

Introduction of Software Engineering

Chapter 3:

Software Models

VŨ THỊ TRÀ

©2018, Danang University of Education

Process Models

1. What is it?

- ✓ A process model provides a specific roadmap for software engineering work. It defines
 - the flow of all activities, actions and tasks,
 - the degree of iteration,
 - the work products,
 - the organization of the work

2. Who does it?

- ✓ Software engineer and their managers adapt the process to their needs and then follow it.
- ✓ Those who have requested the software have a role to play in the process of defining, building, and testing it.

Process Models

3. Why is it important?

- ✓ process provides stability, control, and organization to an activity that can, if left uncontrolled, become quite chaotic.
- ✓ a modern software engineering approach must be "agile."

4. What are steps?

- ✓ The process model provides you with the steps needed to perform disciplined software engineering work.

Process Models

5. What are the work product?

- ✓ the work product is a customized description of the activities and tasks defined by the process.

6. How do I ensure that I've done it right?

- ✓ a number of software process assessment mechanisms that enable organizations to determine the "maturity" of their software process.
- ✓ the quality, timeliness, and long-term viability of the product you build are the best indicators of the efficacy of the process.

Process Models

- ▶ Process Models were originally proposed to bring order to the chaos of software developments.
- *The purpose of process models is to try to reduce the chaos present in developing new software products.*

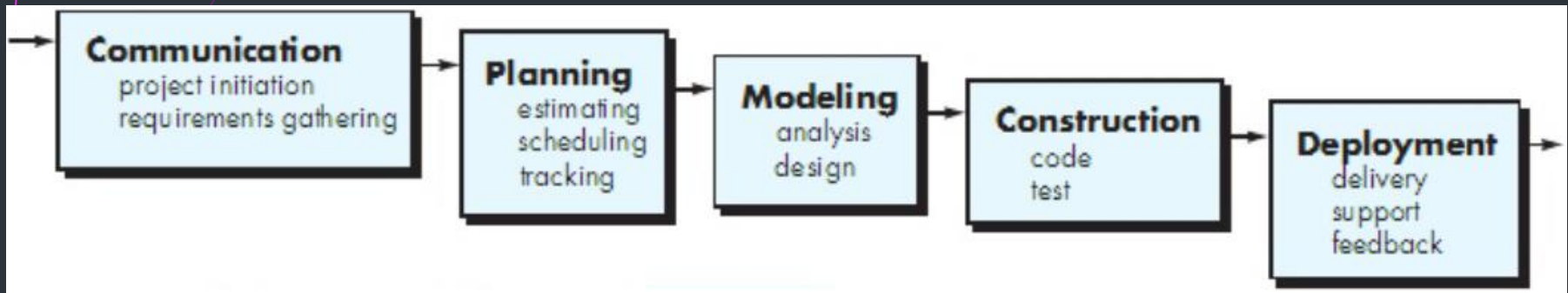
CONTENTS

- **Prescriptive Process Models**
- **Specialized Process Models**
- **The Unified Process Model**
- **Personal and Team Process Models**

Prescriptive Process Models

- ▶ A process model defines
 - ✓ *a prescribed set of process elements*
 - ▶ activities, actions, tasks
 - ▶ work products
 - ▶ quality assurance
 - ▶ change control mechanisms
 - ✓ *a predictable process work flow*
- ▶ Prescriptive Process Models includes
 - ✓ The Waterfall Model
 - ✓ Incremental Process Models
 - ✓ Evolutionary Process Models
 - ✓ Concurrent Models

The Waterfall Model



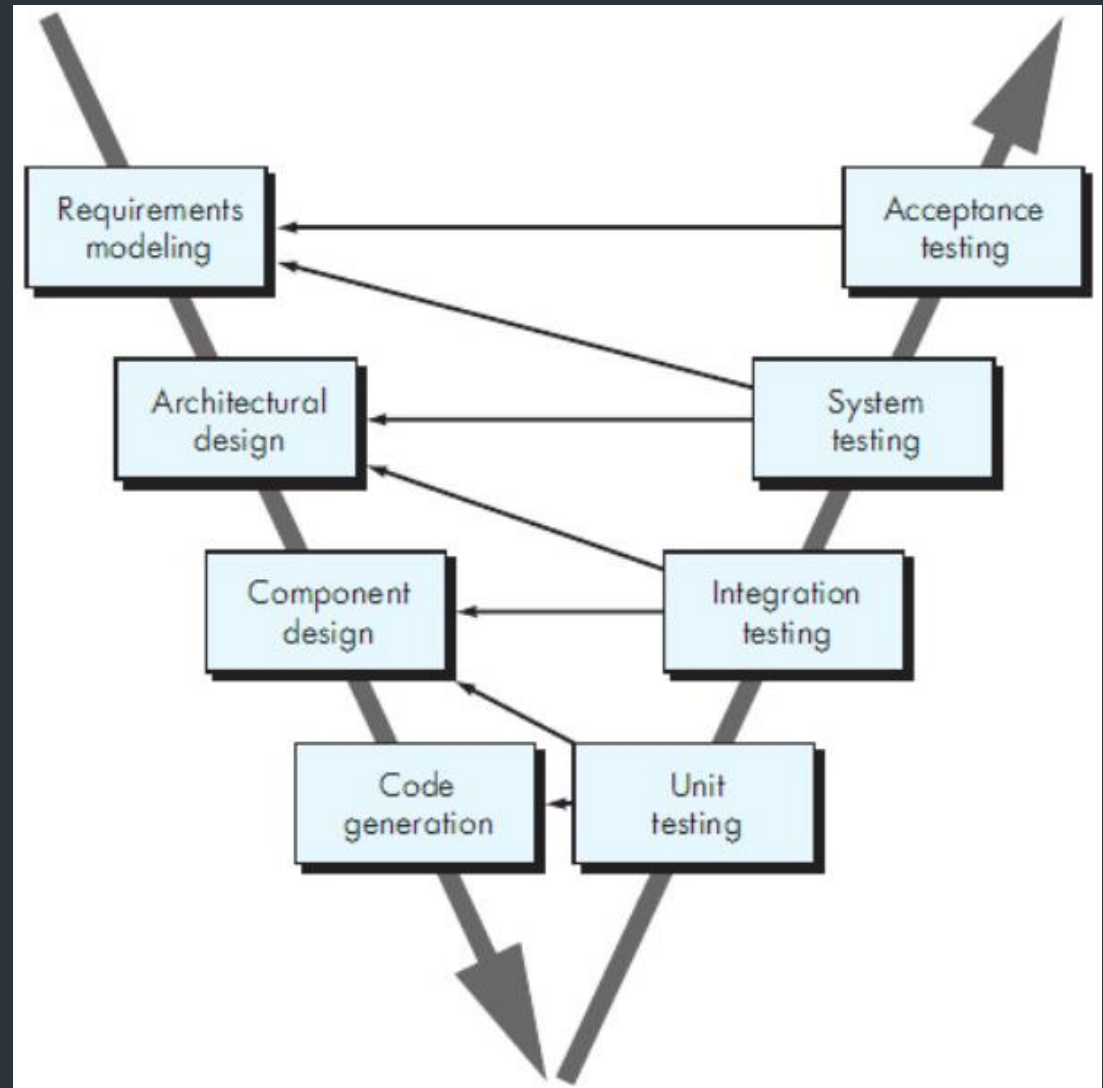
✓ classic life cycle

Reasons for Limitations of the Waterfall Model

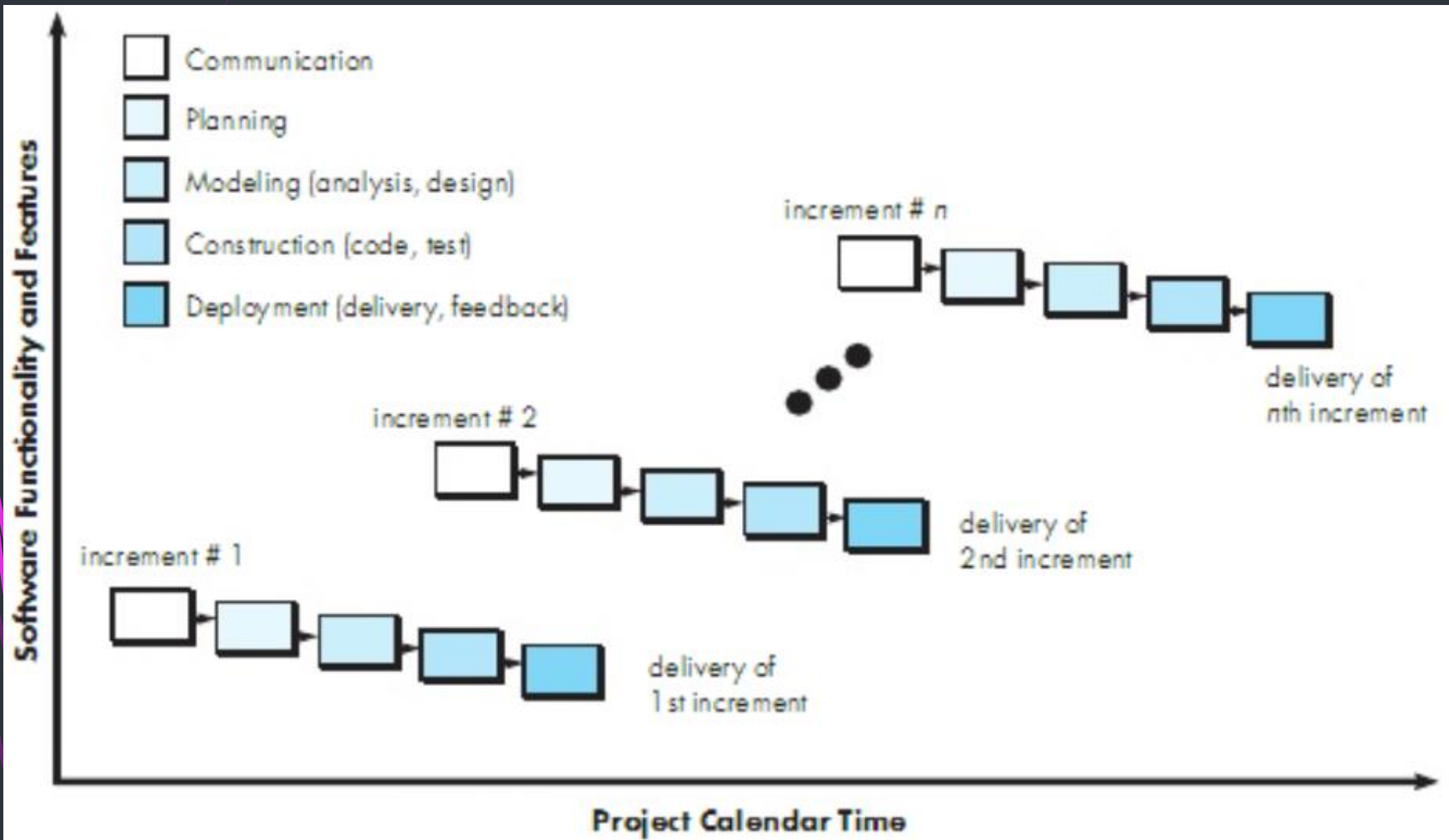
- Real projects rarely follow the sequential flow
- The natural uncertainty exists at the beginning of many projects. It is often difficult for the customer to state all requirements explicitly.
- A major blunder, if undetected until the working program is reviewed, can be disastrous .

The V-model

- The V-model illustrates how verification and validation actions are associated with earlier engineering actions.
- No fundamental difference between the classic life cycle and the V-model



Incremental Process Models



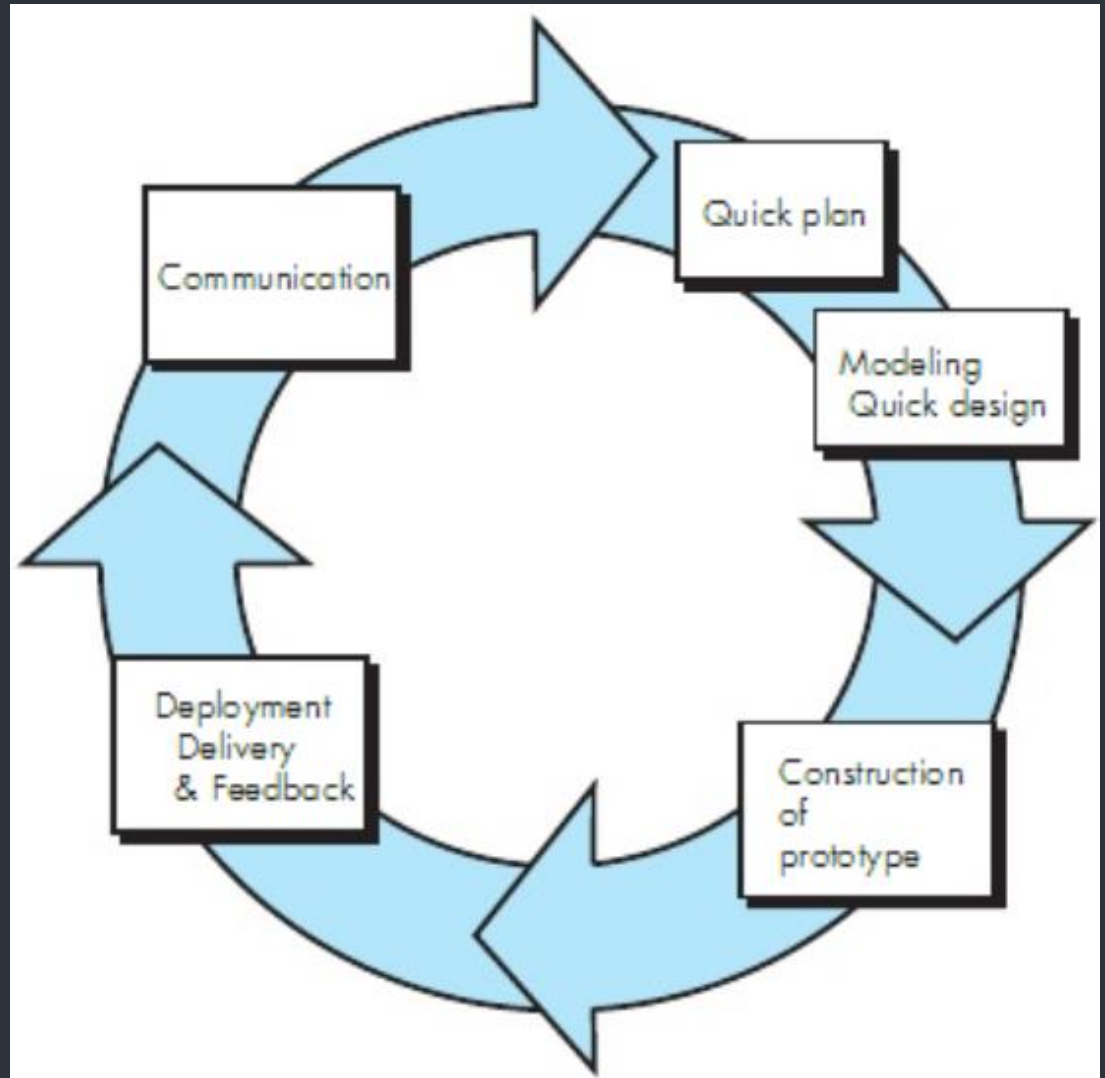
Incremental Process Models

- ▶ The incremental model delivers a series of releases, called increment, that provide progressive more functionality for the customer as each increment is delivered.
- ▶ *Your customer demands delivery by a date that is impossible to meet. Suggest delivering one or more increment by that date and the rest of the software (additional increments) later.*

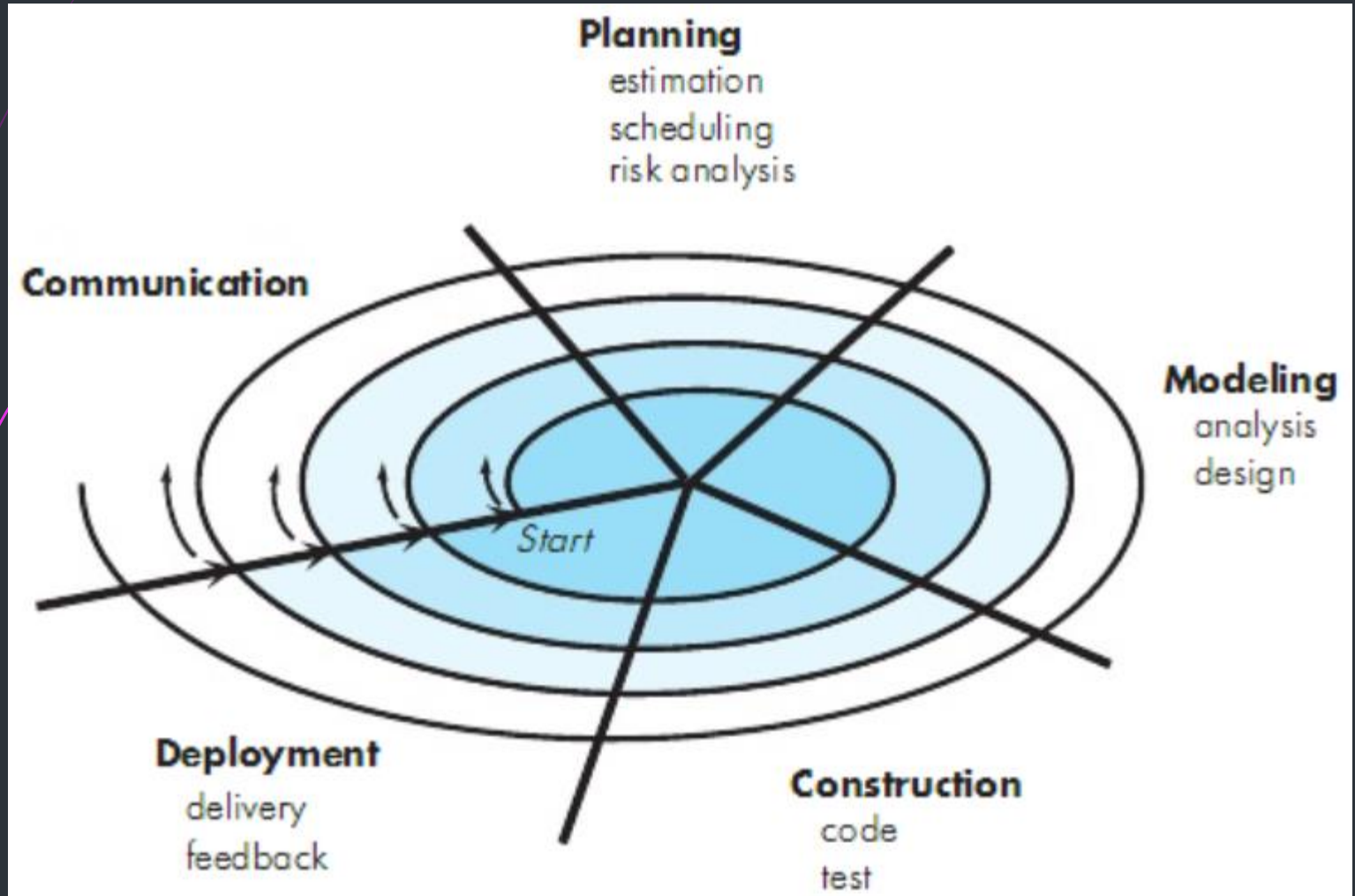
Evolutionary Process Models

- Software, like all complex systems, evolves over a period of time.
- A set of core product or system requirements is well understood, but the details of product or system extensions have yet to be defined.
- Evolutionary models are iterative and enable to develop increasingly more complete versions of the software.

Prototyping



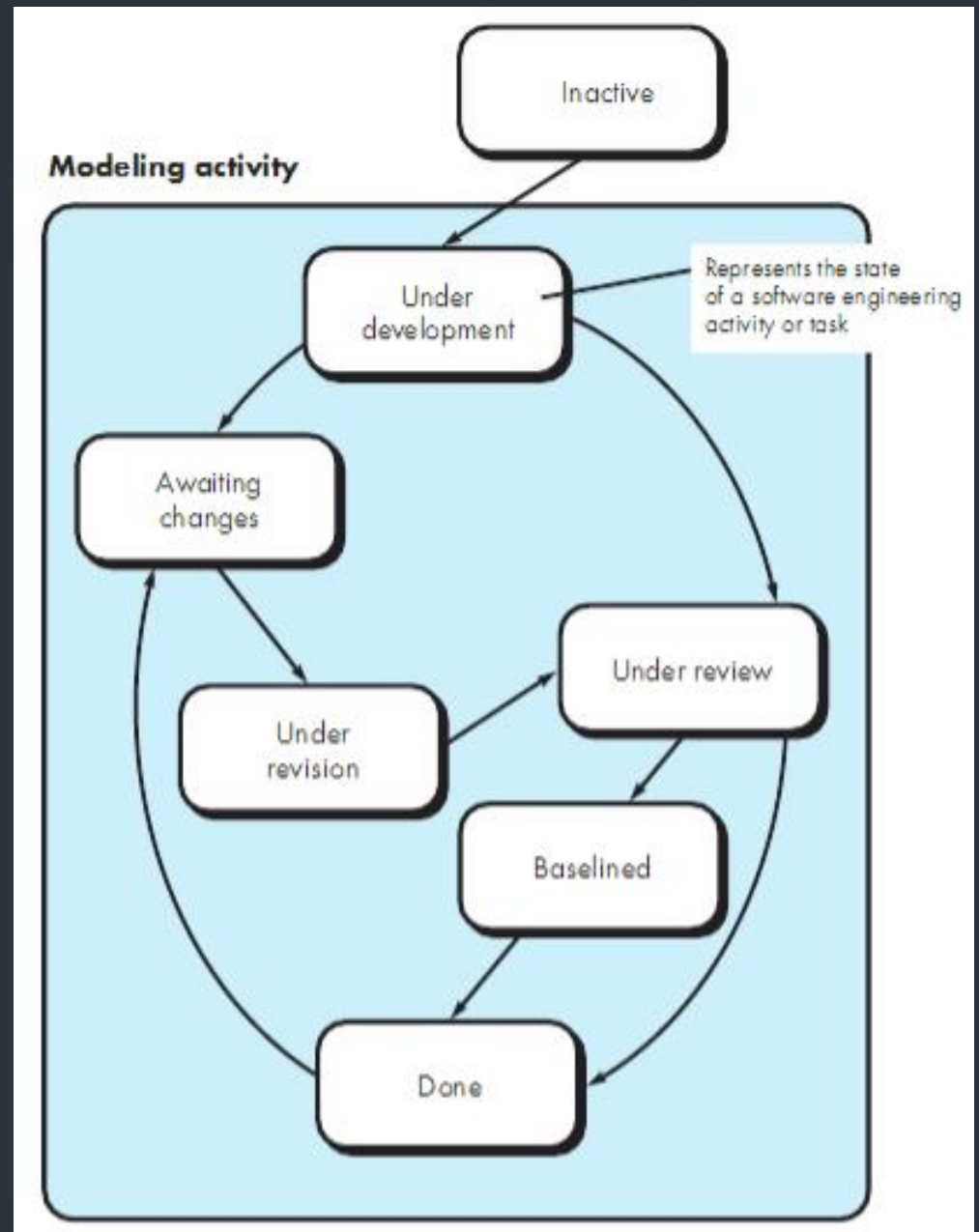
A Typical Spiral Model



A Typical Spiral Model

- Software is developed in a series of evolutionary releases.
- The spiral model can be adapted to apply throughout the entire life cycle of an application from concept development to maintenance.
- Risk is considered as each revolution is made. *Anchor point* milestones—a combination of work products and conditions that are attained along the path of the spiral are noted for each evolutionary pass.

Concurrent Models



Concurrent Models

- ▶ Apply to all types of software development and provide an accurate picture of the current stage of a project.
- ▶ Each activity, action, or task on the network exists simultaneously with other activities, actions, or tasks.
- ▶ Project plans must be viewed as living documents, progress must be assessed often and reviewed to take changes into account.
- ▶ *The concurrent model is often more appropriate for product engineering projects where different engineering teams are involved.*

CONTENTS

- **Prescriptive Process Models**
- **Specialized Process Models**
- **The Unified Process Model**
- **Personal and Team Process Models**

Specialized Process Models

- ▶ Component-based Development
 - ✓ Reduction in development cycle time
 - ✓ Reduction project cost
- ▶ The formal methods model
 - ✓ Formal methods to specify, develop, and verify a computer-based system.
- ▶ Aspect-Oriented Software Development (AOSD)
 - ✓ A process or methodological approach for defining, specifying, designing, constructing aspect – mechanisms beyond subroutines and inheritance for localizing the expression of a crosscutting concern.

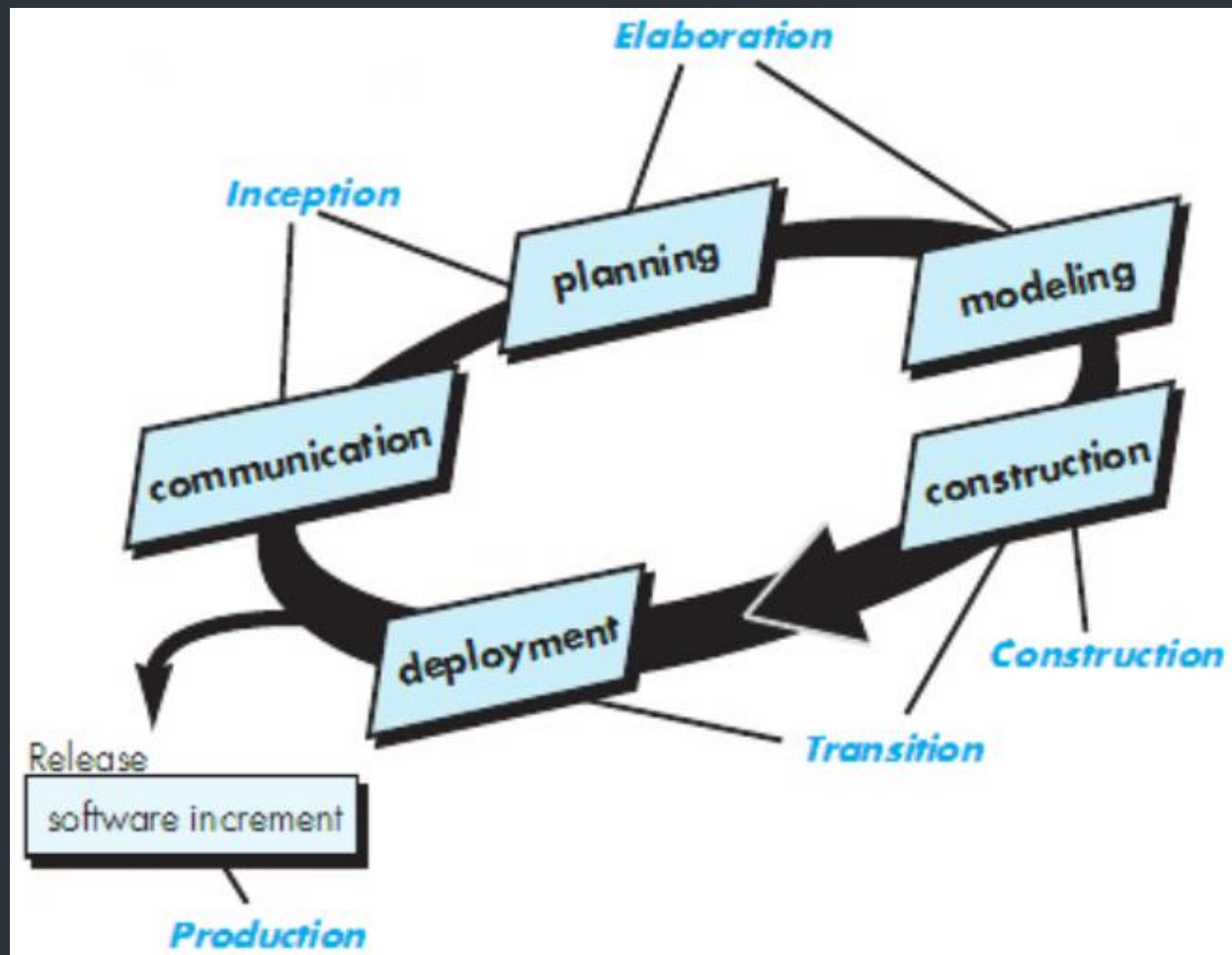
CONTENTS

- **Prescriptive Process Models**
- **Specialized Process Models**
- **The Unified Process Model**
- **Personal and Team Process Models**

The Unified Process Model

- ▶ an attempt to draw on the best features and characteristics of traditional software process models, but characterize them in a way that implements the best principles of agile software development .

Phases of The Unified Process



Phases of The Unified Process

1. Inception Phase
2. Elaboration Phase
3. Construction Phase
4. Transition Phase
5. Production Phase

CONTENTS

- **Prescriptive Process Models**
- **Specialized Process Models**
- **The Unified Process Model**
- **Personal and Team Process Models**

Personal and Team Process Models

- Personal Software Process
- Team Software Process

Personal Software Process (PSP)

- ▶ emphasize personal measurement
 - ✓ the work product
 - ✓ the resultant quality of the work product
- ▶ make the practitioner responsible for project planning.
- ▶ empower the practitioner to control the quality of all work products.

PSP' Five Activities

1. Planning

- ✓ Estimating and scheduling
- ✓ All metrics are record

2. High-level design

- ✓ Component design created
- ✓ Prototypes are built
- ✓ Uncertainty exits
- ✓ All issues are recored and tracked

3. High-level design review

- ✓ Verification methods are applied
- ✓ Metrics are maintained

PSP' Five Activities

4. Development

- ✓ The component-level design is refined and reviewed.
- ✓ Code is generated, reviewed, compiled, and tested.
- ✓ Metrics are maintained for important tasks and work results.

5. Postmortem

- ✓ Measures and metrics will provide guidance for modifying the process to improve its effectiveness.

→ PSP emphasizes the need to record and analyze the types of errors you make, so that you can develop strategies to eliminate them.

Team SoftwareProcess (TSP)

- Build self-directed teams that plan and track their work, establish goals, and own their processes and plans.
- Show managers how to coach and motivate their works and how to help them sustain peak performance.
- Accelerate software process improvement by making CMM level 5 behavior normal and expected.
- Provide improvement guidance to high-maturity organizations
- Facilitate team skills.