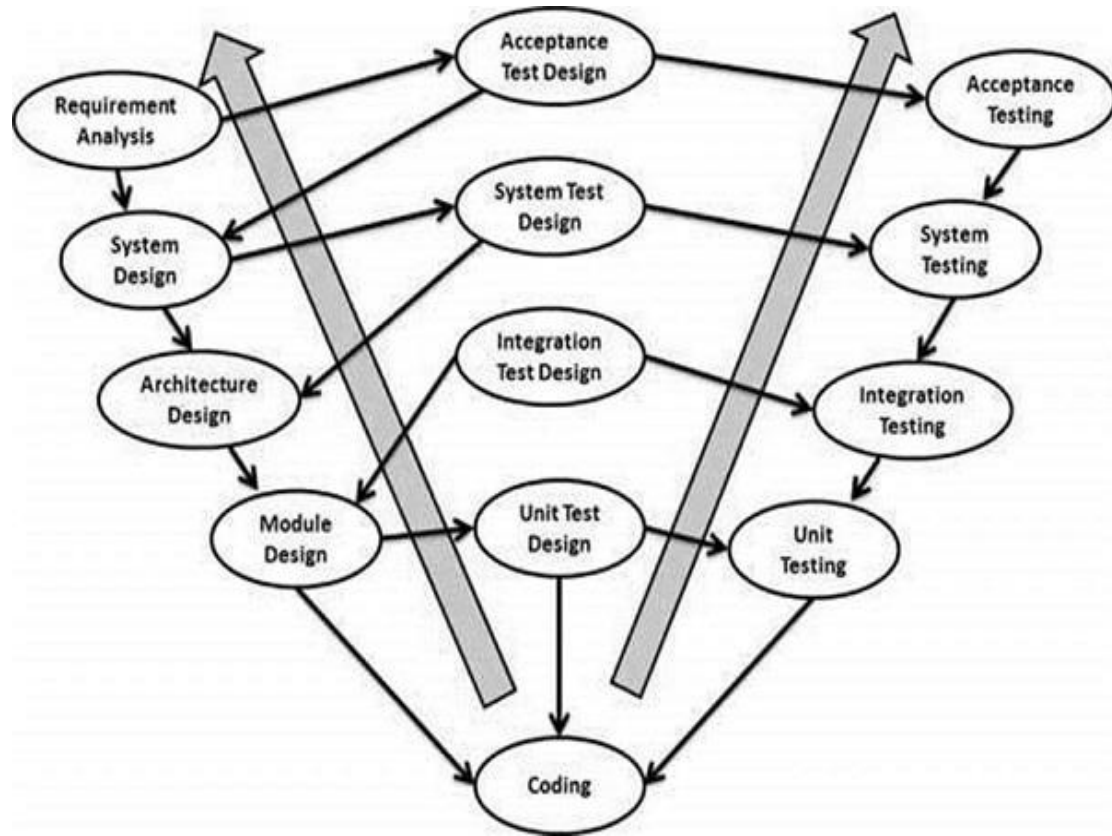


- Illustrates how verification and validation actions are associated with earlier engineering actions
- **Depicts the relationship of quality** assurance actions to the actions associated with communication, modeling, and early construction activities
- As a software team moves down the left side of the V, basic problem requirements are refined into progressively more detailed and technical representations of the problem and its solution
- Once code has been generated, the team moves up the right side of the V, essentially performing a series of tests (quality assurance actions) that validate each of the models created as the team moved down the left side

V-Model (2)

I. Sommerville,
Software Engineering,
9th Ed.





Evolutionary/Iterative/ Incremental Process Models

Incremental Model

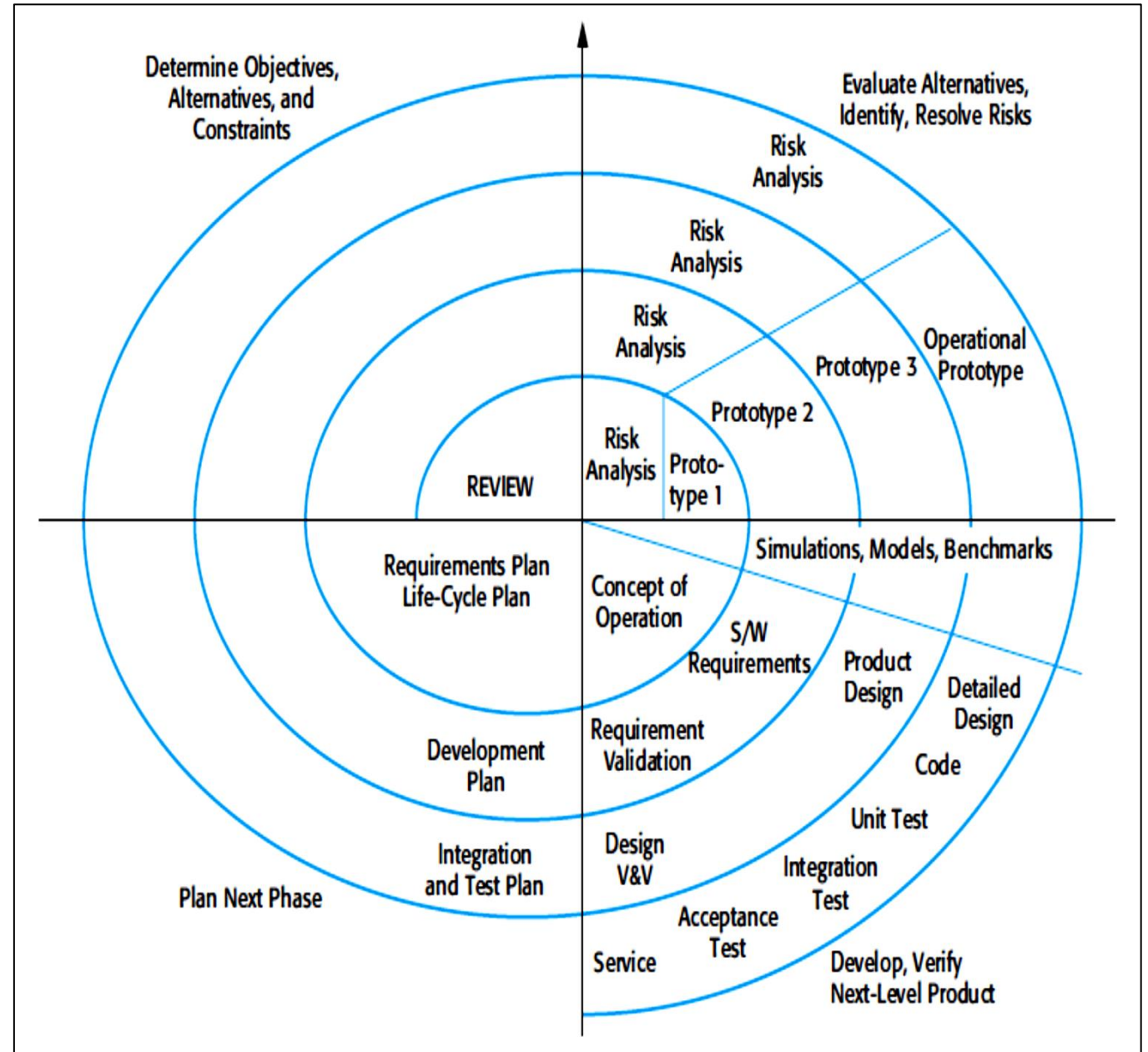
Spiral Model

Prototyping

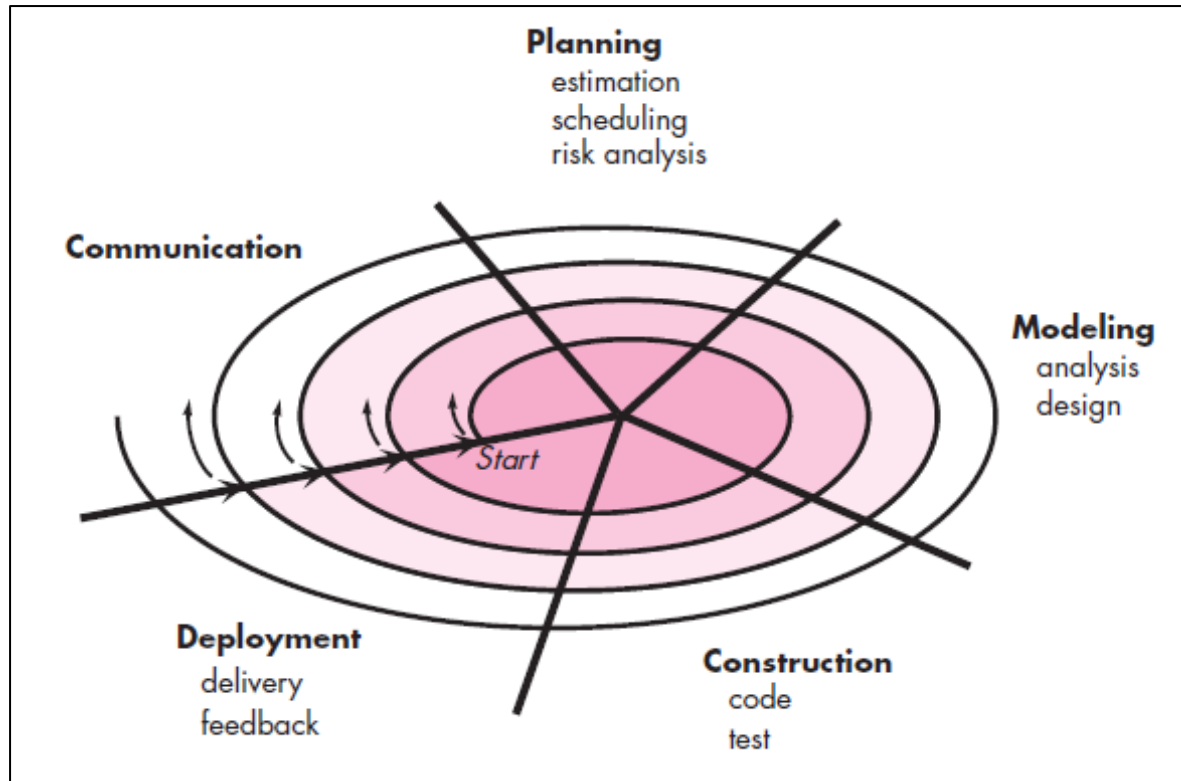
Spiral model

(Boehm, 1988)

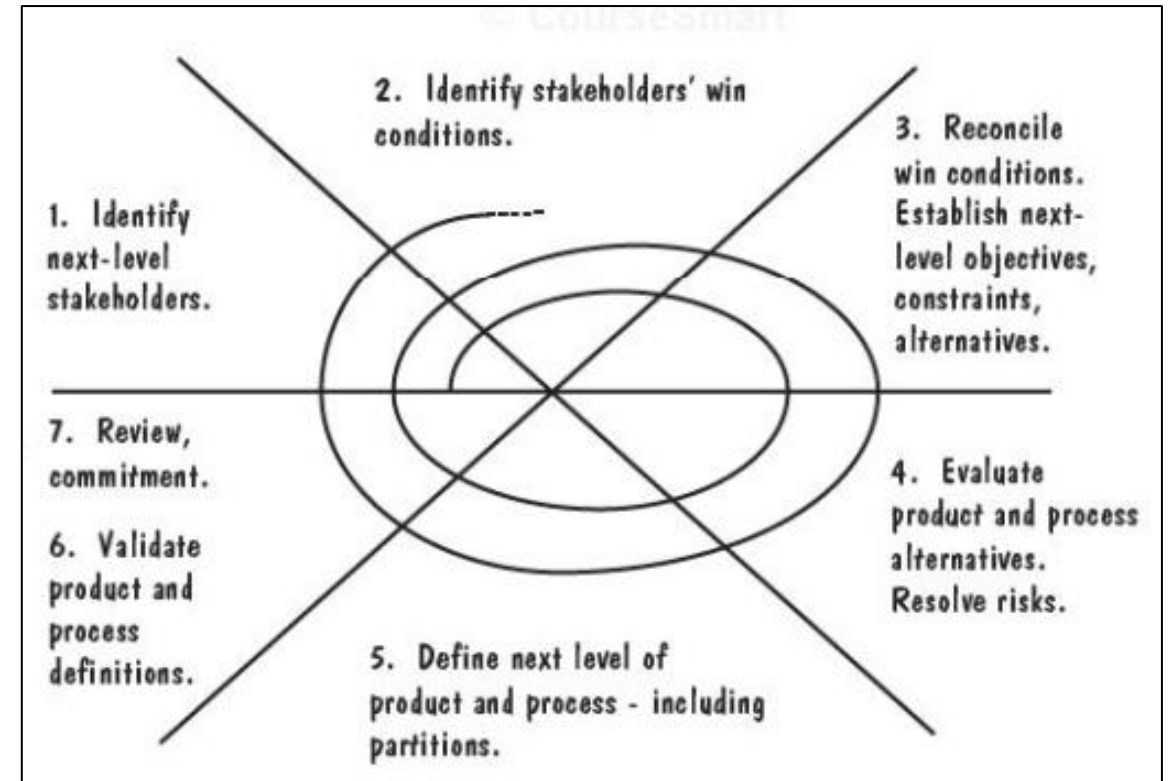
- A software project repeatedly passes through these phases in iterations called Spirals
- Combines change avoidance with change tolerance
- Assumes that changes are a result of project risks and includes explicit risk management activities to reduce these risks
- **Four phases:**
 - Objective setting
 - Risk assessment and reduction
 - Development and validation
 - Planning



Spiral Model (cont.)



R. Pressman, Software Engineering 7th Ed.



Pfleeger and Atlee, Software Engineering 4th Ed.

Spiral Model (cont.)

Phases

1. Objective setting

- Specific objectives for that phase of the project are defined
- Constraints on the process and the product are identified and a detailed management plan is drawn up
- Project risks are identified
- Alternative strategies, depending on these risks, may be planned

2. Risk assessment and reduction

- For each of the identified project risks, a detailed analysis is carried out
- Steps are taken to reduce the risk
- For example, if there is a risk that the requirements are inappropriate, a *prototype system* may be developed

Spiral Model Phases (cont.)

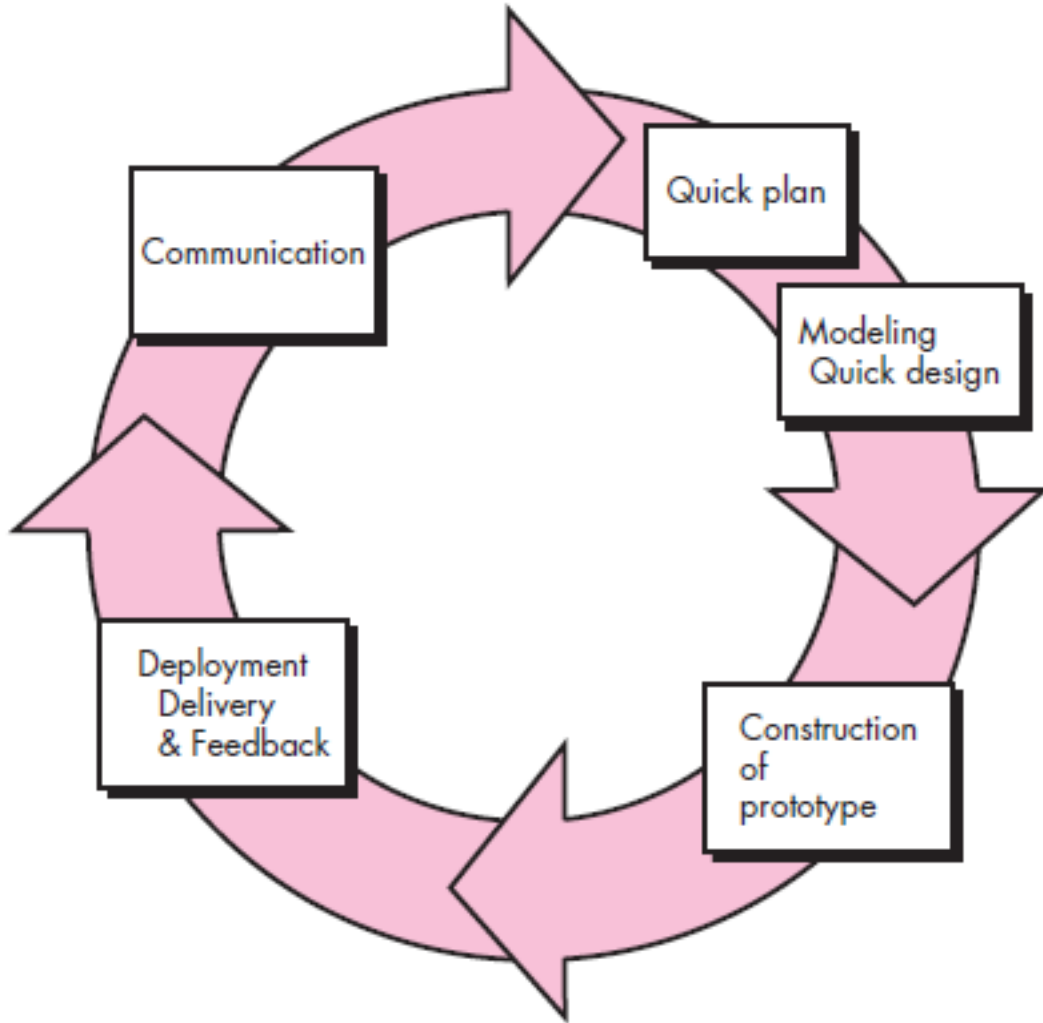
3. Development and validation

- After risk evaluation, a *development model* for the system is chosen
- For example
 - ❖ *throwaway prototyping* may be the best development approach if user interface risks are dominant
 - ❖ if safety risks are the main consideration, development based on *formal transformations* may be the most appropriate process, and so on
 - ❖ If the main identified risk is sub-system integration, the *waterfall model* may be the best development model to use

4. Planning

- The project is reviewed and a decision made whether to continue with a further loop of the spiral
- If it is decided to continue, plans are drawn up for the next phase of the project

Prototyping



- building software application prototypes which displays the functionality of the product under development, but may not actually hold the exact logic of the original software
- enables to understand customer requirements at an early stage of development
- helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development

Prototyping (cont.)

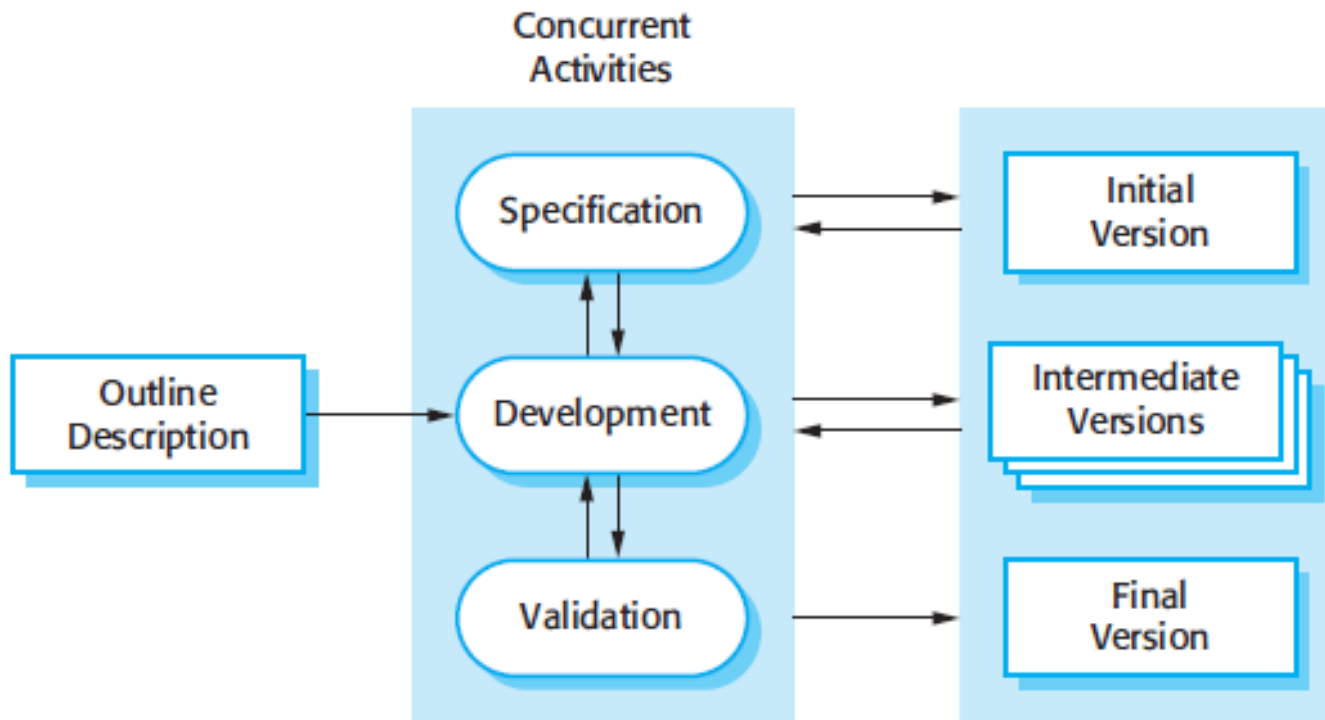
Types

- Throwingaway Prototyping
 - also called as rapid or close ended prototyping
 - uses very little efforts with minimum requirement analysis to build a prototype
 - once the actual requirements are understood, the prototype is discarded and the actual system is developed with a much clear understanding of user requirements
- Evolutionary Prototyping
 - also called as breadboard prototyping
 - based on building actual functional prototypes with minimal functionality in the beginning
 - the prototype developed forms the heart of the future prototypes on top of which the entire system is built
 - by using evolutionary prototyping, the well-understood requirements are included in the prototype and the requirements are added as and when they are understood

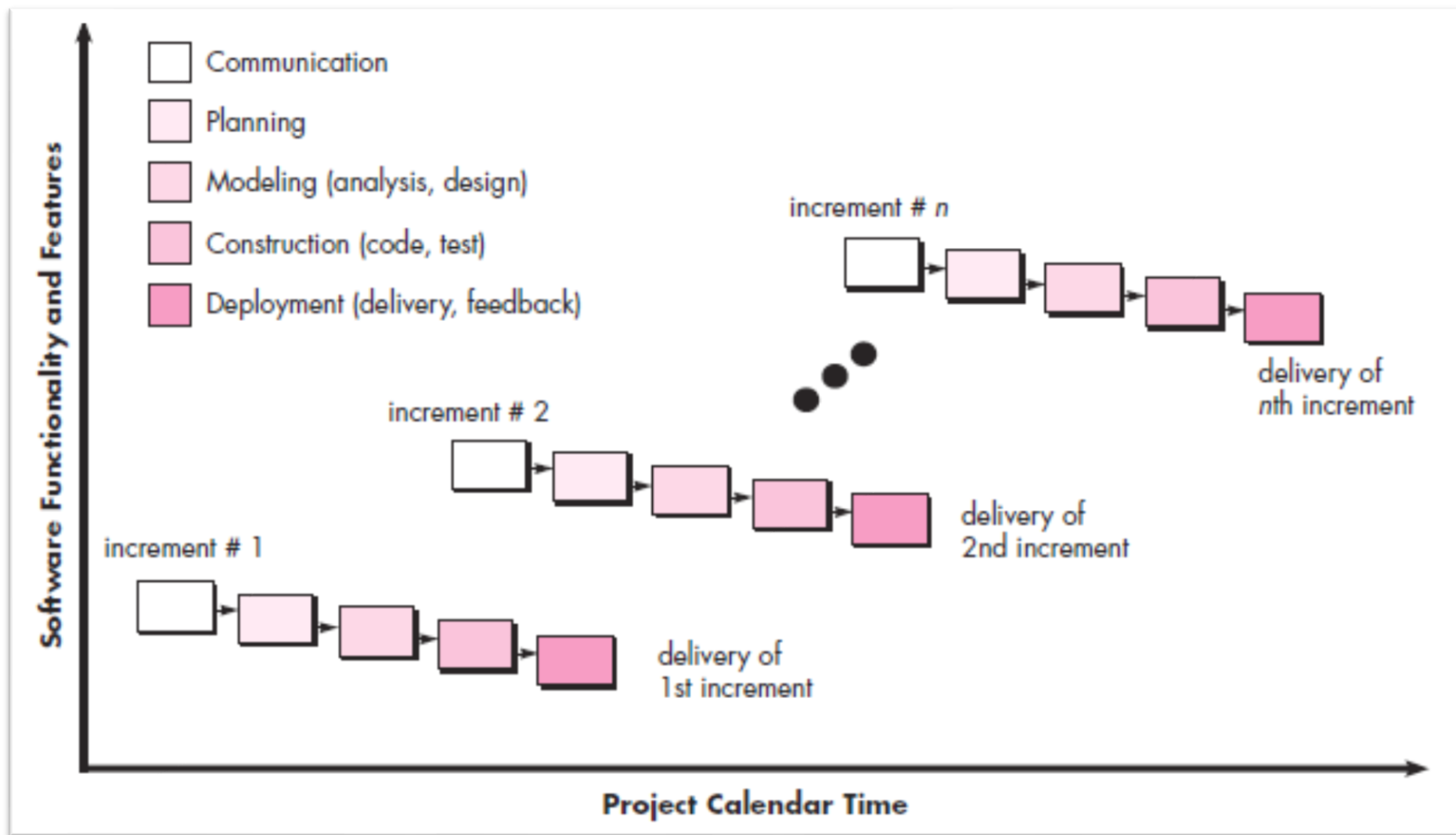
Prototyping (cont.)

- Incremental Prototyping
 - refers to building multiple functional prototypes of the various sub-systems then integrating all the available prototypes to form a complete system
- Extreme Prototyping
 - used in the web development domain
 - consists of three sequential phases
 1. a basic prototype with all the existing pages is presented in the HTML format
 2. the data processing is simulated using a prototype services layer
 3. the services are implemented and integrated to the final prototype
 - This process is used to draw attention to the second phase of the process, where a fully functional UI is developed with very little regard to the actual services

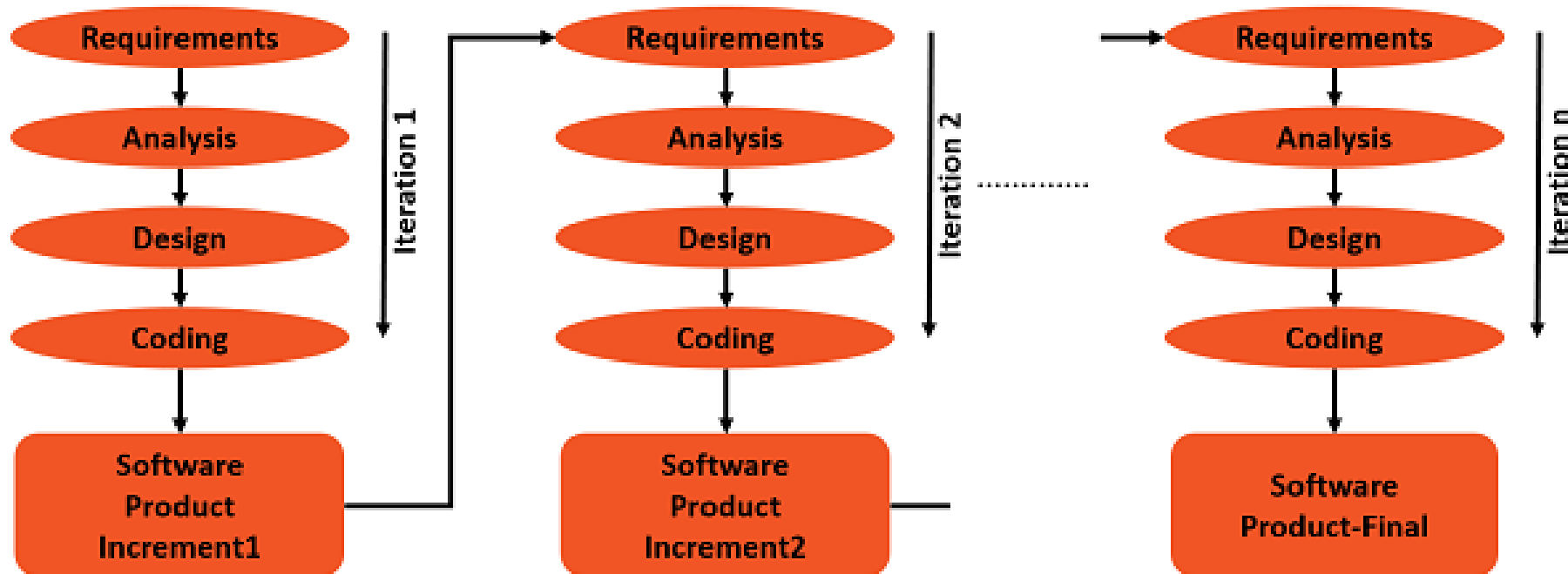
Incremental Model



- based on the idea of developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been developed
- specification, development, and validation activities are interleaved rather than separate, with rapid feedback across activities
- a fundamental part of agile approaches
- better than a waterfall approach for most business, e-commerce, and personal systems
- it is cheaper and easier to make changes in the software as it is being developed



Iterative and Incremental Model



https://www.tutorialspoint.com/adaptive_software_development/sdlc_iterative_incremental_model.htm

The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system (https://en.wikipedia.org/wiki/Iterative_and_incremental_development)

Reading Assignments

What are the advantages and disadvantages of the following software development models?

1. Waterfall Model
2. V-Model
3. Spiral Model
4. Prototyping
5. Incremental
6. Incremental/Iterative Model

End of presentation