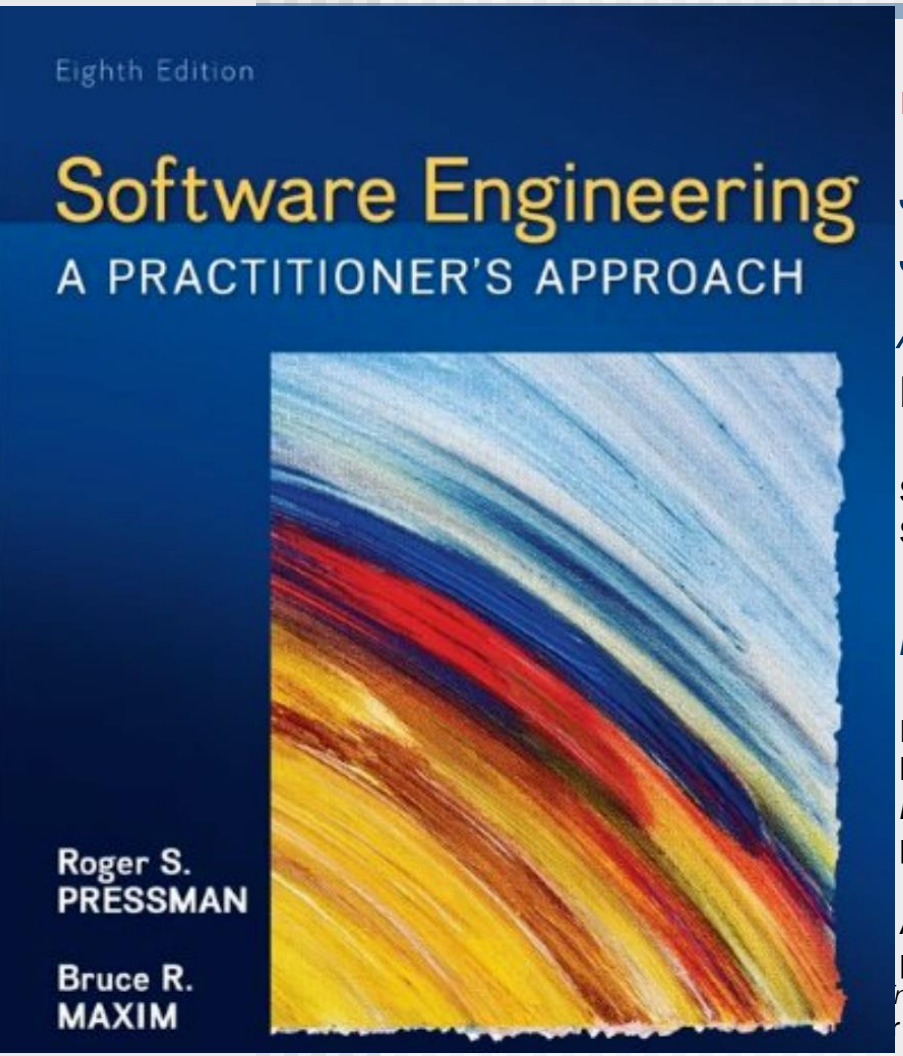
Chapter 4 & Chapter 5 Important Concepts



- **Process Models &Agile**

Slide Set to accompany
Software Engineering: A Practitioner's Approach, 8/e

by Roger S. Pressman and Bruce R. Maxim

Prescriptive Models

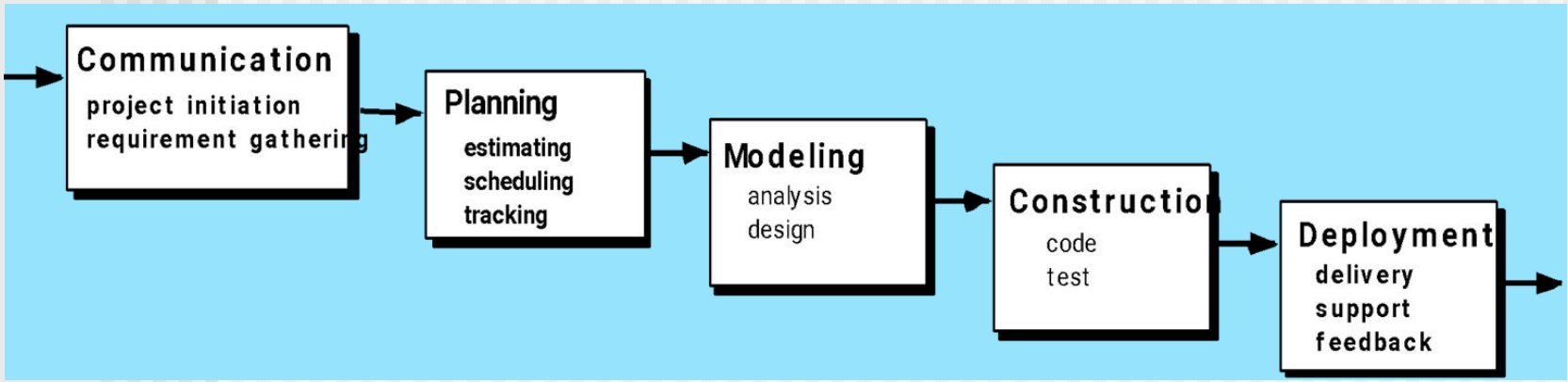
- Prescriptive process models advocate an orderly approach to software engineering

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?

Process Model

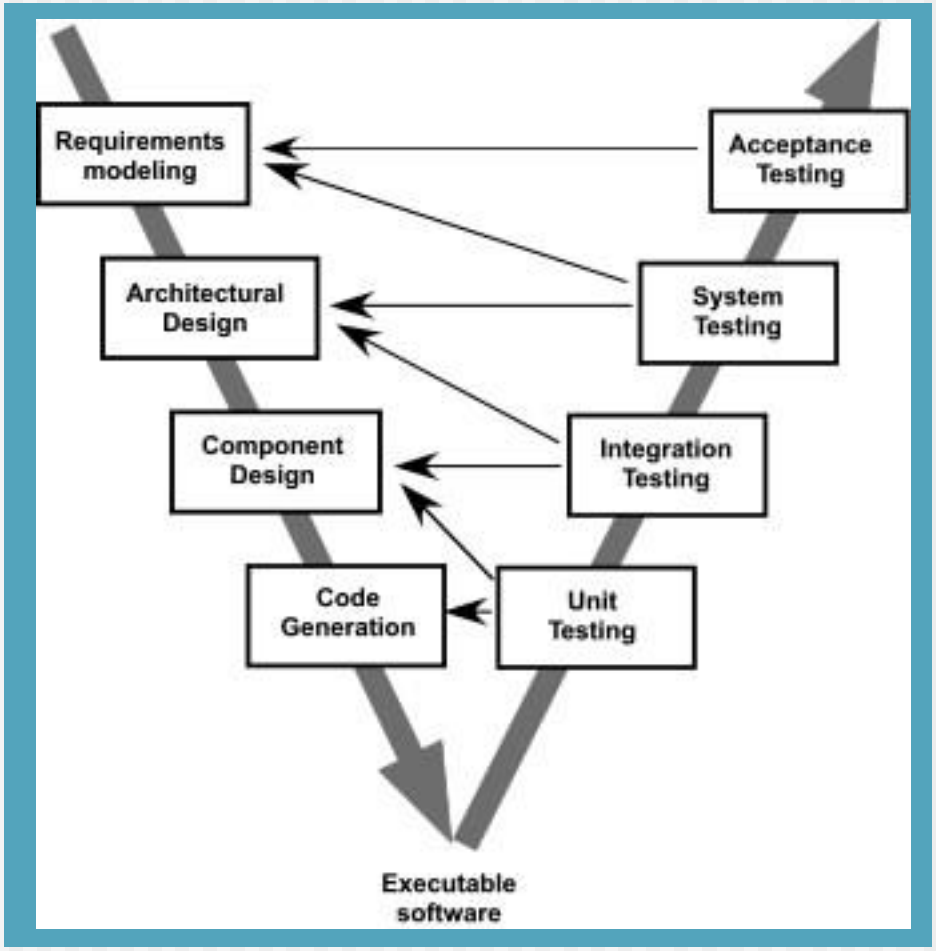
1. The Waterfall Model



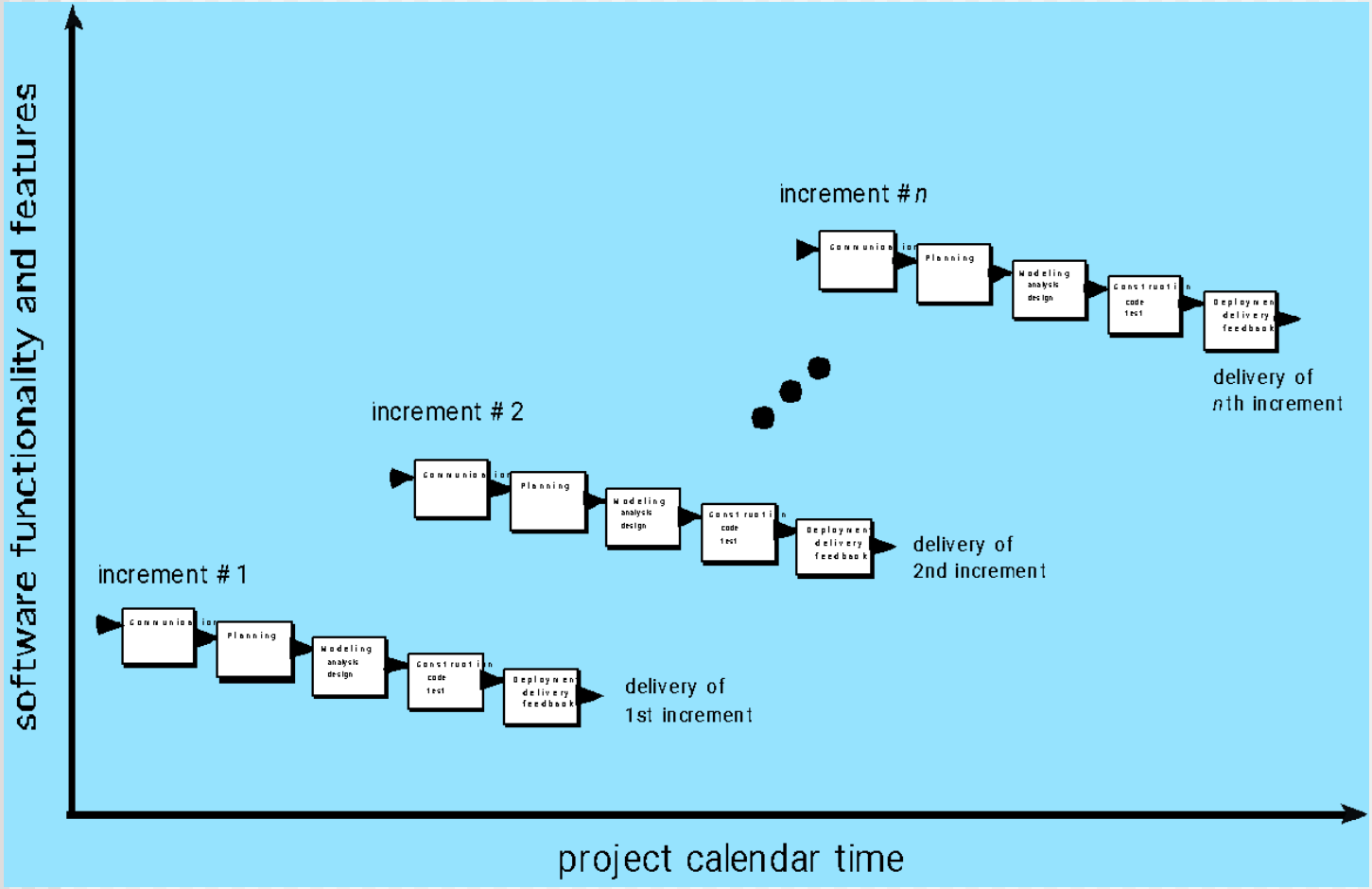
- Requirements are very well documented, clear and fixed.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- The project is short.

- Disadvantage - *it does not allow for much reflection or revision.*

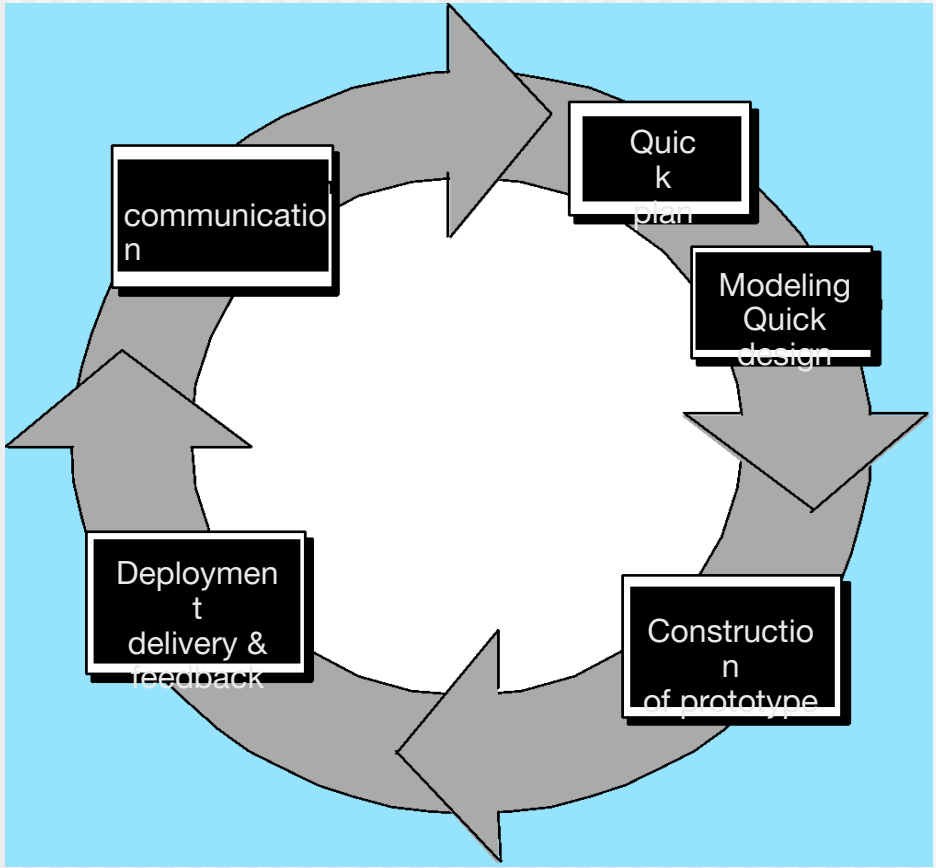
2. The V-Model



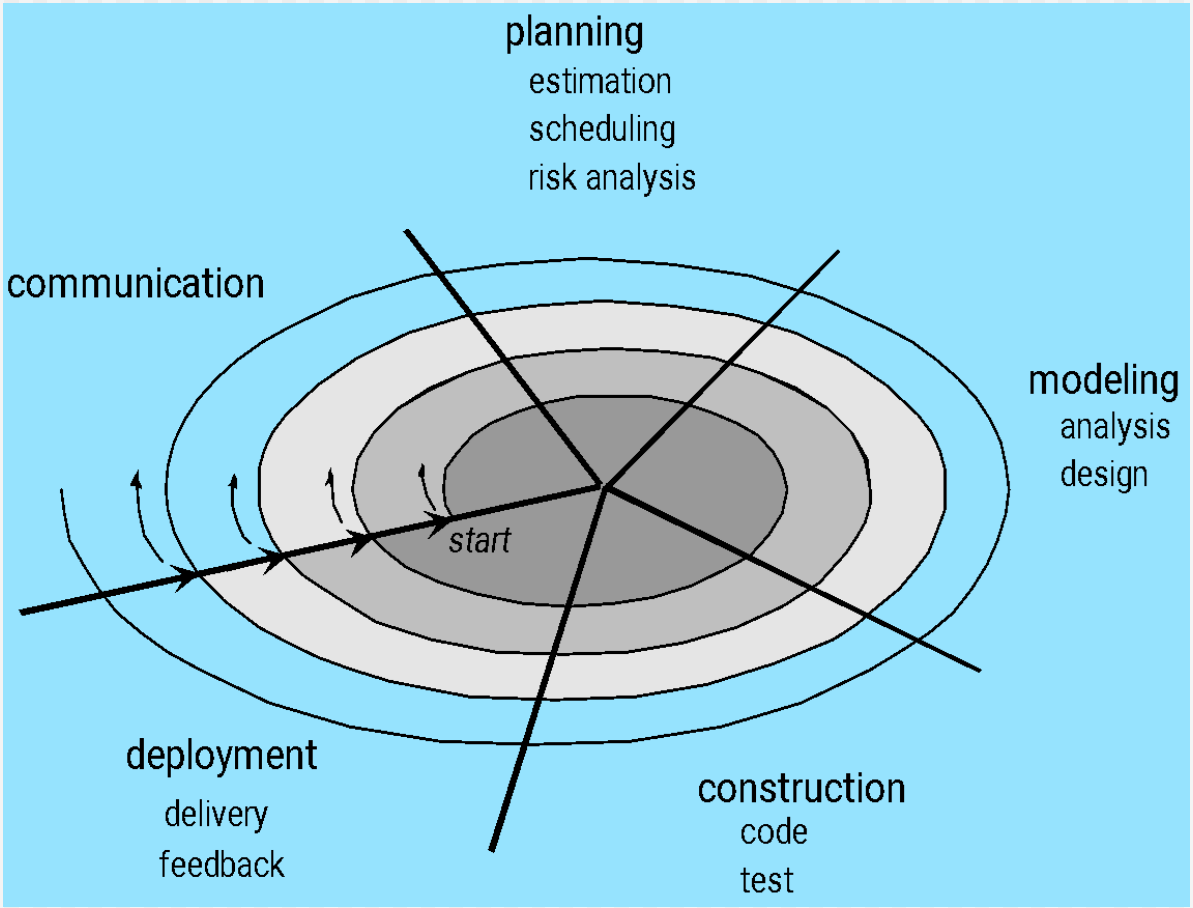
3. The Incremental Model



4. Evolutionary Models: a. Prototyping

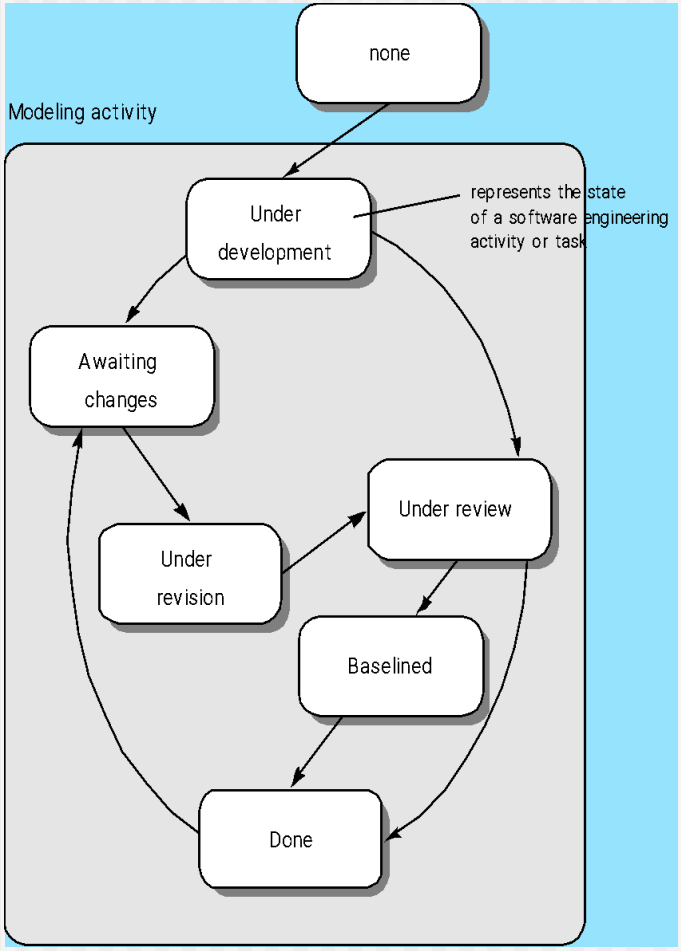


4. Evolutionary Models: b. The Spiral

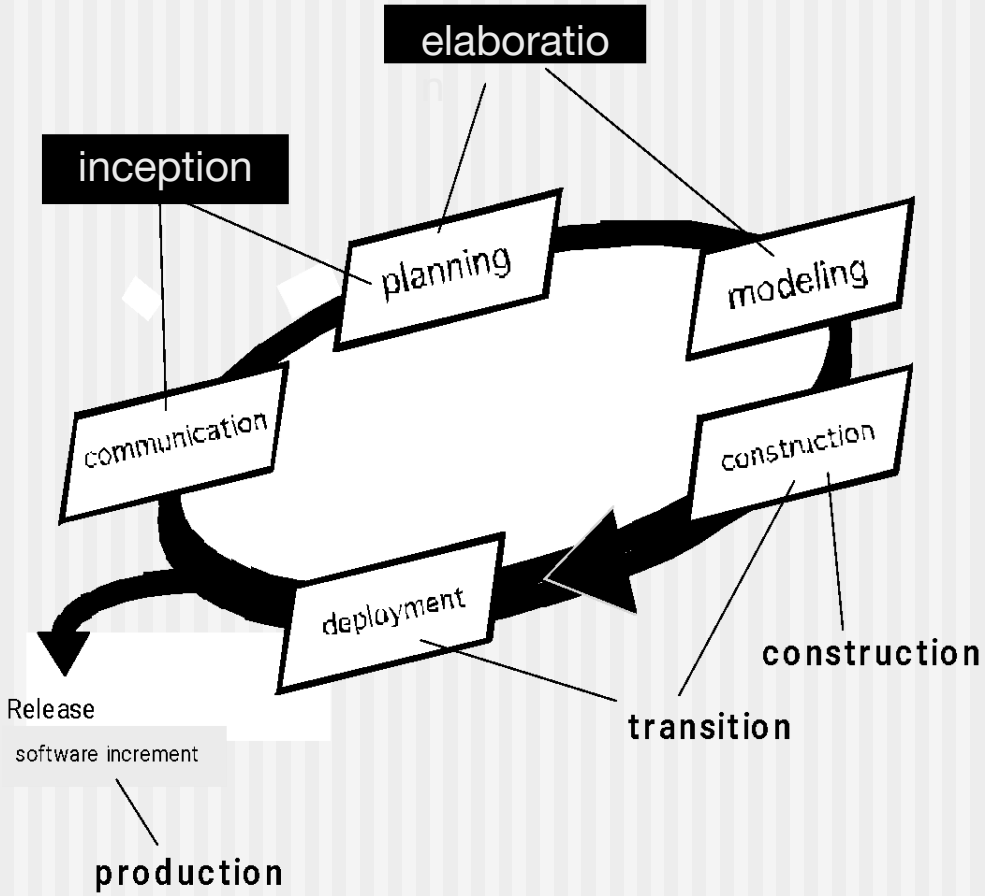


4. Evolutionary Models: c. Concurrent

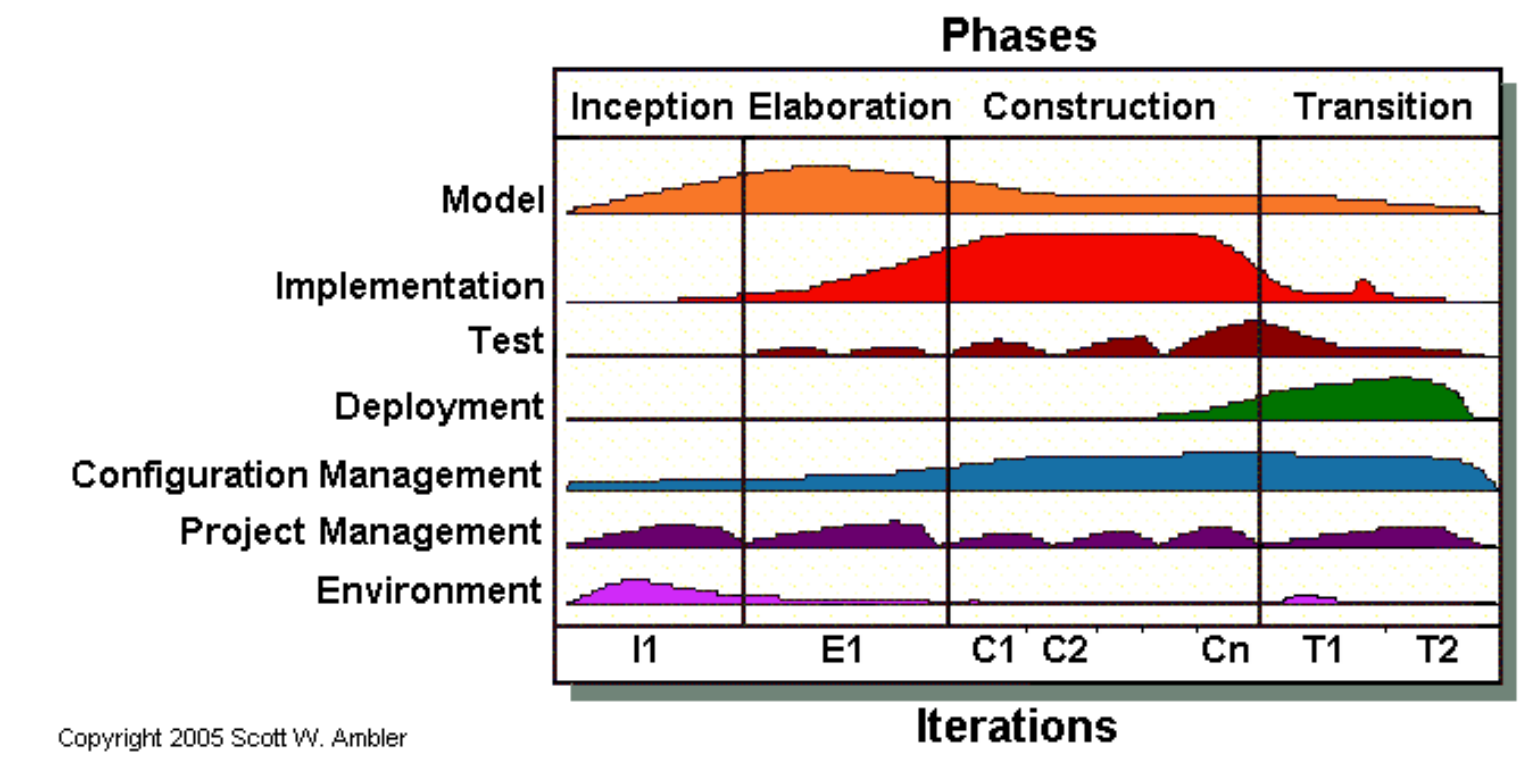
- In real life the software development activities do not take place in sequence
- Most activities will be going on concurrently but reside in different states.
- The states will change when some event occurs
- All the activities are shown along with their states at any point of time.
- As time goes on the states of the activities will change.



The Unified Process (UP)



The Unified Process (UP)



Chapter 5

- **Agile Development**

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 8/e

by Roger S. Pressman and Bruce R. Maxim

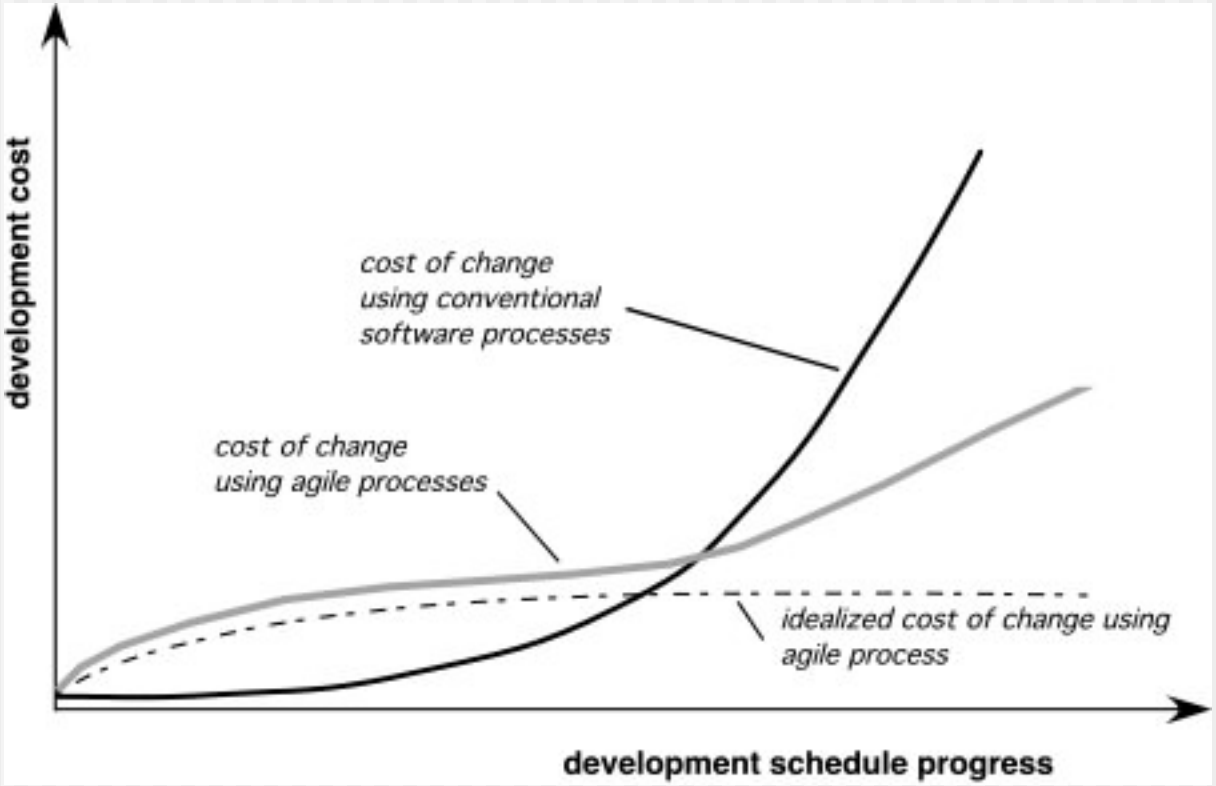
What is "Agility"?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding …

- Rapid, incremental delivery of software

Agility and the Cost of Change

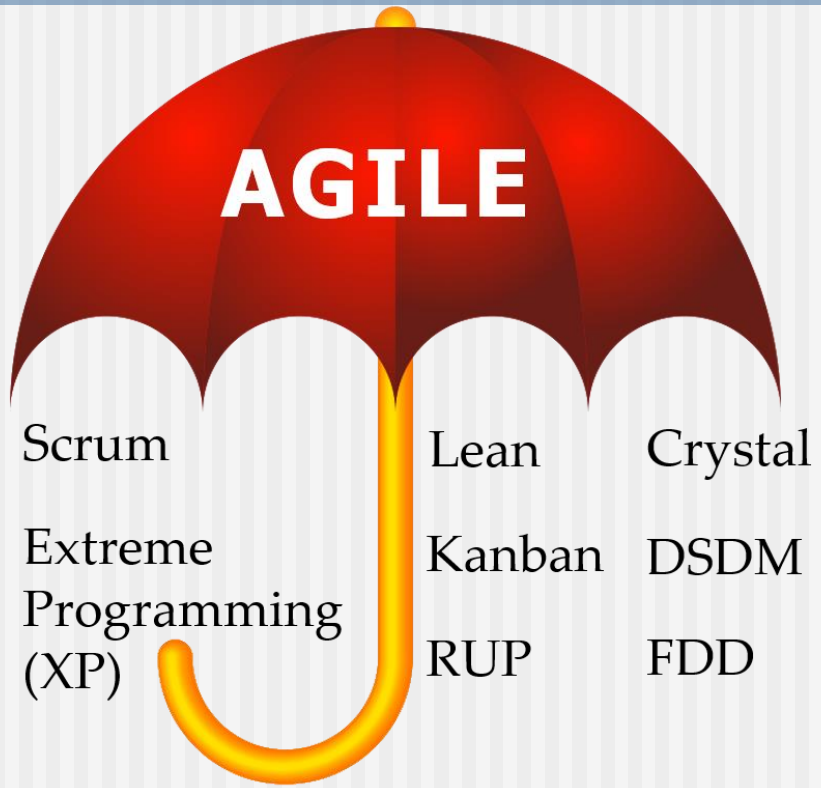


The cost of change increases nonlinearly as the project progresses

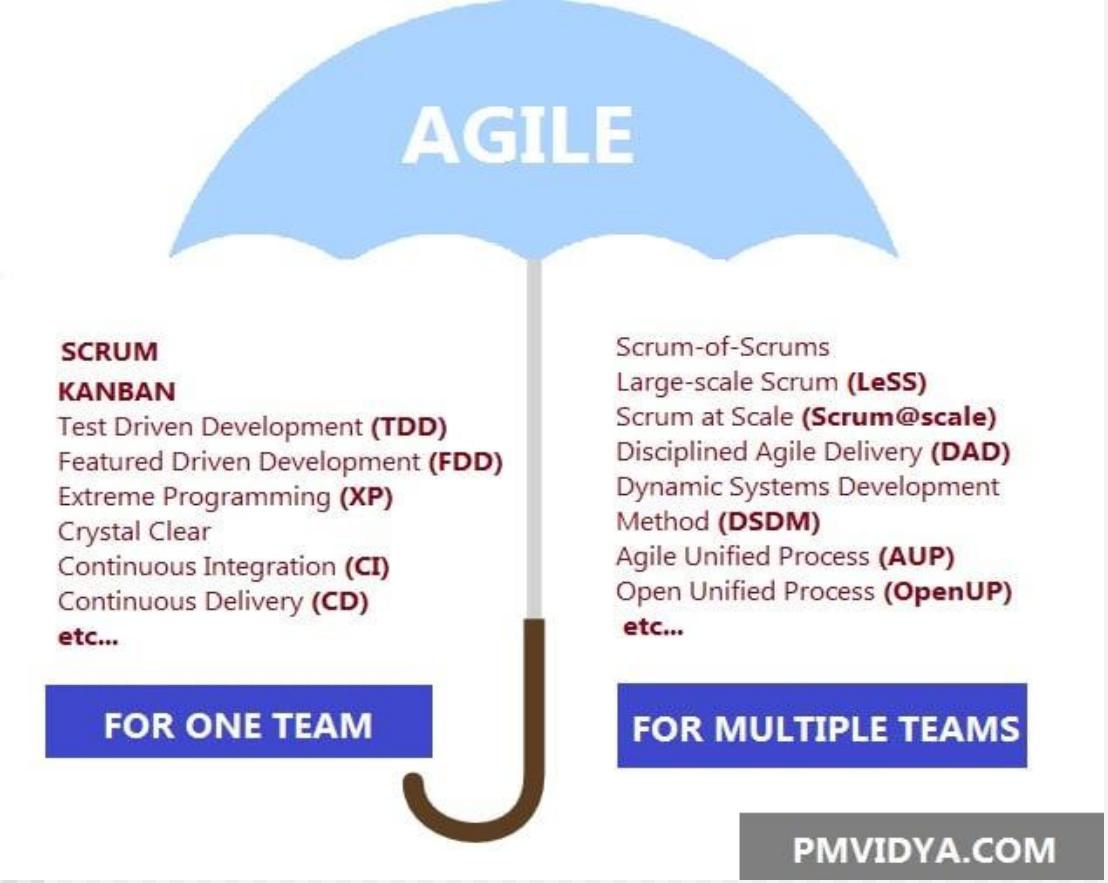
An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adapts as changes occur

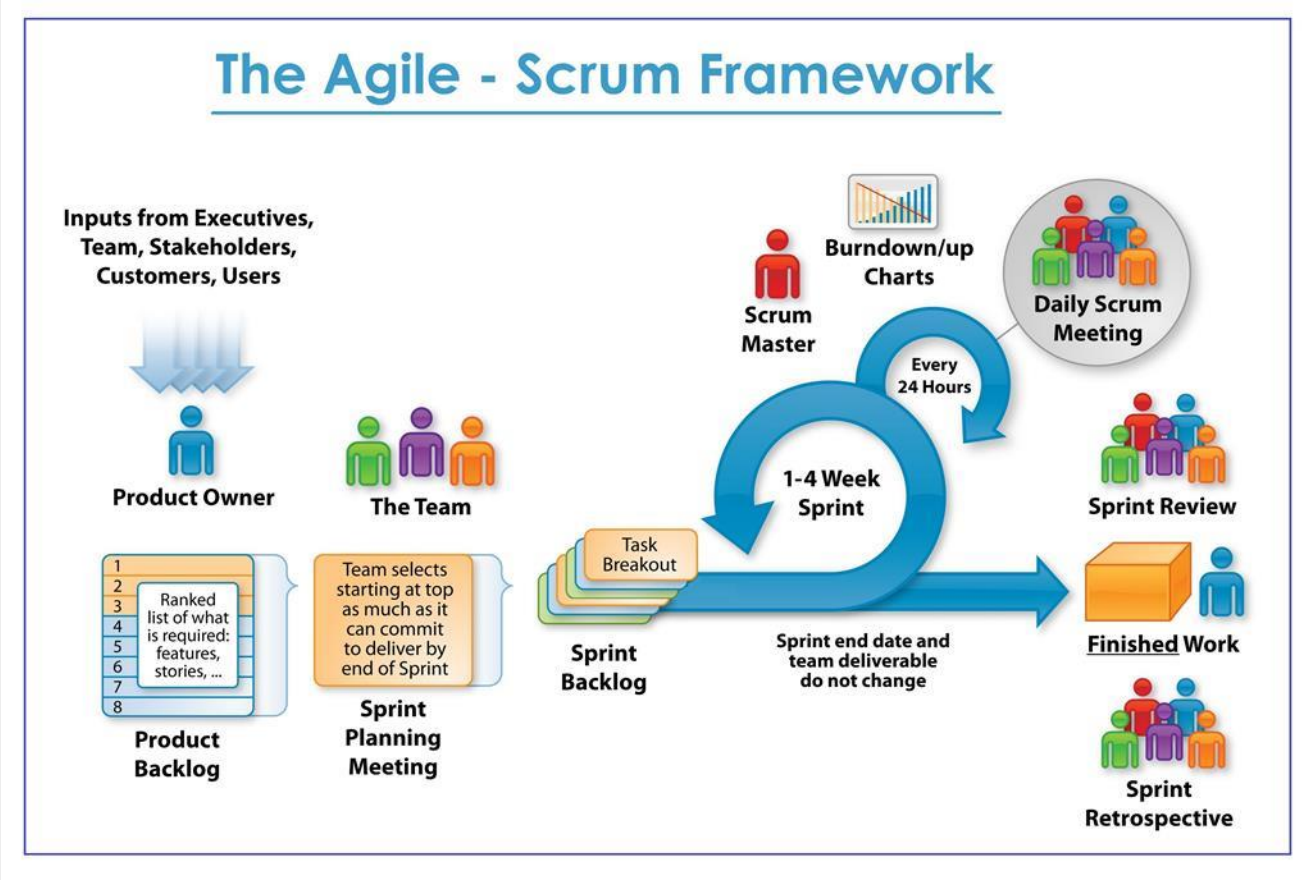
Agile Frameworks

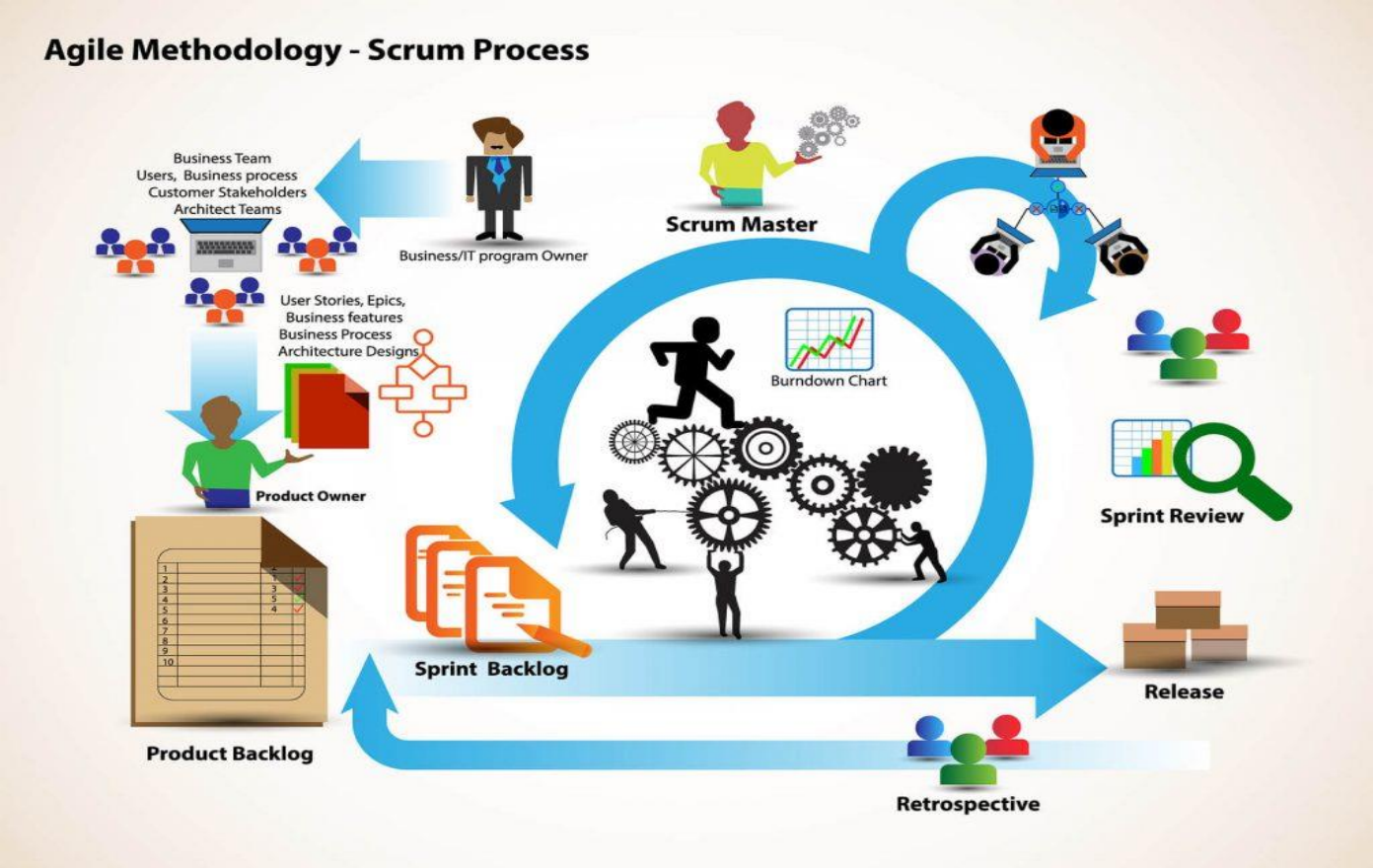


More Agile Framework



Scrum




Agile Methodology - Scrum Process
Business Team
Users, Business process
Customer Stakeholders
Architect Teams
Business/IT program Owner
Scrum Master
User Stories, Epics,
Business features
Business Process
Architecture Designs
Burndown Chart
Product Owner
Sprint Review
Sprint Backlog
Release
Product Backlog
Retrospective

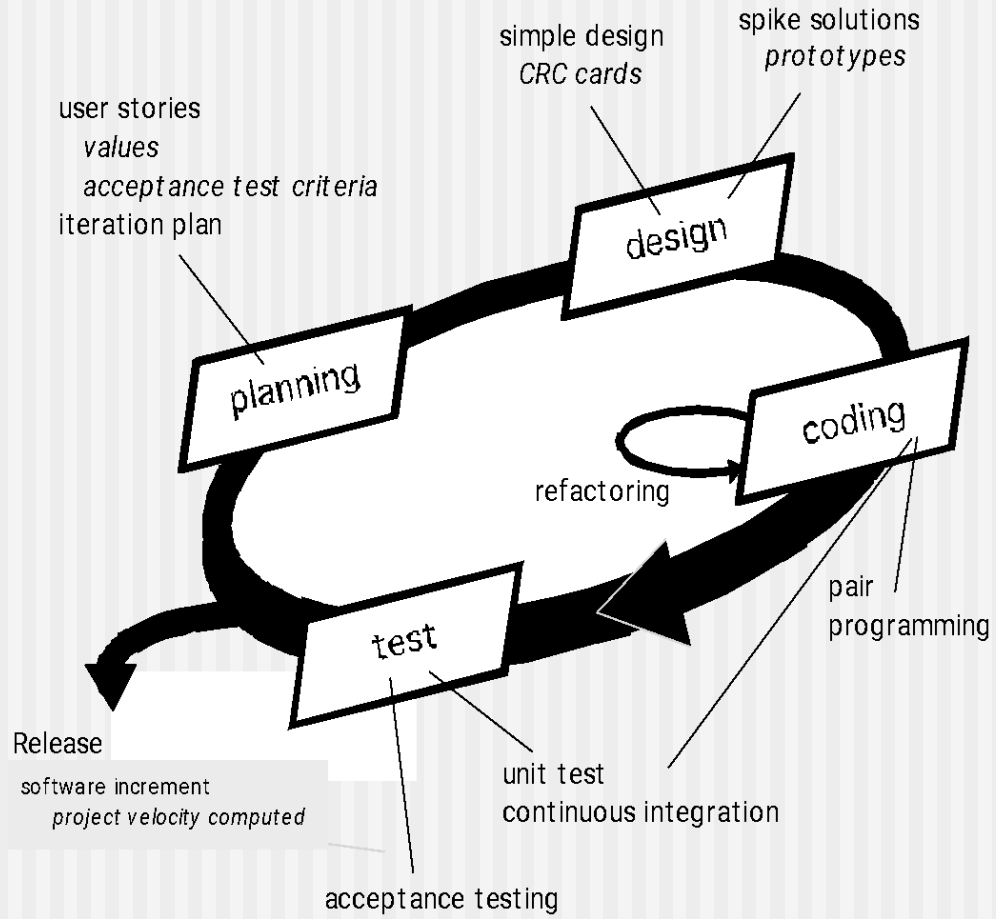
Extreme Programming (XP)

- The most widely used agile process, originally proposed by Kent Beck
- **XP Planning**
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment is made on delivery date
  - After the first increment "project velocity" is used to help define subsequent delivery dates for other increments

Extreme Programming (XP)

- **XP Design**
  - Follows the KISS principle
  - Encourage the use of CRC cards (see Chapter 8)
  - For difficult design problems, suggests the creation of "spike solutions"—a design prototype
  - Encourages "refactoring"—an iterative refinement of the internal program design
- **XP Coding**
  - Recommends the construction of a unit test for a store *before* coding commences
  - Encourages "pair programming"
- **XP Testing**
  - All unit tests are executed daily
  - "Acceptance tests" are defined by the customer and excuted to assess customer visible functionality

Extreme Programming (XP)



Review at Home