Software Engineering (20CS440)

The Presentation Slides are Influenced by the Text Book *Software Engineering: A Practitioner's Approach, 8/e* (McGraw-Hill)

Dr. Trisiladevi C. Nagavi

Associate Professor

Unit I : Software Process

(Software and Software Engineering)

Chapter 1:The Nature of Software

Chapter 2: Software Engineering

Chapter 3: Software Process Structure

Chapter 4: Process Models

Chapter 4: Process Models

- **4.1 Prescriptive Process Models**
  - 4.1.1 The Waterfall Model
  - 4.1.2 Incremental Process Models
  - 4.1.3 Evolutionary Process Models
  - 4.1.4 Concurrent Models
  - 4.1.5 A Final Word on Evolutionary Processes

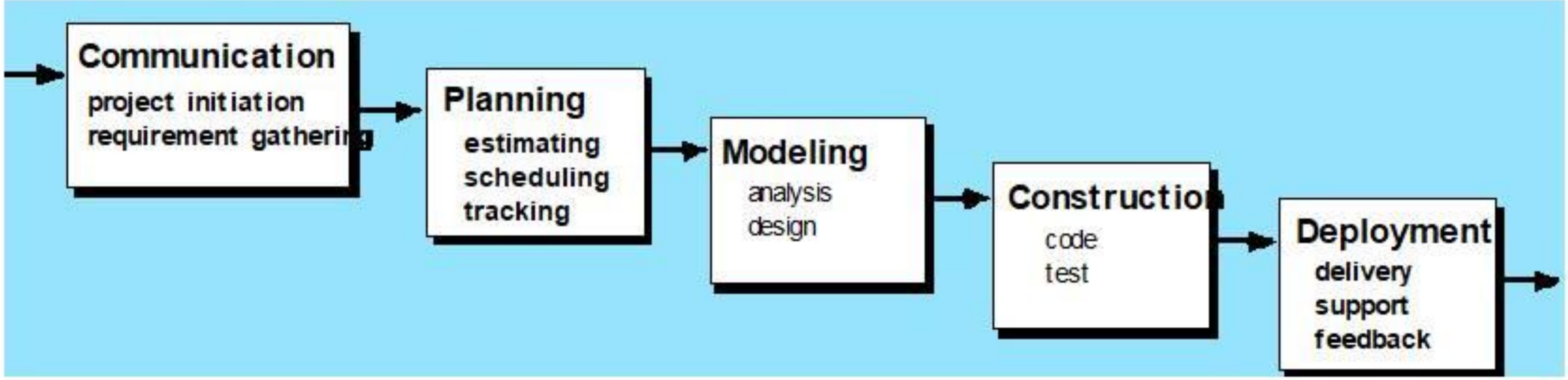
4.1 Prescriptive Models

- Advocate an **orderly approach** to software engineering.
- Scope for **creativity (If followed? If not)**

- Prescribe process elements - **framework activities, software engineering actions, tasks, work products, quality assurance, and change control mechanisms for each project. Also prescribes a process flow.**

- All process models accommodate generic framework activities, but each applies a different emphasis to activities and process flow.

4.1 Prescriptive Models: i) Waterfall Model

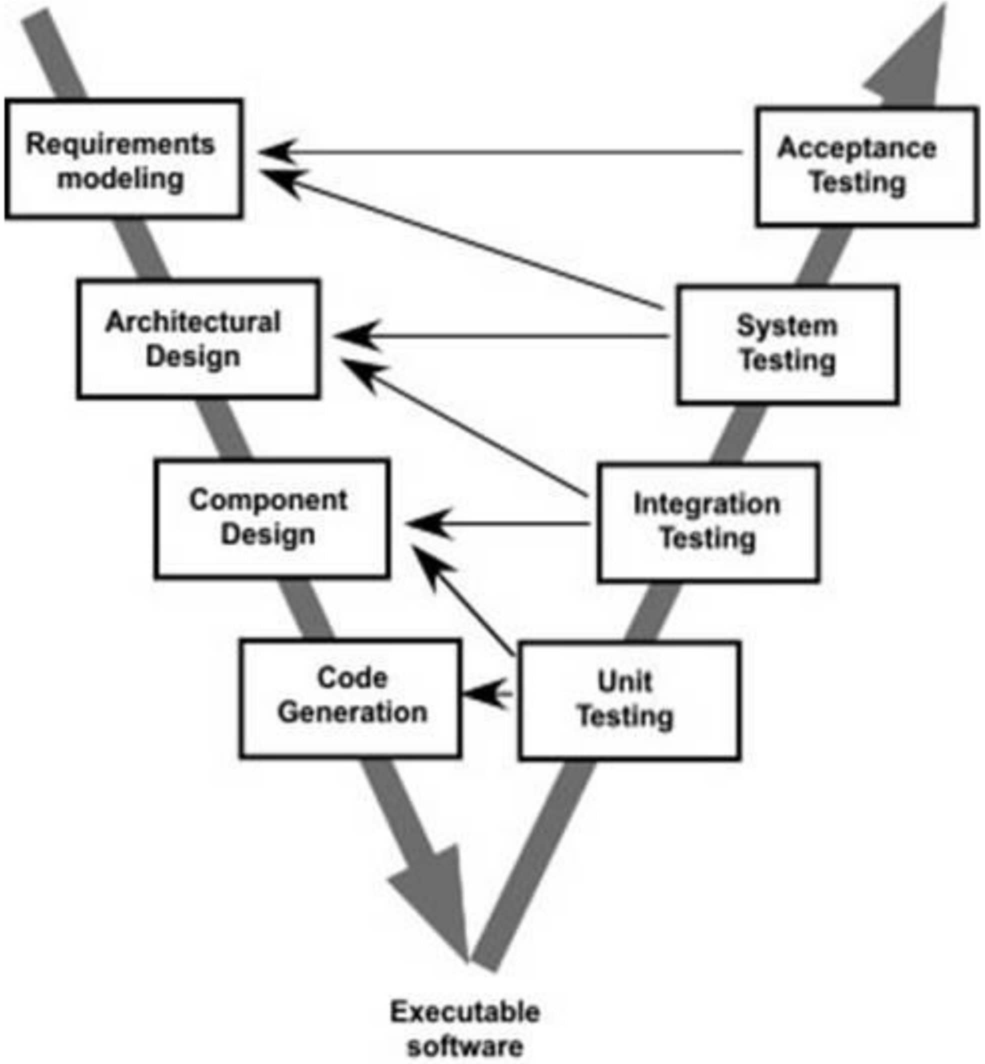


- Classic life cycle : **requirements are well understood**.

- **Well-defined adaptations or enhancements** to an existing system (e.g., an adaptation to accounting software that has been mandated because of changes to government regulations).

4.1 Prescriptive Models: i) Waterfall Model

The V-Model



- **Variation** in waterfall.
- Depicts **relationship** of quality assurance actions to actions associated with communication, modeling and code construction.
- Team **first moves down the left side then moves up right side** by performing a series of tests.

4.1 Prescriptive Models: i) Waterfall Model

- **Application Domains (Needs to be perfect)**

- Military soft wares,
- health systems,
- bank systems,
- control systems for industrial processes involving dangerous chemicals, nuclear materials or extreme conditions.

4.1 Prescriptive Models: i) Waterfall Model

- **No difference** between classical and v-model. **Visualization** is different.

- **Problems:**

1. Real projects **rarely sequential**. They accommodate **iteration for changes which causes confusion** as the project team proceeds.

2. **Difficulty accommodating the natural uncertainty that exists** at the beginning of many projects.

3. Customer must have **patience. A major blunder, if undetected until the working program is reviewed, can be disastrous.**

4.1 Prescriptive Models : ii) Incremental Model

- Initial requirements well defined.
- First increment is core product. Users use it and evaluate it with more modifications.

- Combines linear and parallel process flows. Each linear sequence produces deliverable increments.

- Repeated until the complete product is produced.
- Early increments are stripped-down versions, but they provide capability that serves the user and also provide a platform for evaluation.

4.1 Prescriptive Models : ii) Incremental Model

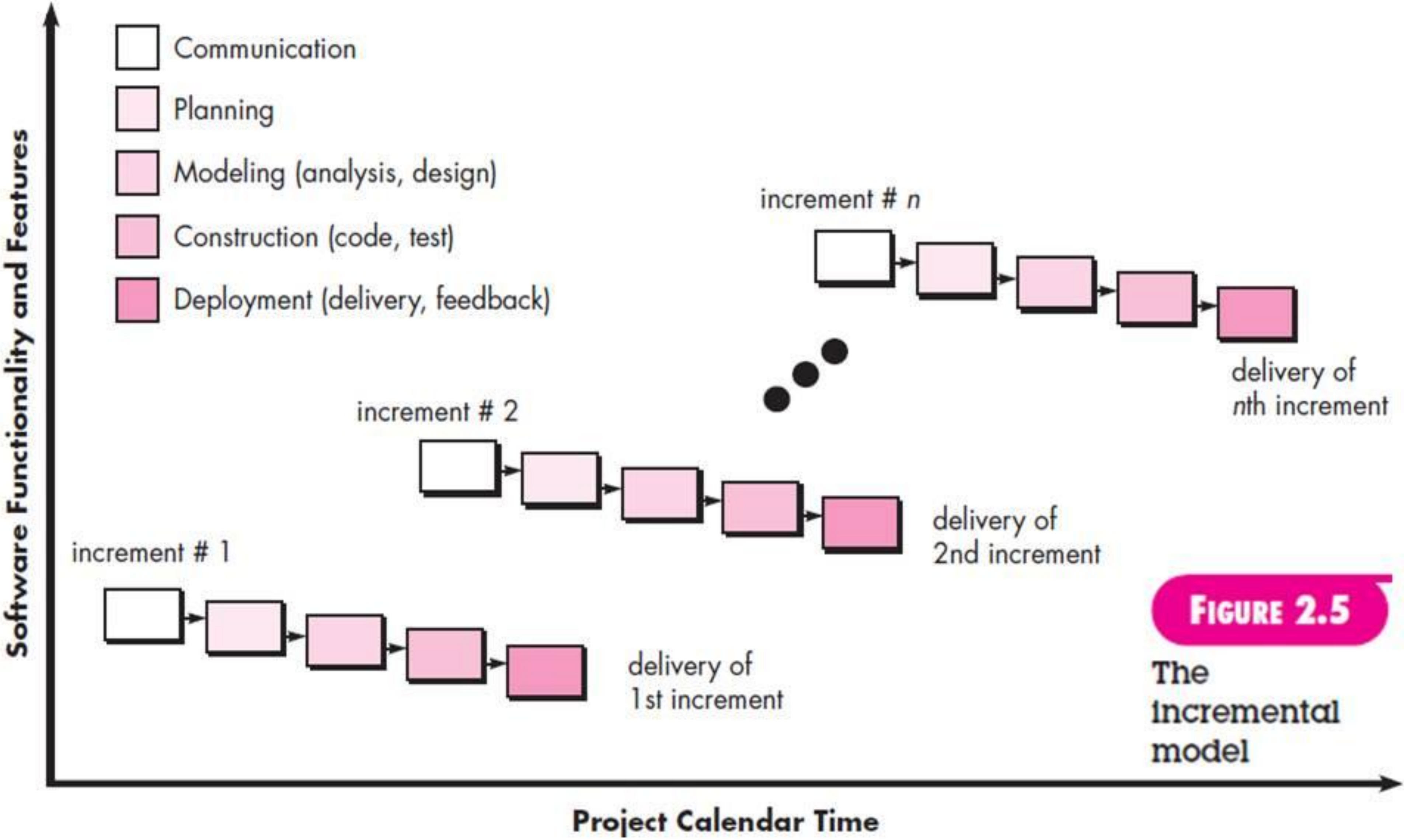

Communication
Planning
Modeling (analysis, design)
Construction (code, test)
Deployment (delivery, feedback)
increment # n
delivery of nth increment
increment # 2
delivery of 2nd increment
increment # 1
delivery of 1st increment
Software Functionality and Features
Project Calendar Time

FIGURE 2.5

The incremental model

Prepared by Dr. Trisiladevi C. Nagavi, CS & E, SJCE, Mysore

4.1 Prescriptive Models : ii) Incremental Model

Ex 1: word-processing software

- basic file management, editing, and document production functions in the first increment;
- more sophisticated editing and document production capabilities in the second increment;
- spelling and grammar checking in the third increment;
- advanced page layout capability in the fourth increment.

Ex 2

4.1 Prescriptive Models : ii) Incremental Model

- **Useful**
  - **Less number** of staffing
  - If **core product is well received**, then additional staff are added.

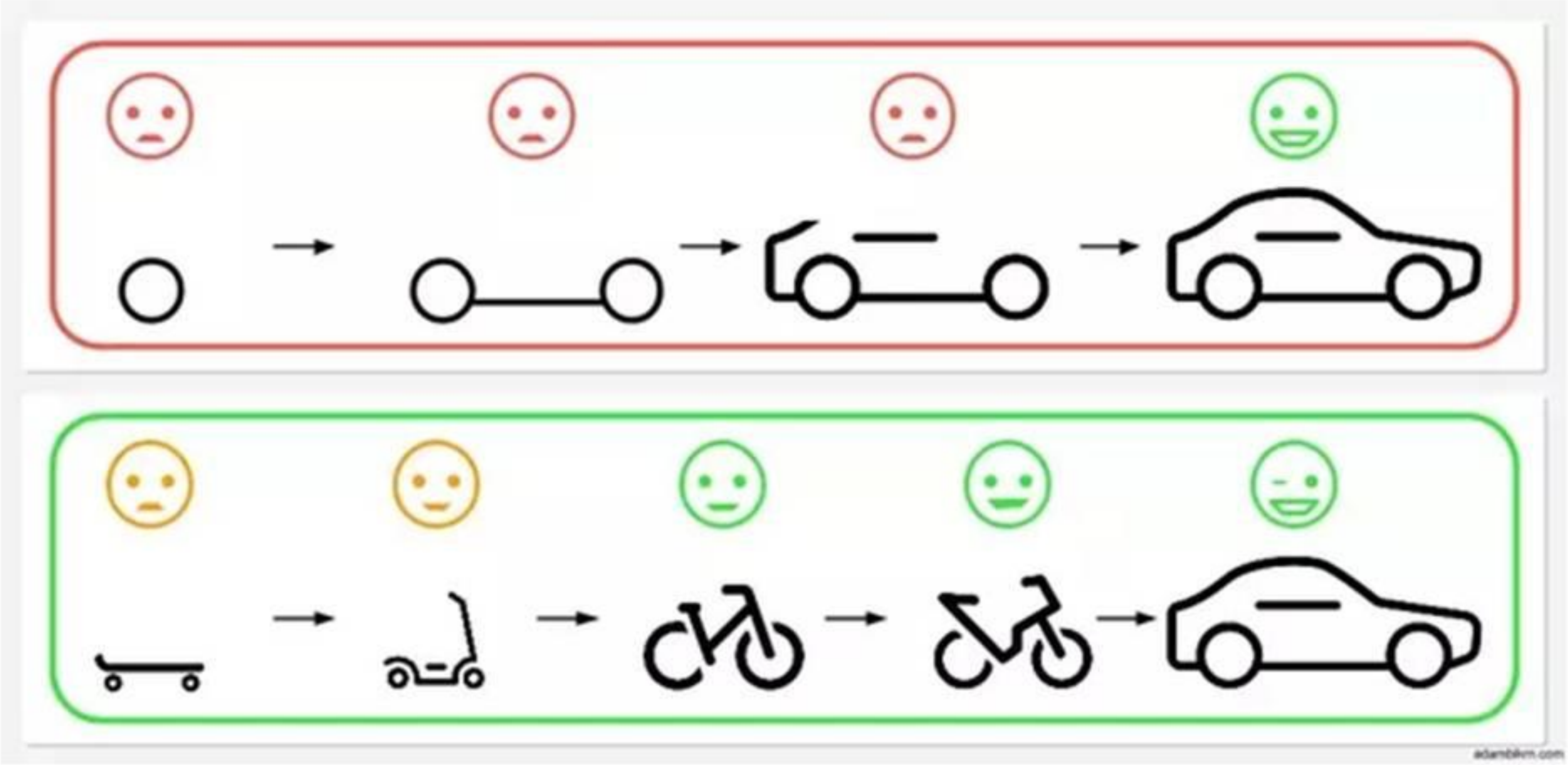
  - Increments are planned to manage **technical risks.**
  - **Ex:** System requires the **availability of new hardware** that is under development and whose delivery date is uncertain. Early increments delivered without hardware.

4.1 Prescriptive Models : iii) Evolutionary Models

- Software **evolves over time** as requirements change as development proceeds. **Limited version** is delivered to meet competitive pressure.
- Set of **core product requirements** is understood, but details and extension have yet to be defined.

- **Iterative enables** to develop increasingly more **complete version** of the software.
- Two types : **Prototyping and Spiral models**.

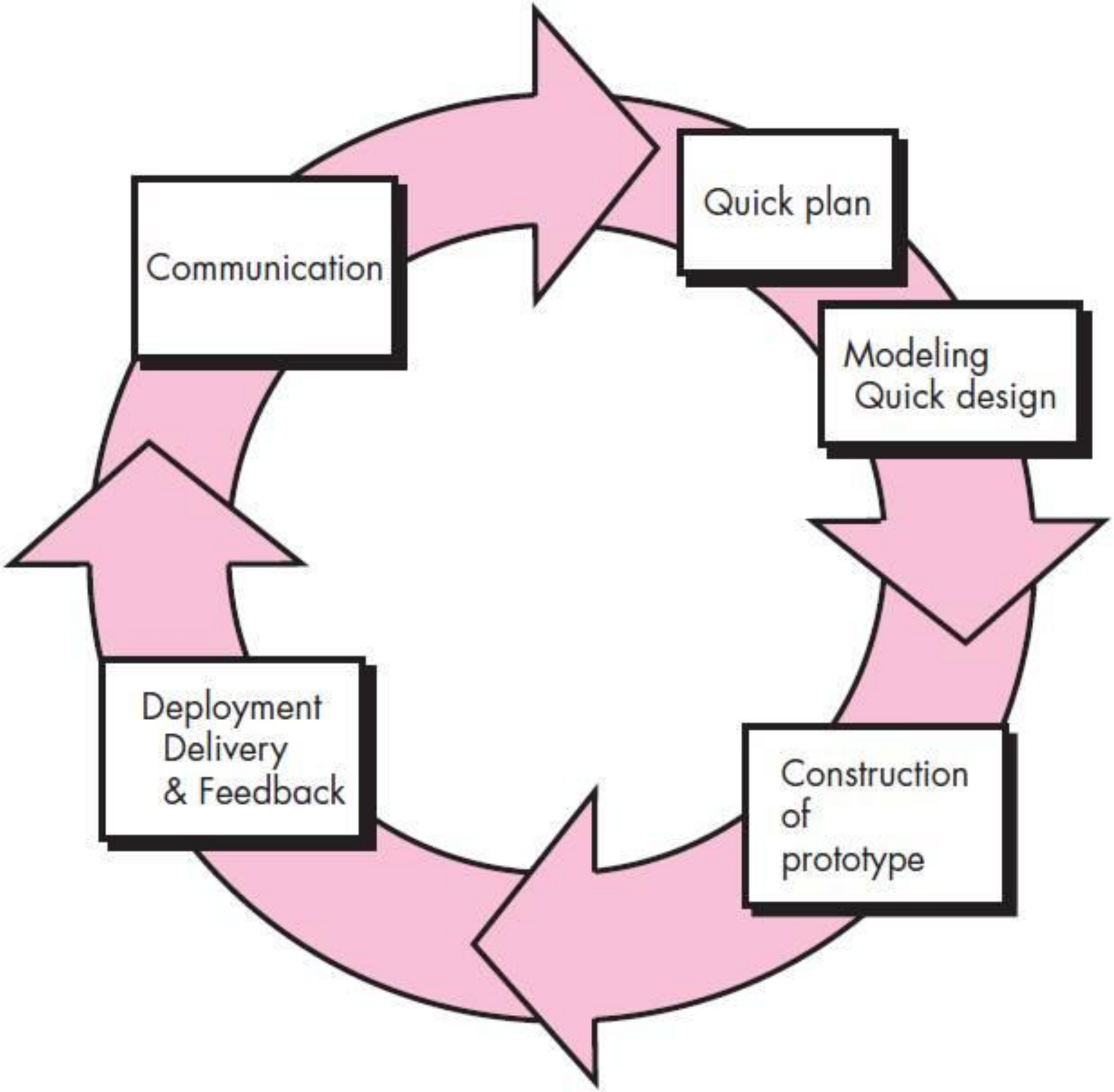
Incremental on the top and Iterative on the bottom



Prepared by Dr. Trisiladevi C. Nagavi, CS & E, SJCE, Mysore

4.1 Prescriptive Models : iii) Evolutionary Models

A) Prototyping

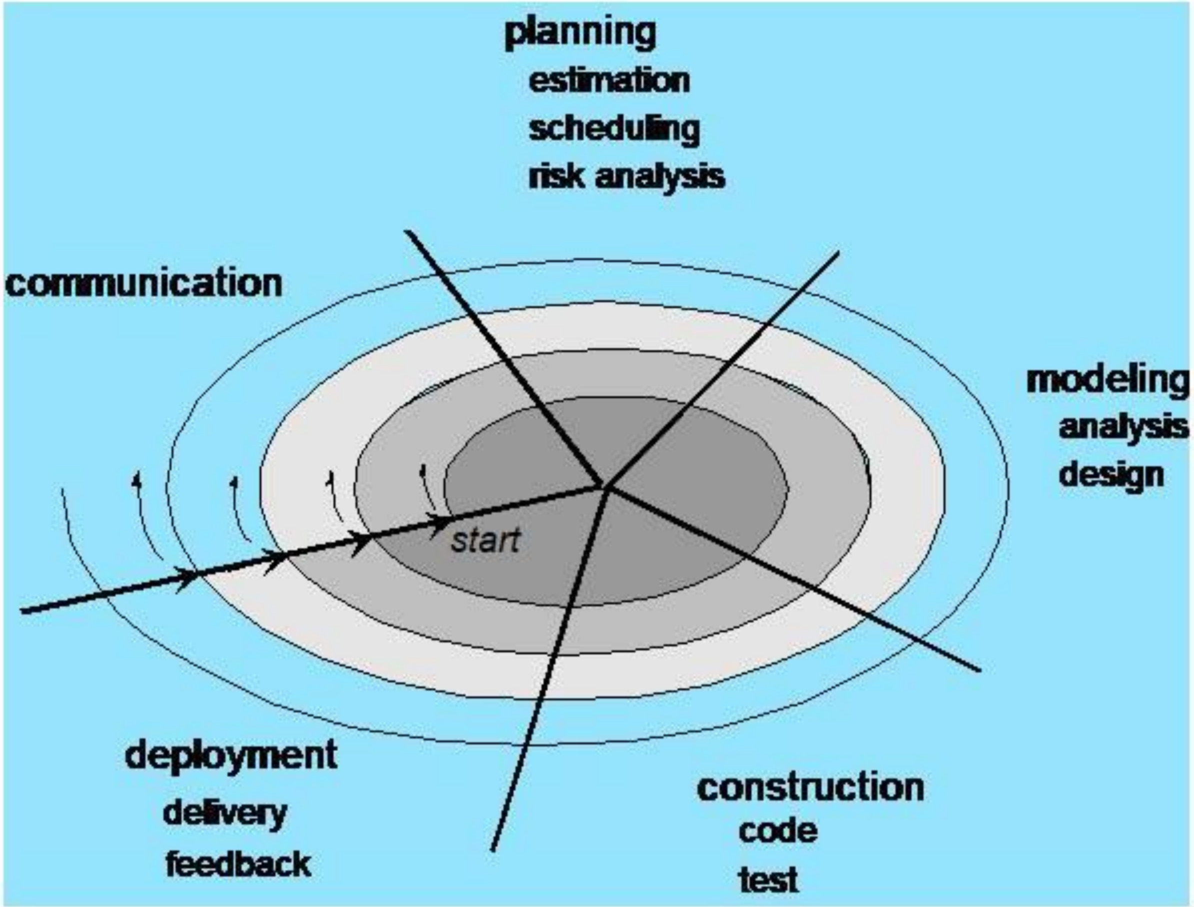


4.1 Prescriptive Models : iii) Evolutionary Models

A) Prototyping

  - Customer defines general objectives but does not identify detailed requirements.
  - Developer is unsure about the algorithm.
- **What step:**
  - Begins with communication with stakeholders to identify further definition.
  - Quick design leads prototype which will be deployed and evaluated.
- **Advantages:**
  - Stakeholders and software engineers to get a feel and idea of actual system immediately.
  - Prototype evolves into Product if not throw away.
- **Limitations:**
  - Engineers may make compromises in order to get a prototype working quickly.

4.1 Prescriptive Models : iii) Evolutionary Models
B) Spiral



17

4.1 Prescriptive Models : iii) Evolutionary Models
B) Spiral

- Couples iterative prototyping with aspects of waterfall model and is risk-driven.
- Two features: **i) cyclic approach** for incrementally growing system's degree of definition and implementation with decreasing risk.
- **ii) anchor point milestones** for ensuring stakeholder commitment to feasible and satisfactory system solutions.

- Series of **evolutionary releases** are delivered. Early iterations, the release might be **model or prototype.**
- Later iterations, **more complete version** of the engineered system are produced.

4.1 Prescriptive Models : iii) Evolutionary Models
B) Spiral

- Divided into **framework activities**, with each evolutionary process, team performs activities around the spiral in a clockwise direction, beginning at the **center**.
- **Anchor point milestones—a** *combination of work products and conditions*.

- 1st spiral results in specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software.
- Each pass results in **adjustments to the project plan.** (Cost, schedule, number of iterations).
- Unlike other process models, spiral model can be adapted to apply **throughout the life** of software.

4.1 Prescriptive Models : iii) Evolutionary Models

B) Spiral

- **Three spirals: initial, middle, later**
- 1st spiral : "**concept development project**" that starts at the core of the spiral and continues for multiple iterations.
- If concept is developed product: process proceeds outward and **"new product development project**" commences.
- Later circuit around the spiral : represents "**product enhancement project.**"

4.1 Prescriptive Models : iii) Evolutionary Models
B) Spiral

- Uses:
  - Remains operative until the software is retired. There are times when the process is dormant, but whenever a change is initiated, the process starts at the appropriate entry point (e.g., product enhancement).
  - Good to develop large-scale system as software evolves as the process progresses and risk should be understood and properly reacted to. Prototyping is used to reduce risk.

- However, it may be difficult to convince customers that it is **controllable as it demands considerable risk assessment expertise**.

4.1 Specialized Process Models

iv) Concurrent Model

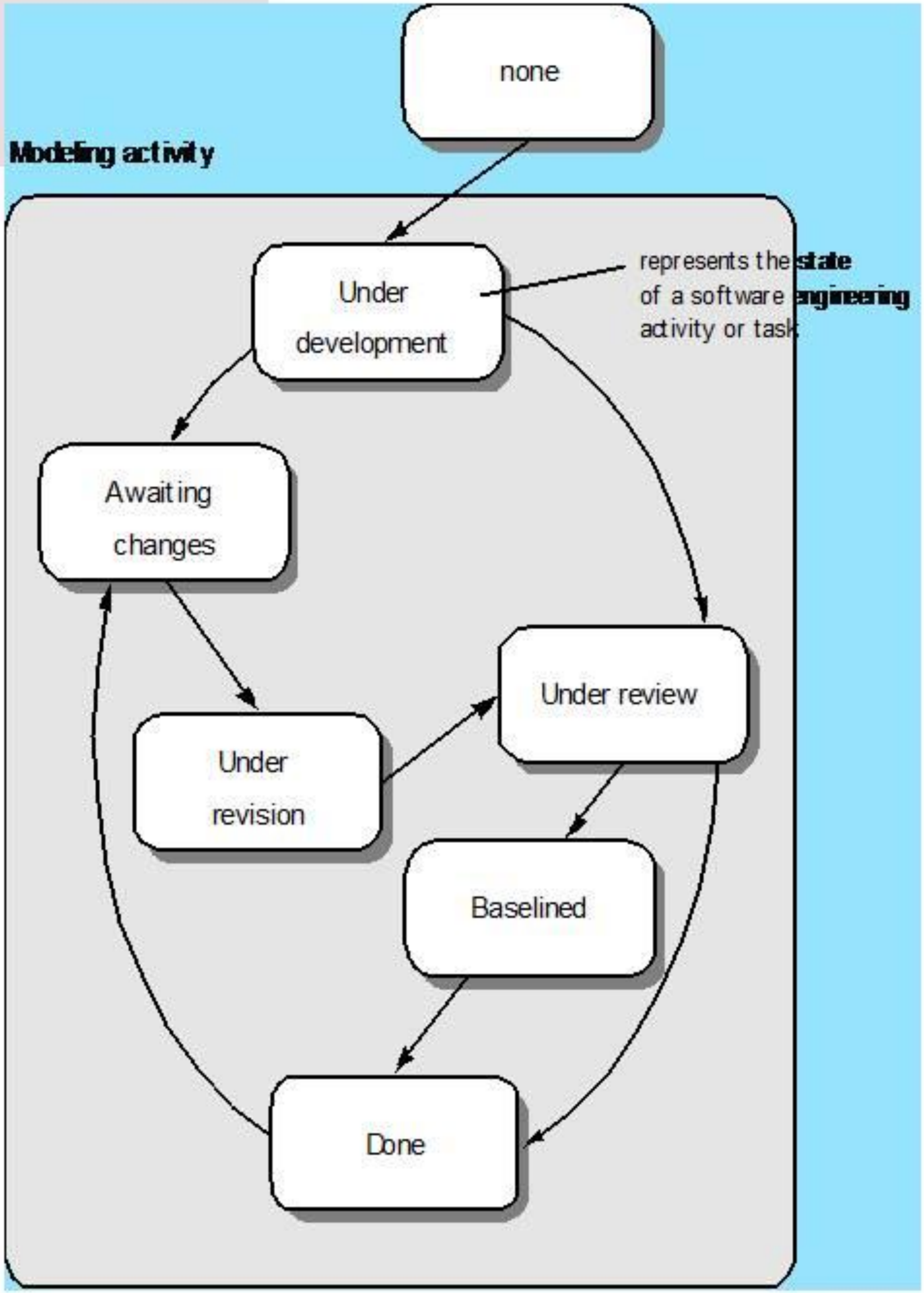


FIGURE 2.8

One element of the concurrent process model

4.1 Specialized Process Models iv) Concurrent Model

- Allows a software team to represent **iterative and concurrent elements** of any of the process models. For example, the modeling activity defined for the spiral model is accomplished by invoking one or more of the following actions: prototyping, analysis and design.

- Rather than confining software engineering activities, actions and tasks to a sequence of events, it **defines a process network.** Each activity, action or task on the network exists simultaneously with other activities, actions or tasks. Events generated at one point trigger transitions among the states.
- Used for all type of softwares.

4.1 Prescriptive Models : v) A Final Word on Evolutionary Proccesses

- **Three Concerns on Evolutionary Processes**
- i) **Prototyping** poses a problem to project planning because **uncertain number of cycles** required.
- Ii) Does not establish the **maximum speed of the evolution**. If the evolution occur too fast, without a period of **relaxation**, it is certain that the process will fall into chaos. On the other hand if the speed is too slow then **productivity** could be affected.
- Iii) Processes should be focused on flexibility and extensibility rather than on high quality. We should prioritize the speed of the development over zero defects. **Extending the development in order to reach high quality** could result in a late delivery of the product when the opportunity niche has disappeared.

- Focusing on **flexibility, extensibility, and speed of development over high quality** is difficult.