UNIVERSITI TENAGA NASIONAL

B. Computer Sci. (SE) (Hons.)

CSEB233: Fundamentals of Software Engineering

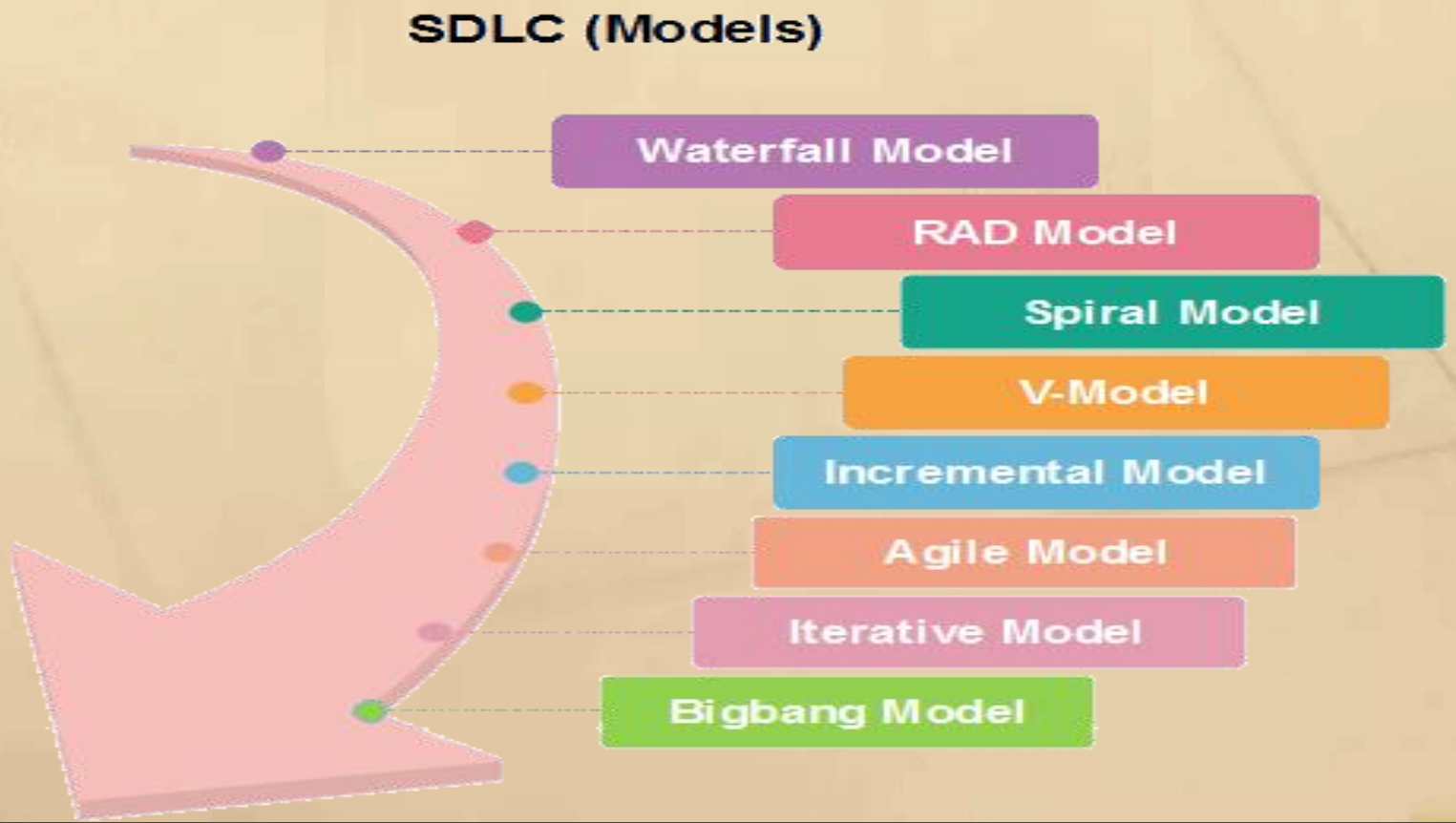
Process Models

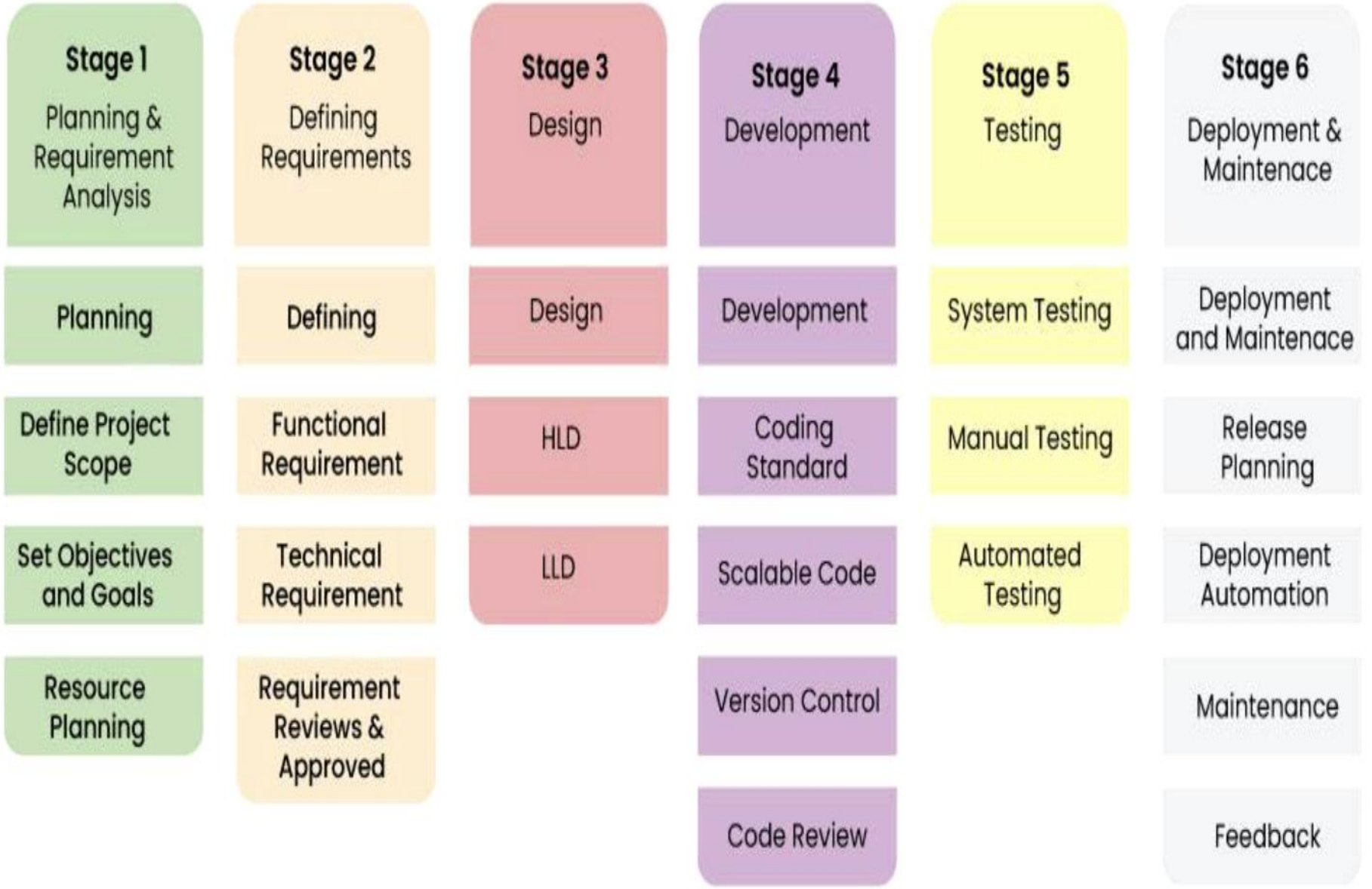
Lesson Objectives

- Describe the types of process flows
- Determine task set for software process activities
- Explain software process patterns
- Discuss several process assessment and improvement frameworks
- Analyze prescriptive and specialized software process models
- Select a process model for software development project

SDLC

- SDLC stands for Software Development Lifecycle.
- It is the framework that describes the activity performed at each stage of a software development project.





6 Stages of Software Development Life Cycle

REASONS FOR USING SDLC MODEL

- ✓ Provides basis for project planning ,estimating and scheduling
- ✓ Provide framework for standard set of terminologies, activity& deliverable
- ✓ Provide mechanism for project tracking & control
- ✓ Increase visibility of project progress to all stakeholders

Advantage of choosing An Appropriate SDLC

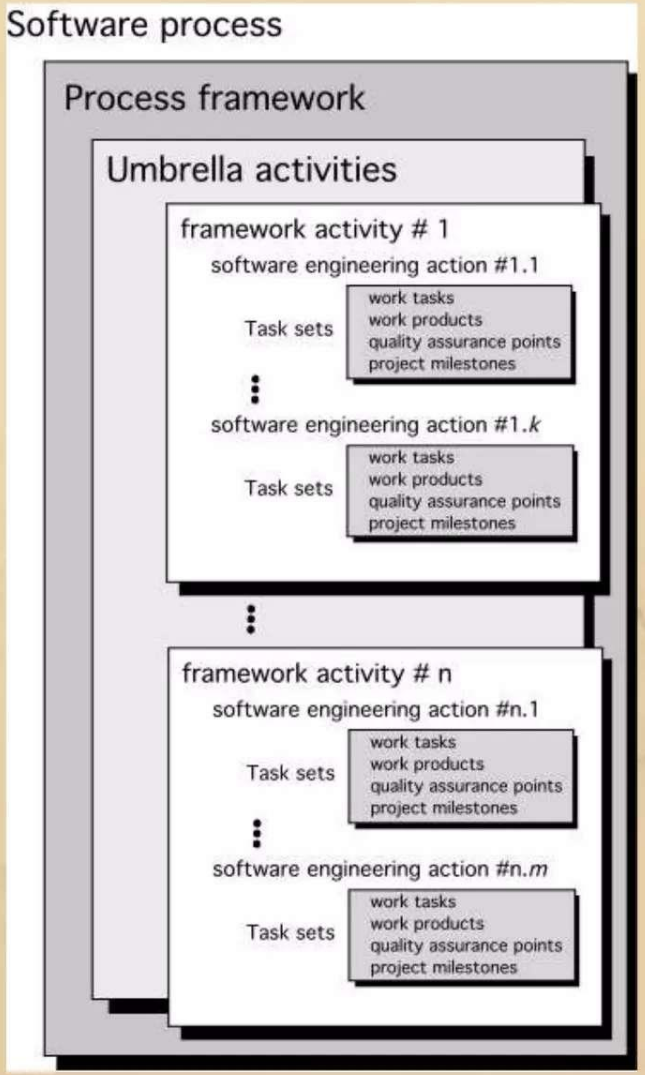
- Increase development speed
- Increase product quality
- Improve tracking and control
- Improve client and relation
- Decrease project risk
- Decrease project management overhead

A Generic Process Model

- A software process:
  - a collection of work activities, actions, tasks, which are performed when software is to be created.
- A process model or framework
  - is where these activities, actions, and tasks reside, and that defines their relationship with the process and with one another.
  - Also known as an abstract representation of a software process.

A Generic Process Model

- Shows the technical work hierarchy
  - activities encompassing actions, populated by tasks
- Each action is defined by a set of task that defines:
  - the actual work to complete
  - the work products to produce
  - the quality assurance filters to apply, and
  - the milestones that are used to indicate the project and product progress



Process Activities
Framework & Umbrella

- Framework activities :
  - generic activities that are applicable to all software projects, regardless of their size or complexity
  - Include communication; planning; modeling, construction; and deployment
- Umbrella activities:
  - complementary activities applied throughout a software project and help manage and control progress, quality, change, and risk
  - Include project tracking and control; risk management; software quality assurance (SQA); technical reviews; configuration management (CM); etc.

UNIVERSITI TENAGA NASIONAL

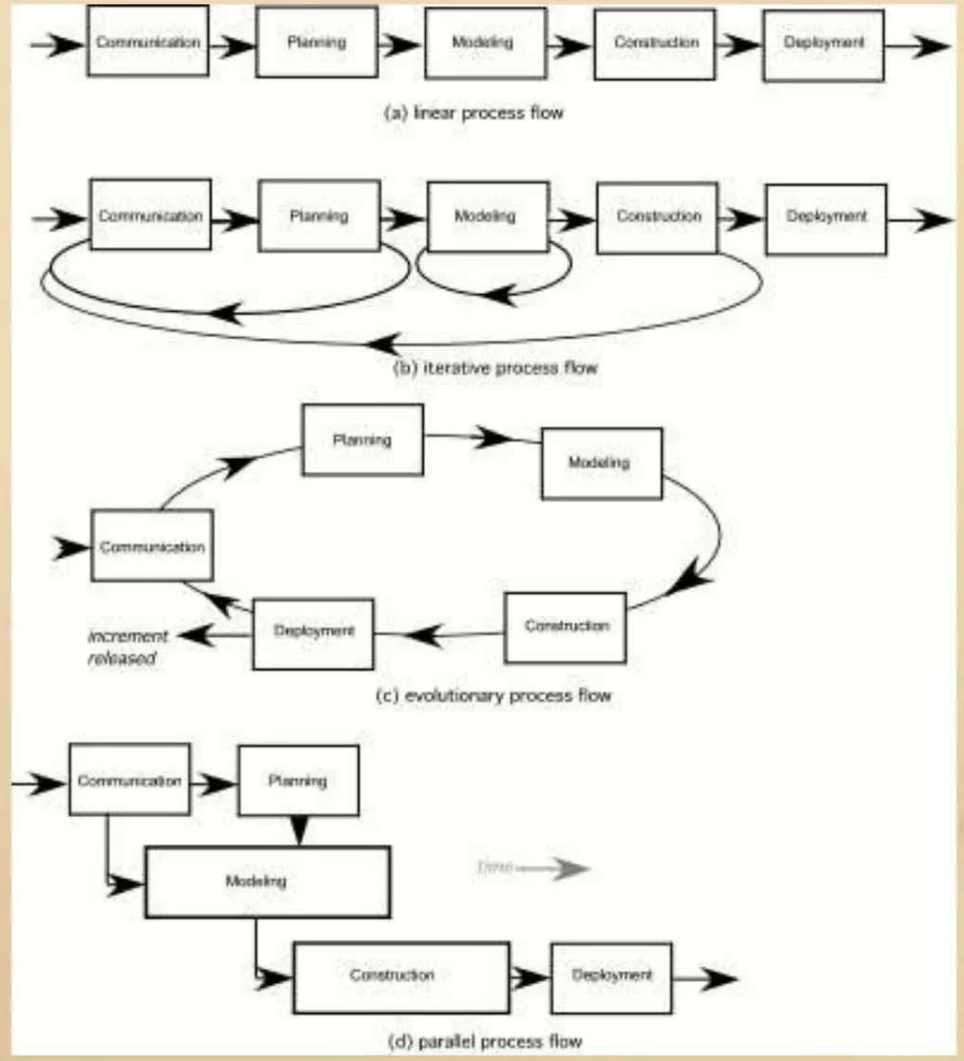
Process Models

Process Flows

Process Flow

- Describes how the framework activities and the actions and tasks that occur within each activity are organized with respect to sequence and time
- The flows:
  - Linear: execute the framework activities in sequence
  - Iterative: repeats one or more of the activities before proceeding to the next
  - Evolutionary: execute the activities in a ‘circular’ manner
  - Parallel: executes one or more activities simultaneously with other activities

Process Flow



UNIVERSITI TENAGA NASIONAL

Process Models

Task Set

Identifying a Task Set

- A task set defines the actual work to be done to accomplish the objectives of a software engineering action
  - A list of the tasks to be accomplished
  - A list of the work products to be produced
  - A list of the quality assurance filters to be applied

Identifying a Task Set

Key Questions in Determining Task Set

- Different projects require different task sets
  - The tasks should be selected based on problems and project characteristics
- Q: What actions are appropriate for a framework activity, given:
  - the nature of the problem to be solved;
  - the characteristics of the people doing the work; and
  - the stakeholders of the project?
- Q: What work tasks (task set) that these actions should encompass?

Identifying a Task Set
Example

- Nature of the problems and project :
  - A small software project requested by one person (at a remote location) with simple, straightforward requirements.
- Actions:
  - Communication action: develop requirements
- Task set:
  - Make contact with stakeholder via telephone.
  - Discuss requirements and take notes.
  - Organize notes into a brief written statement of requirements.
  - E-mail to stakeholder for review and approval.

UNIVERSITI TENAGA NASIONAL

Process Models

Process Patterns

Process Patterns

- A process pattern
  - describes a process-related problem that is encountered during software engineering work,
  - identifies the environment in which the problem has been encountered, and
  - suggests one or more proven solutions to the problem
- In more general terms, a process pattern provides a template
  - a consistent method for describing problem solutions within the context of the software process

Process Pattern Types

- Stage patterns
  - defines a problem associated with a framework activity for the process
- Task patterns
  - defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice
- Phase patterns
  - define the sequence of framework activities that occur with the process, even when the overall flow of activities is iterative in nature

UNIVERSITI TENAGA NASIONAL

Process Models

Process Assessment & Improvement

Process Assessment and Improvement: Approaches

- Standard CMMI Assessment Method for Process Improvement (SCAMPI):
  - Used to identify strengths, weaknesses, and ratings relative to SEI CMMI appraisal reference model, which is applicable to internal process improvement and external capability determination
- CMM-Based Appraisal for Internal Process Improvement (CBA IPI):
  - Provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment [Dun01].
  - However, CMM has been retired by SEI since the introduction of CMMI group of standards

Process Assessment and Improvement: Approaches

- SPICE (ISO/IEC15504):
  - Standard that defines a set of requirements for software process assessment.
  - The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process [ISO08]
- ISO 9001:2000 for Software:
  - A generic standard that applies to any organization that wants to improve the overall quality of the products, systems, or services that it provides.
  - Therefore, the standard is directly applicable to software organizations and companies [Ant06]

UNIVERSITI TENAGA NASIONAL

Process Models

Prescriptive & Specialized Process Models

Prescriptive Process Models

- Promote an orderly, structured approach to SE
- That leads to a few questions . . .
  - If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
  - Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

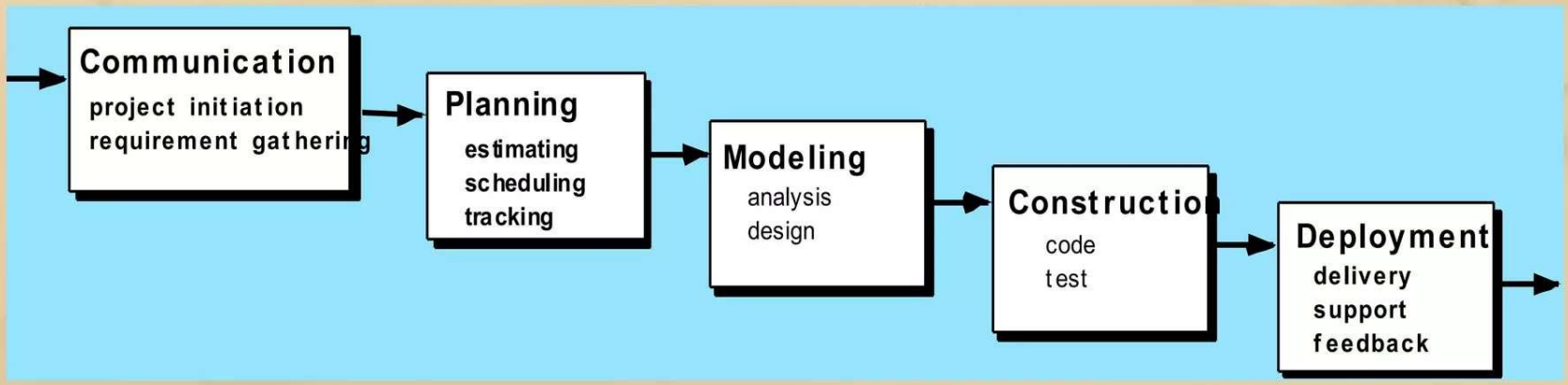
Prescriptive Process Models

- Waterfall Model
  - represents elements of a linear process flow
    - V
- Incremental Model
  - combines elements of linear and parallel process flows
- Evolutionary Model
  - follows the evolutionary process flow that combines elements of linear and iterative process flows
    - Prototyping
    - Spiral
- Concurrent Model
  - combines elements of iterative and parallel process flows

Prescriptive Process Models

The Waterfall Model

- Represents a linear process flow from communication through deployment
  - Also known as the classic SDLC
- The original Waterfall model proposed by Winston Royce in 1970 made provision for feedback loops
  - but many organizations apply this model as if it were strictly linear



Prescriptive Process Models

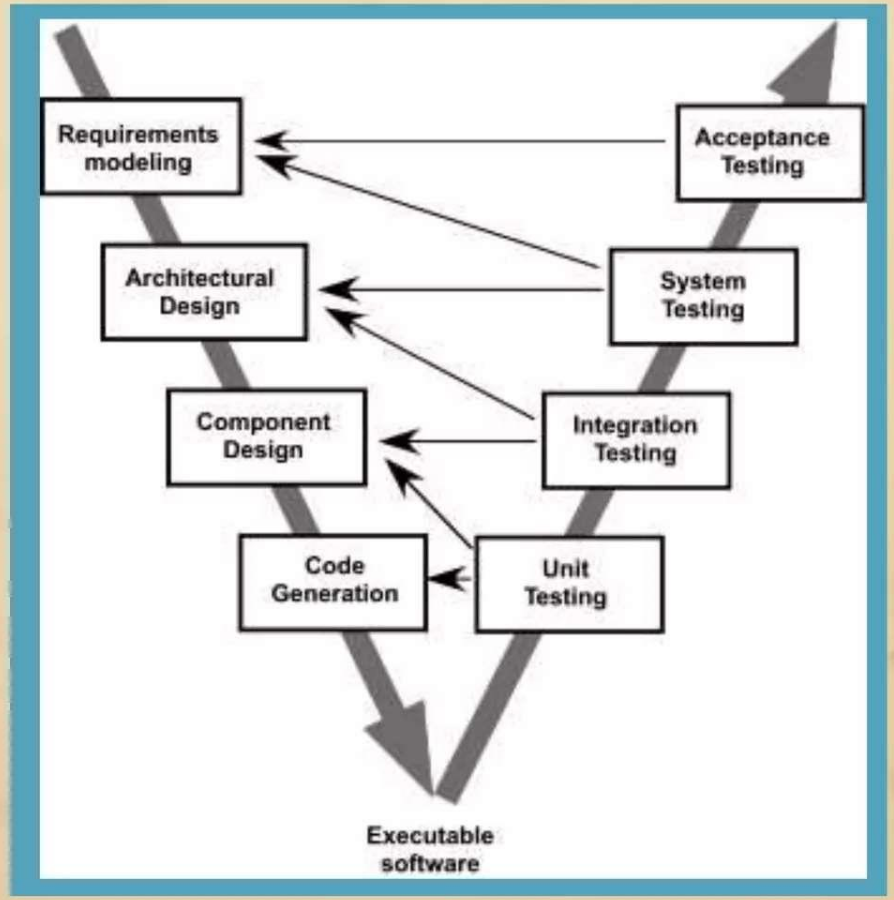
An Analysis of Waterfall Model

Characteristics	Strengths	Weaknesses	Applicability
• It suggests a systematic, sequential approach to SE, starting from requirements specification through planning, modeling, construction, testing, deployment and support of the completed system. • Each major activity is marked by milestones and deliverables (i.e. documents).	• Simple and easy to use/explain to customers. • The staged development cycle enforces discipline: every phase has a defined start and end point, and progress can be conclusively identified (through the use of milestones)	• Real projects rarely follow the linear flow that the model proposes. Although iteration is indirectly allowed, changes are costly, involve significant rework and can cause confusion to project team. • The model requires the customer to state all requirements explicitly, which is often very difficult to achieve. • The working software will not be available until late in the project, which can be disastrous for late discovery of major defects. • Leads to “blocking states” in which some project team members must wait for others to complete dependent tasks.	• When requirements are well understood and unlikely to change radically during system development (e.g., in a well-defined enhancement to an existing system). • When software development technologies and tools are well known. • The work tasks in the project are to proceed to completion in a linear manner.

Prescriptive Process Models

The V-Model

- Variation in representing the Waterfall model
- Illustrates how V&V actions are associated with earlier SE action
- There is no fundamental difference between the Waterfall model and the V-model



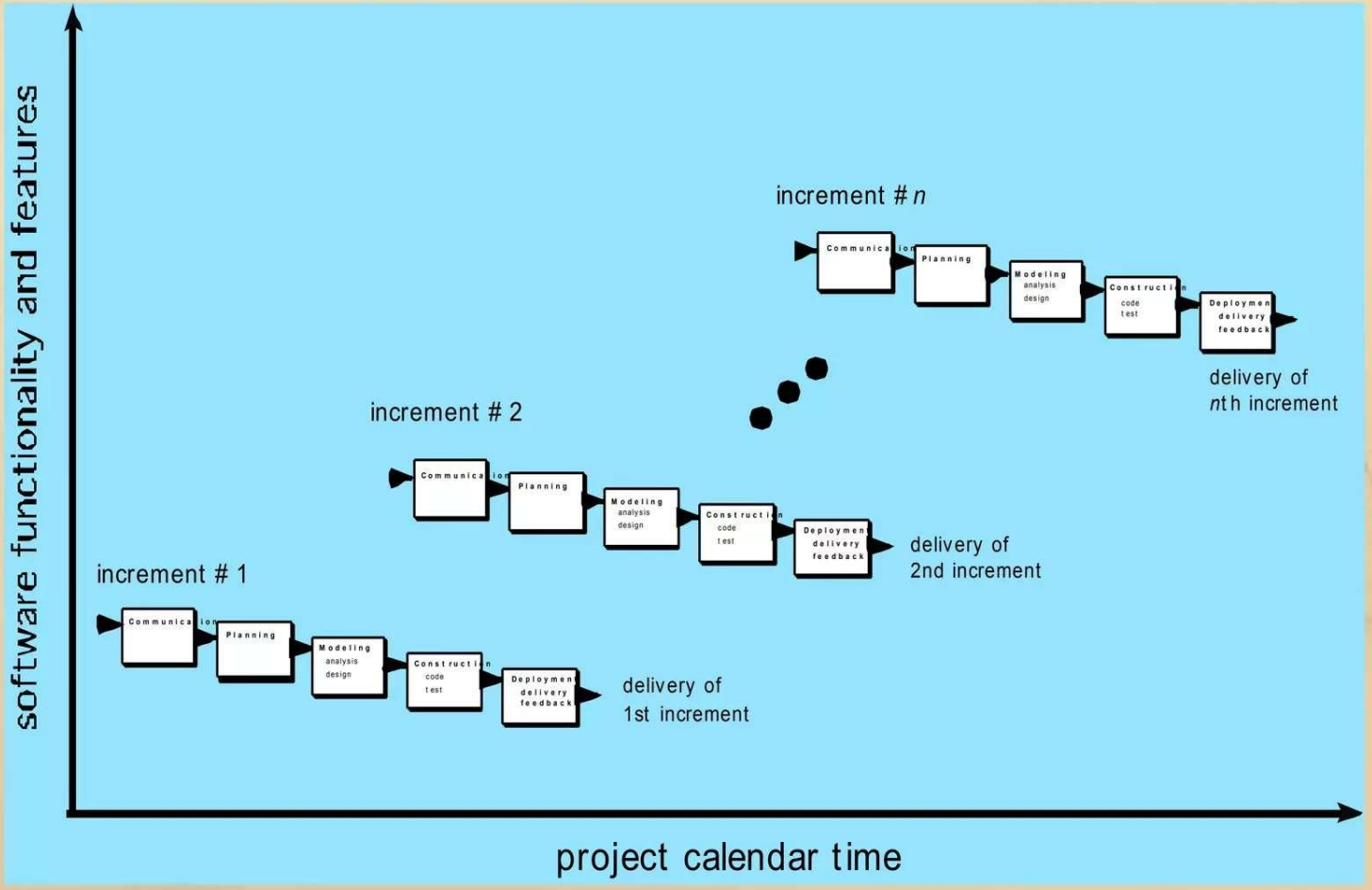
Prescriptive Process Models

The Incremental Model

- Rather than deliver the software product as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments
- Once the development of an increment is started, the requirements are frozen but requirements for later increments can continue to evolve

Prescriptive Process Models

The Incremental Model



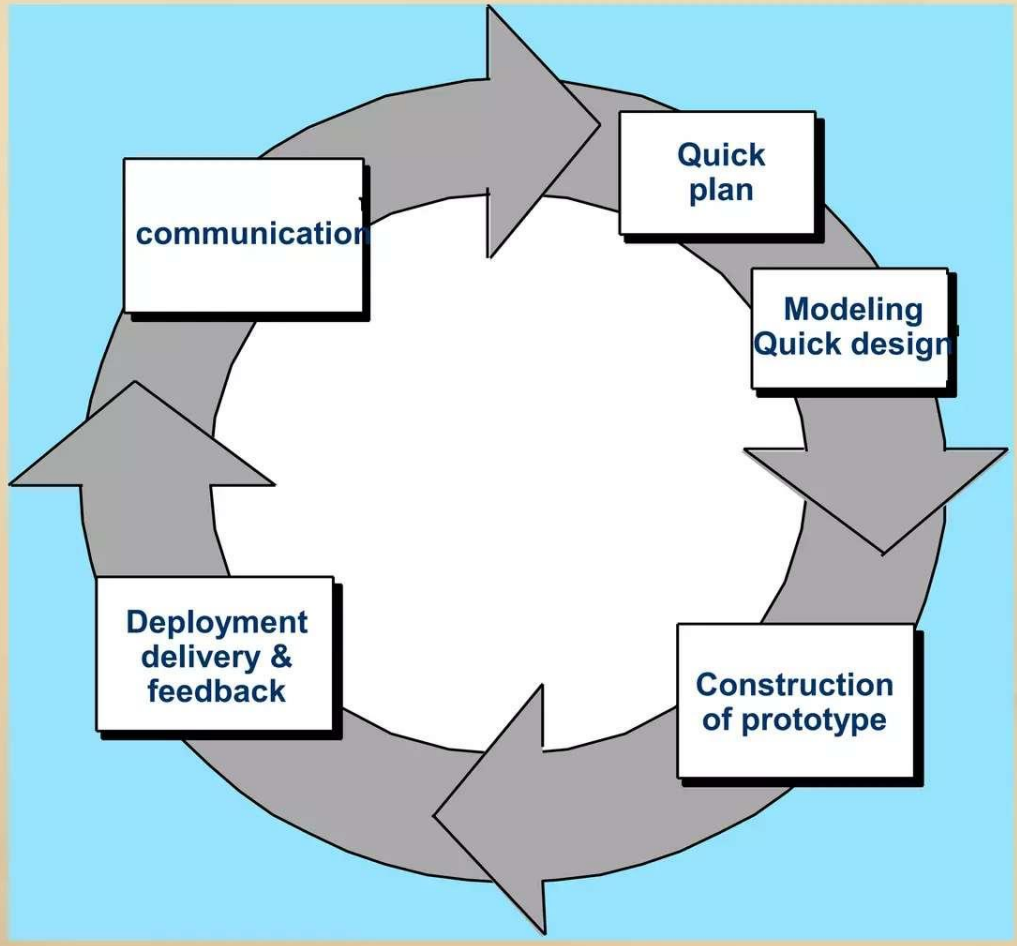
Prescriptive Process Models

The Evolutionary Model: Prototyping

- Using this process model, a prototype - an early approximation of a final software product - is built, tested, and then reworked as necessary
  - until an acceptable prototype is finally achieved from which the complete software product is developed
- Although it can be implemented as a stand-alone process model, it is more commonly used as part of other process models
- The main purpose of the model is to help better understand what it is to built when requirements are fuzzy

Prescriptive Process Models

The Evolutionary Model: Prototyping



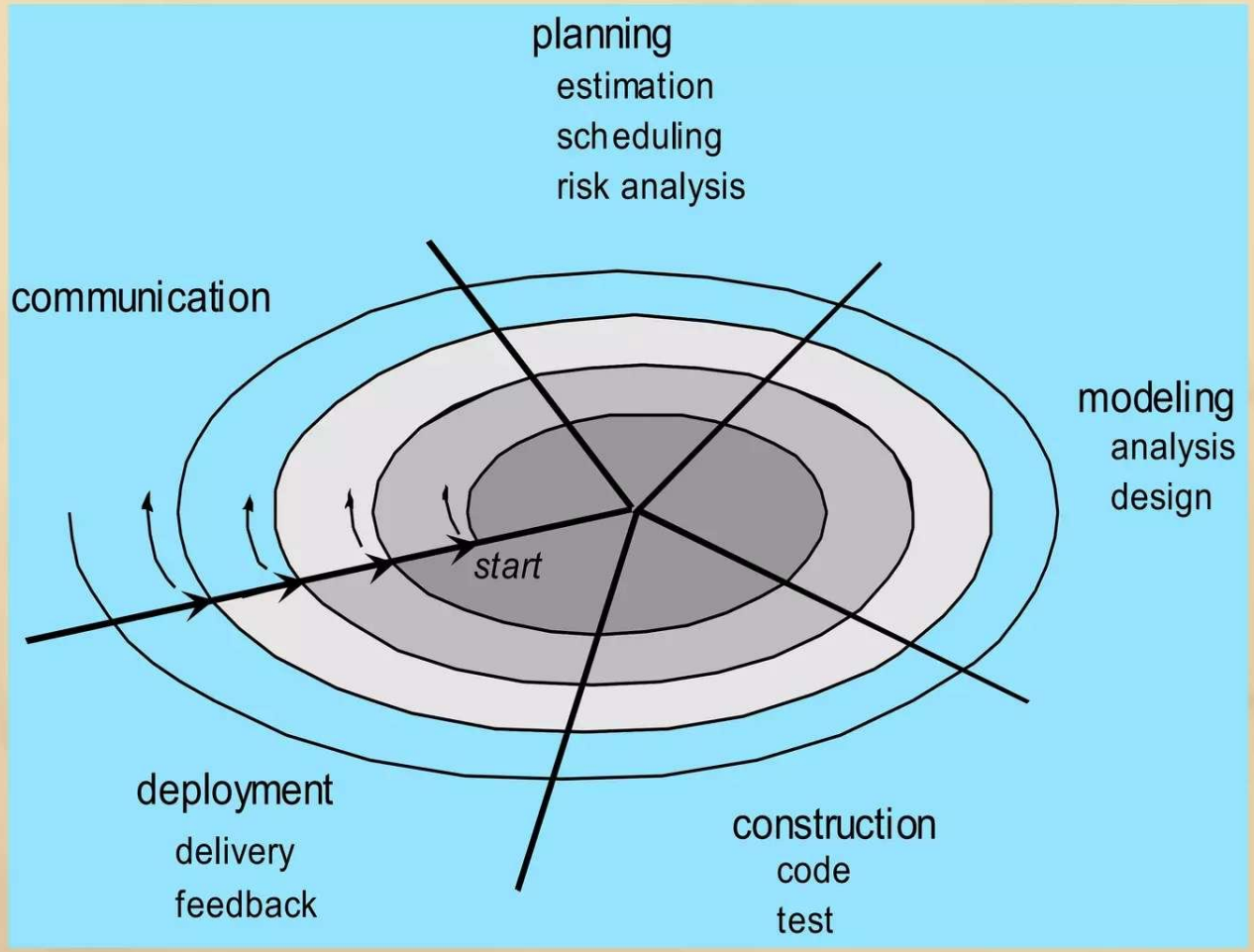
Prescriptive Process Models

The Evolutionary Model: Spiral

- A process model that combines the iterative nature of prototyping with the systematic aspects of waterfall model
- The spiral model can be thought of as a repeating waterfall model that emphasizes risk assessment and that is executed in an incremental fashion
- Each loop/pass through the spiral model consists of risk assessment and other framework activities from communication through deployment

Prescriptive Process Models

The Evolutionary Model: Spiral



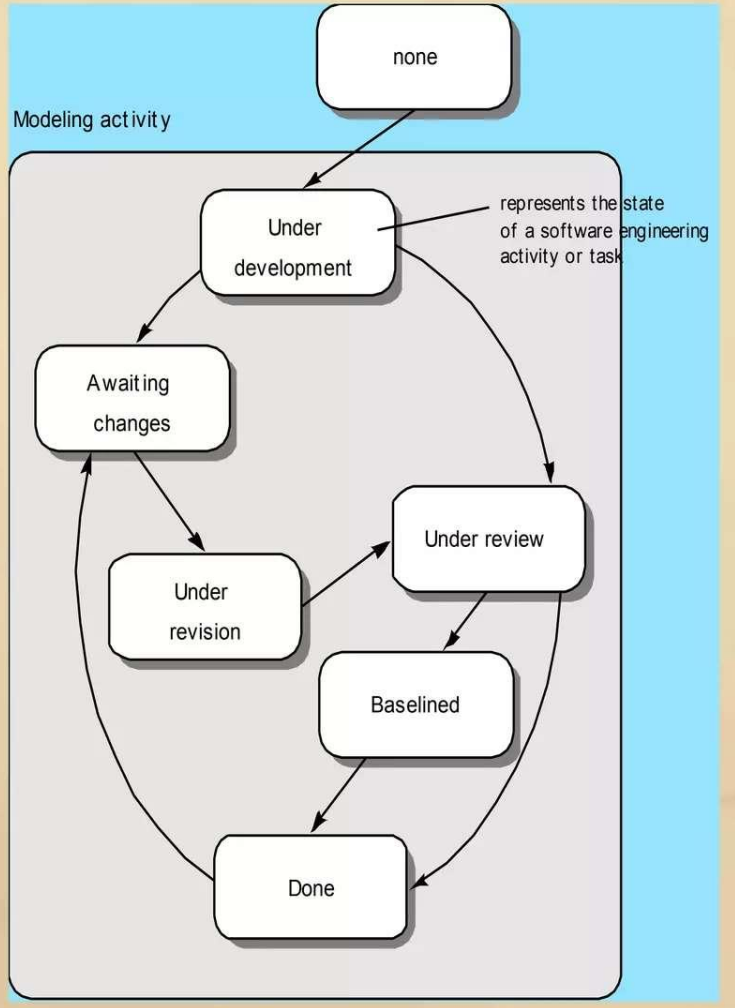
Prescriptive Process Models

The Concurrent Model

- A process model that combines the iterative and parallel elements of any of the prescriptive process models
- In this model, all SE activities (framework or umbrella) exist concurrently but reside in different states

Prescriptive Process Models

The Concurrent Model



Factors	Waterfall	V-Shaped	Evolutionary Prototyping	Spiral	Iterative and Incremental	Agile
Unclear User Requirement	Poor	Poor	Good	Excellent	Good	Excellent
Unfamiliar Technology	Poor	Poor	Excellent	Excellent	Good	Poor
Complex System	Good	Good	Excellent	Excellent	Good	Poor
Reliable system	Good	Good	Poor	Excellent	Good	Good
Short Time Schedule	Poor	Poor	Good	Poor	Excellent	Excellent
Strong Project Management	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
Cost limitation	Poor	Poor	Poor	Poor	Excellent	Excellent
Visibility of Stakeholders	Good	Good	Excellent	Excellent	Good	Excellent
Skills limitation	Good	Good	Poor	Poor	Good	Poor
Documentation	Excellent	Excellent	Good	Good	Excellent	Poor
Component reusability	Excellent	Excellent	Poor	Poor	Excellent	Poor

Specialized Process Models

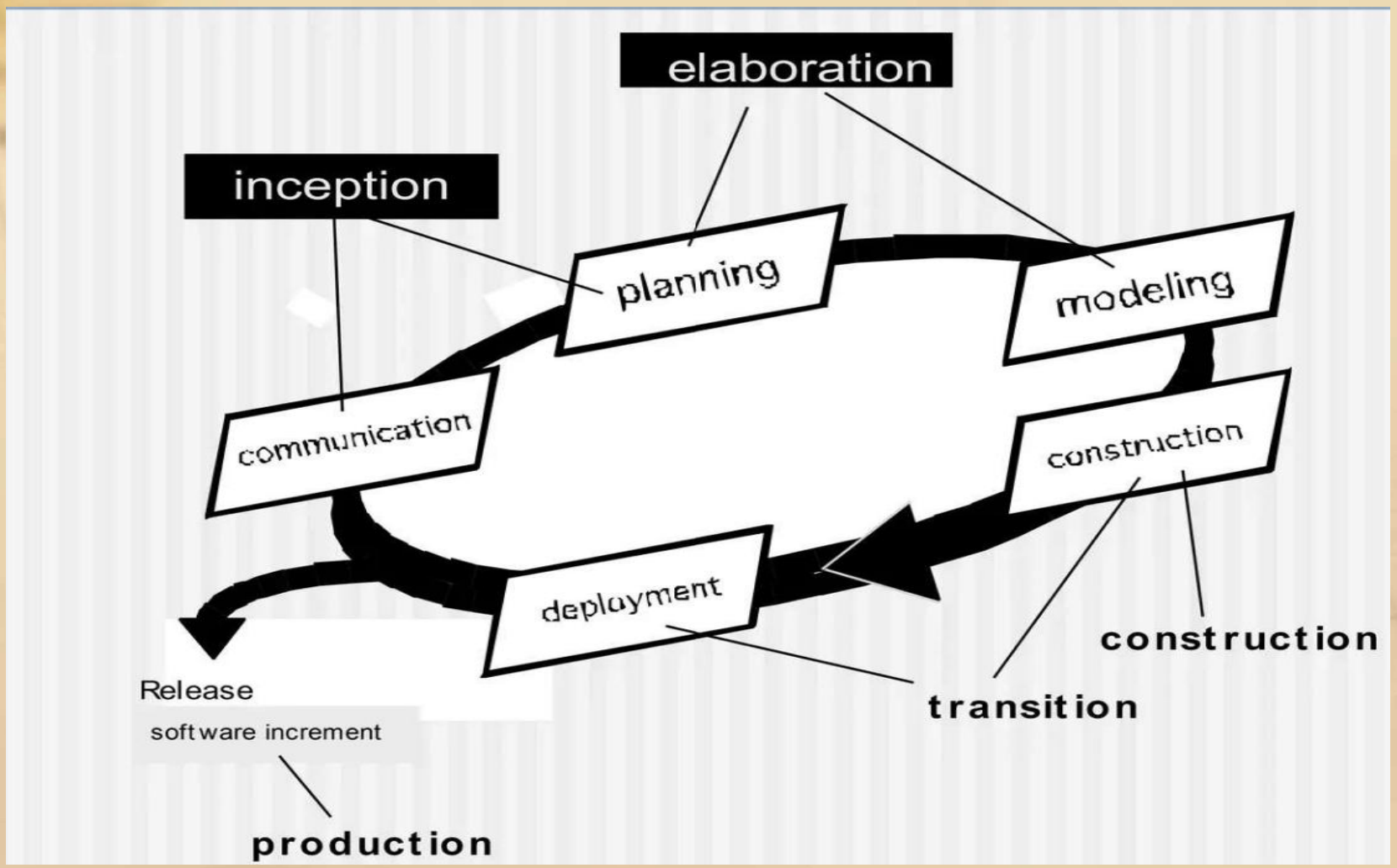
-is a model that developed on their owns because they does'nt trust prescriptive model

- Component based software development (CBSD)
  - the process to apply when reuse is a development objective
- Formal methods
  - emphasizes the mathematical specification of requirements, which can demonstrate software correctness but are not widely used.
- Aspect-oriented software development (AOSD)
  - provides a process and methodological approach for defining, specifying, designing, and constructing aspects
- Unified Process
  - a "use-case driven, architecture-centric, iterative and incremental" software process closely aligned with the Unified Modeling Language (UML)

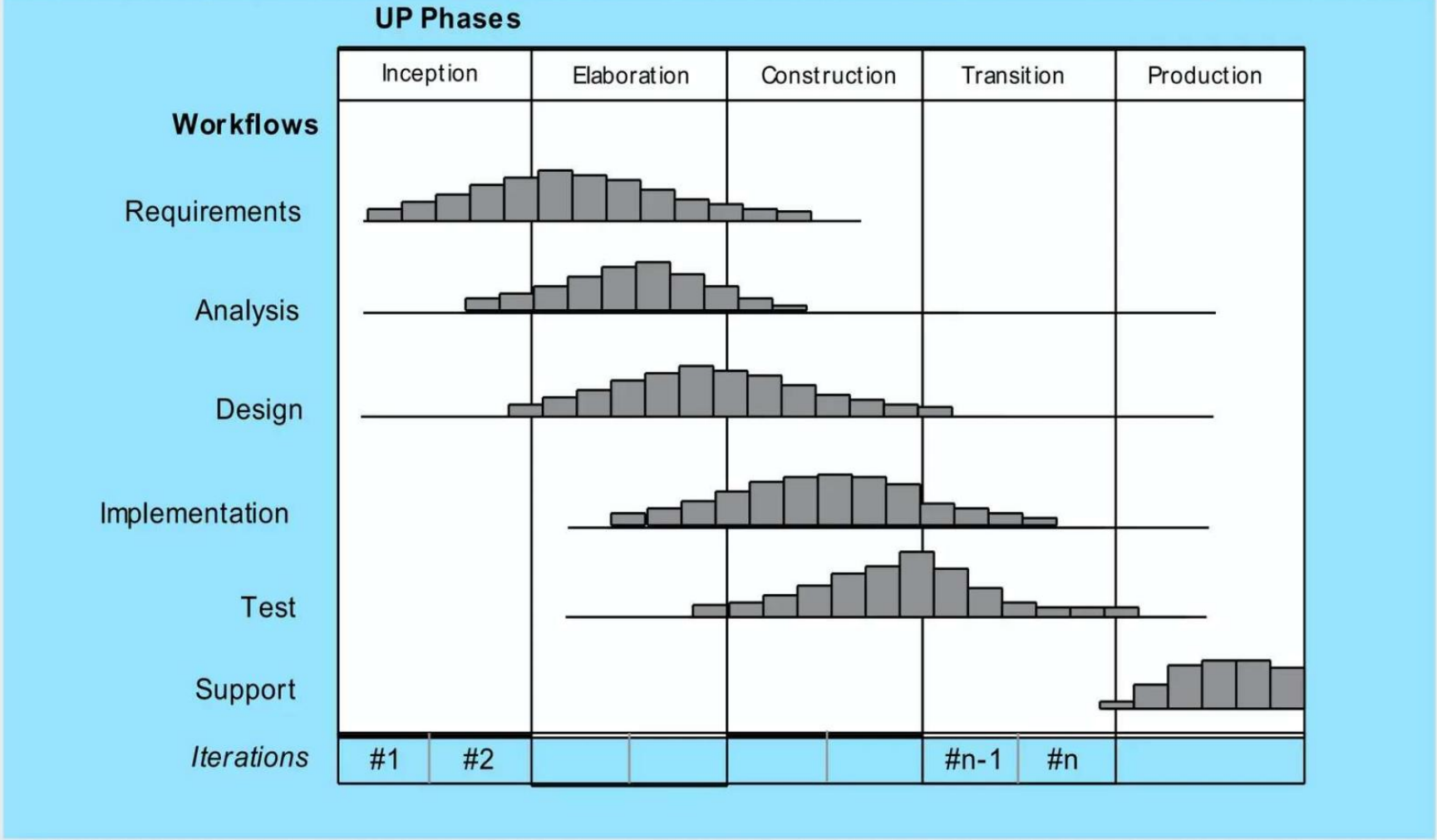
Specialized Process Models

- Agile Process
  - An iterative approach to requirements specification, construction and deployment, which support rapid changes to requirements
- Personal Process Model
  - Emphasizes the need to record and analyze errors each individual practitioner made, so that he/she can develop a strategy to eliminate them
- Team Process Model
  - Build self-directed teams that plan and track their work, establish goals, and own their processes and plans. These can be pure software teams or integrated product teams (IPT) of three to about 20 engineers
  - Show managers how to coach and motivate their teams and how to help them sustain peak performance

The Unified Process (UP)



UP Phases


UP Phases
Inception
Elaboration
Construction
Transition
Production
Workflows
Requirements
Analysis
Design
Implementation
Test
Support
Iterations
#1
#2
#n-1
#n

UP Work Products

Inception phase

Vision document
Initial use-case model
Initial project glossary
Initial business case
Initial risk assessment.
Project plan,
phases and iterations.
Business model,
if necessary.
One or more prototypes

Elaboration phase

Use-case model
Supplementary requirements
including non-functional
Analysis model
Software architecture
Description.
Executable architectural
prototype.
Preliminary design model
Revised risk list
Project plan including
iteration plan
adapted workflows
milestones
technical work products
Preliminary user manual

Construction phase

Design model
Software components
Integrated software
increment
Test plan and procedure
Test cases
Support documentation
user manuals
installation manuals
description of current
increment

Transition phase

Delivered software increment
Beta test reports
General user feedback

Personal Software Process (PSP)

- **Planning.** This activity isolates requirements and develops both size and resource estimates. In addition, a defect estimate (the number of defects projected for the work) is made. All metrics are recorded on worksheets or templates. Finally, development tasks are identified and a project schedule is created.
- **High-level design.** External specifications for each component to be constructed are developed and a component design is created. Prototypes are built when uncertainty exists. All issues are recorded and tracked.
- **High-level design review.** Formal verification methods (Chapter 21) are applied to uncover errors in the design. Metrics are maintained for all important tasks and work results.
- **Development.** The component level design is refined and reviewed. Code is generated, reviewed, compiled, and tested. Metrics are maintained for all important tasks and work results.
- **Postmortem.** Using the measures and metrics collected (this is a substantial amount of data that should be analyzed statistically), the effectiveness of the process is determined. Measures and metrics should provide guidance for modifying the process to improve its effectiveness.

Team Software Process (TSP)

- Build self-directed teams that plan and track their work, establish goals, and own their processes and plans. These can be pure software teams or integrated product teams (IPT) of three to about 20 engineers.
- Show managers how to coach and motivate their teams and how to help them sustain peak performance.
- Accelerate software process improvement by making CMM Level 5 behavior normal and expected.
  - The Capability Maturity Model (CMM), a measure of the effectiveness of a software process, is discussed in Chapter 30.
- Provide improvement guidance to high-maturity organizations.
- Facilitate university teaching of industrial-grade team skills.

UNIVERSITI TENAGA NASIONAL

Process Models

Selecting a Process Model

Selecting a Process Model

Factors to Consider

- The characteristics of the problems to be solved
  - Such as complexity of the problem, etc.
  - e.g. simple with clear, stable requirements, or complex with changing, unstable requirements, etc.
- The characteristics of the project
  - Such as the customers who have requested the product and the people who will do the work, etc.
  - e.g. Uncertain requirements, breakthrough technology
- The characteristics of the product
  - Such as quality attributes or metric of the product, product domain, etc.
- The project environment in which the software team works
  - Such as political, cultural, language, etc.

Process Management Tools

- Also known as process modeling tools or process technology
- Allows a team to define and manage the elements of a process model (activities, actions, task, work products, milestones, and QA points/filters)
- Such tools also provide detailed guidance on the content of each process element
- The tools may also provide standard project management tasks such as estimating, scheduling, tracking and control.
- Example:
  - Igrafx Process Tools (www.micrografx.com)

Selecting a Process Model
An Exercise

- The Project:
  - Assume that you are in charge of a project to create a portal for the Shah Alam district of Selangor.
  - This portal would include a homepage with links to a wide range of discounted travel packages to major destinations in Selangor, links to certain featured places like golf courses, shopping complexes and places to eat, links to the detailed map of Selangor, and links to news and events listing
  - It also includes a bulletin board and chat room feature where tourists (international and local tourists) can exchange information.
  - The portal should also provide Automated Teller Machine (ATM) locator, time zone converter, and currency converter.

Selecting a Process Model

An Exercise

- Select a software process model that you would recommend to be implemented in the above mentioned project
- Why is the software process model selected?

Summary

- There are four types of process flows – linear, iterative, evolutionary, and parallel.
- Software process patterns may suggest one or more proven solutions to the problem from other projects, which can be reused in another project
- There are several process assessment and improvements frameworks that can be exercised by practitioners
- The analyses of prescriptive and specialized software process models would help select the most appropriate process model for a software development project, which can be proceeded with the identification of task set for the project