Lecture02: Software Processes

EGCI340: SOFTWARE DESIGN

Still remember? (1)

Legacy System

- A method, technology, computer system, or application program that continues to be used, typically because it still functions for the users' needs
- Reasons for keeping legacy system
  - System requires near-constant availability, so it cannot be taken out of service
  - Cost of designing a new system with a similar availability level is high
    - Customers' accounts system in banks
    - Computer reservation systems,
    - Air traffic control
    - Energy distribution (power grids)
    - Nuclear power plants
    - Military defense installations

Still remember? (2)

- What are the attributes of good software?
  - Maintainability, Dependability, Efficiency, and Acceptability
- What are 4 activities of software process?
  - Specification, Development, Validation and Evolution
- What is CASE Tools?

Outline

- Process Framework
- Model Integration
- Process Patterns
- Process Assessment
- Personal and Team Models
- Process Models

Process Framework

A process framework establishes the foundation for a complete software process by

"Identifying a small number of **framework activities** that are applicable to all software projects"

- Software design is one of software process activities

Generic Process Framework

- Communication
- Planning
- Modeling
- Construction
- Deployment

Planning

- Understanding why an information system should be built
- Determining how the project team will go about building

Project Initiation

- Technical feasibility (Can we build it?)
- Economic feasibility (Will it provide business value?)
- Organizational feasibility (If we build it, will it be used?)

Designing

Design phase decides how the system will operate in terms of:

- Hardware, software, and network infrastructure
- User interface, forms and reports
- Specific programs, databases, and files that will be needed.

Design phase has four steps:

- Design Strategy
- Architecture Design
- Database and File Specifications
- Program Design

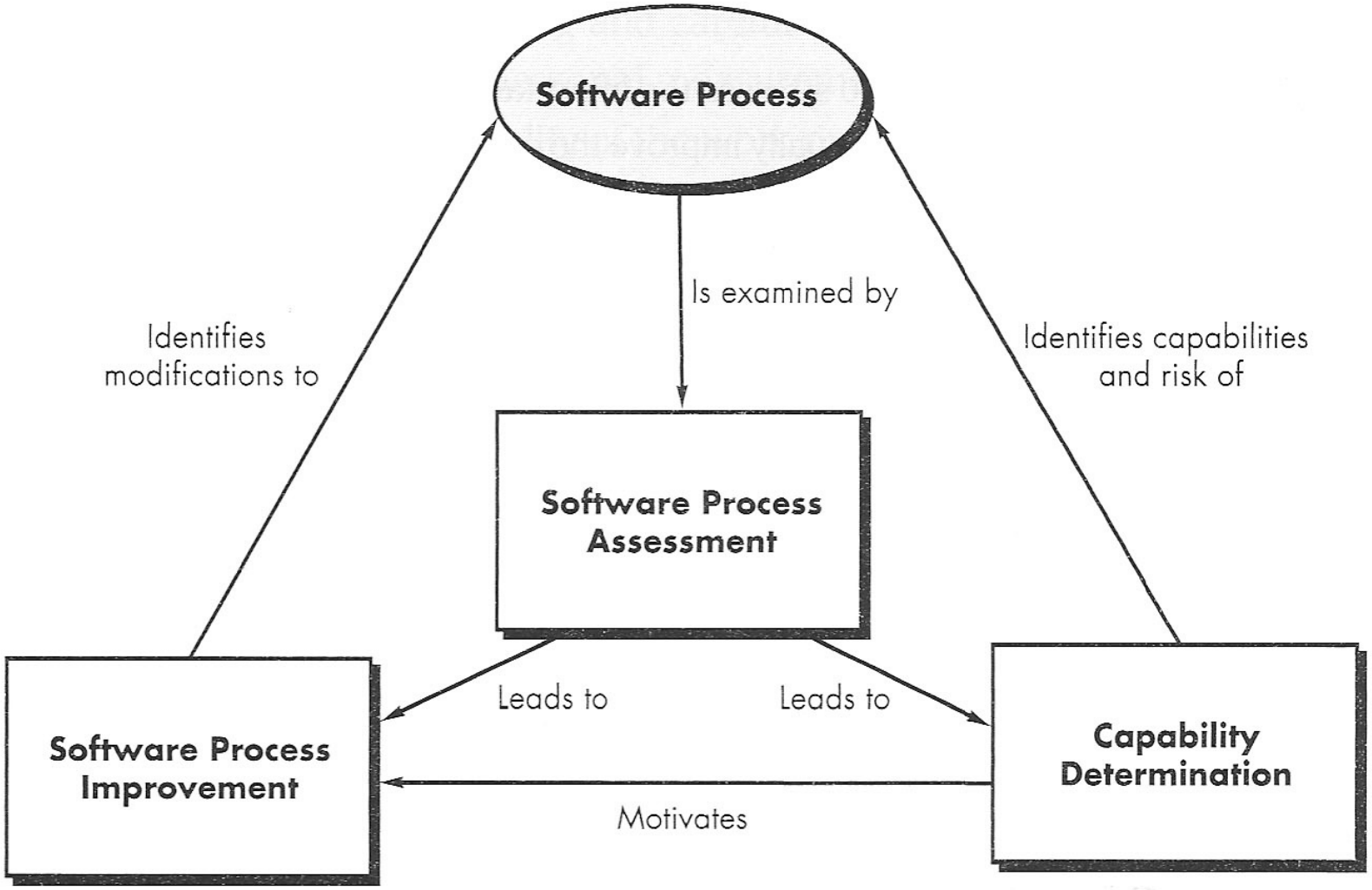
Process Assessment

The existence of a software process is no guarantee that software will be delivered on time

The process assessment depends on:

- Meet the customer's needs
- Exhibit the technical characteristics that will lead to long-term quality

Process Assessment (Cont.) [1]



Software Process Models

- A structured set of activities required to develop a software system

- A software process model is an abstract representation of a process
  - It presents a description of a process from some particular perspective

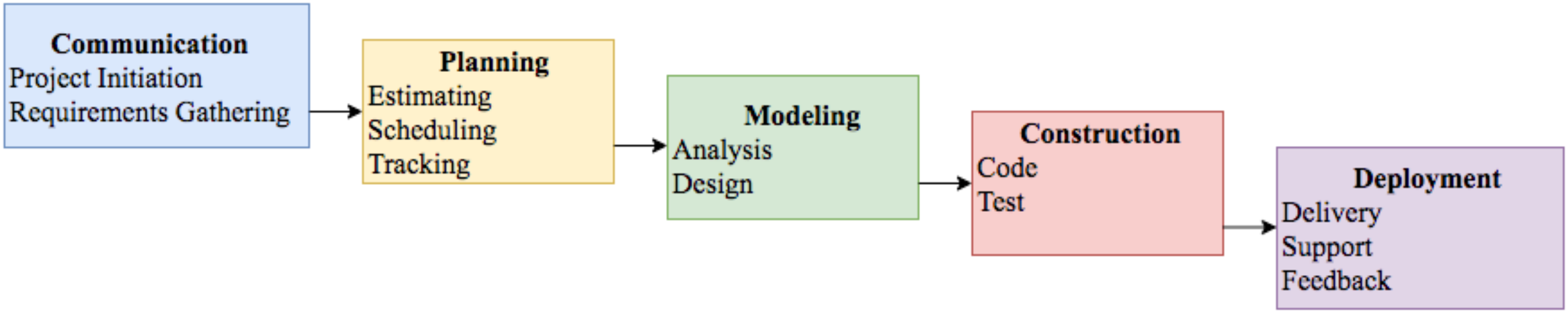
Waterfall Model

- **The classic life cycle**

- This model suggests a systematic and sequential approach to software development
  - The model begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment.

- The oldest paradigm for software engineering

Waterfall Model (Cont.)



Waterfall Model (Cont.)

Waterfall Model (Cont.)

Problem:

- Inflexible
- Difficult to respond of changing the customer requirements

This model is ***only appropriate when***:

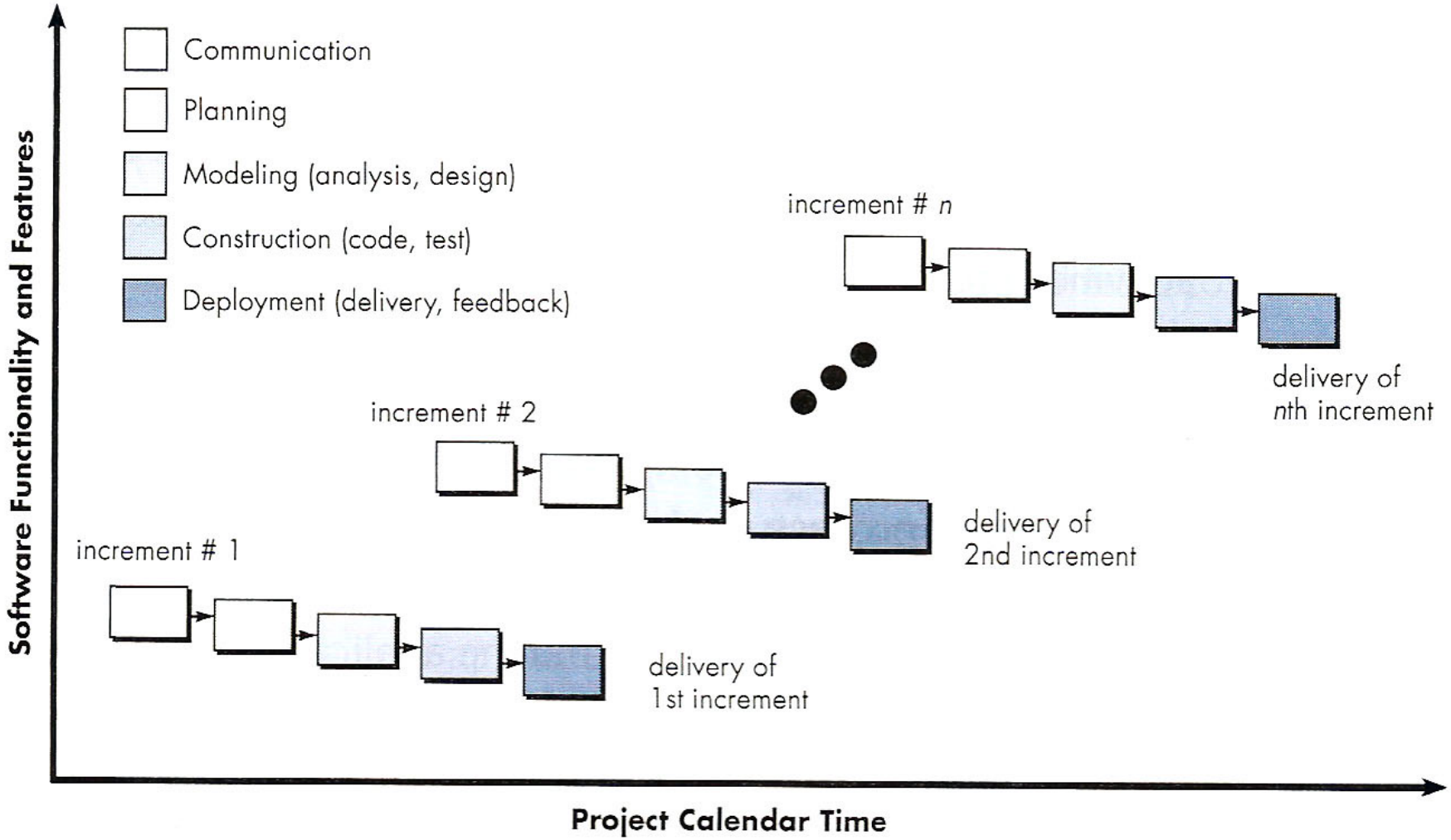
- The requirements are ***well-understood*** and ***changes will be fairly limited*** during the design process
- Requirements are stable

Incremental Process Models

Combines elements of the waterfall model applied in an iterative fashion

- Incremental model applies *linear sequences* as calendar time progresses
- Each linear sequence produces deliverable “increments” of the software

Incremental Process Models (Cont.) [1]



Incremental Process Models (Cont.)

For example: word-processing software developed using incremental paradigm might deliver:

- **1st increment:** Basic file management, editing, and document production functions in the first increment
- **2nd increment (More sophisticated):** editing, and document production capabilities
- **3rd increment:** spelling and grammar checking
- **4th increment:** advantaged page layout capabilities

Incremental development advantages

1. Custom values can be delivered with each increment
   - So system functionality is available earlier

2. Early increments act as a prototype to help elicit (or activate) requirements for later increments

3. Lower risk of overall project failure

4. Highest priority system services tend to receive the most testing

Incremental development advantages

1. Custom values can be delivered with each increment
   - So system functionality is available earlier

2. Early increments act as a prototype to help elicit (or activate) requirements for later increments

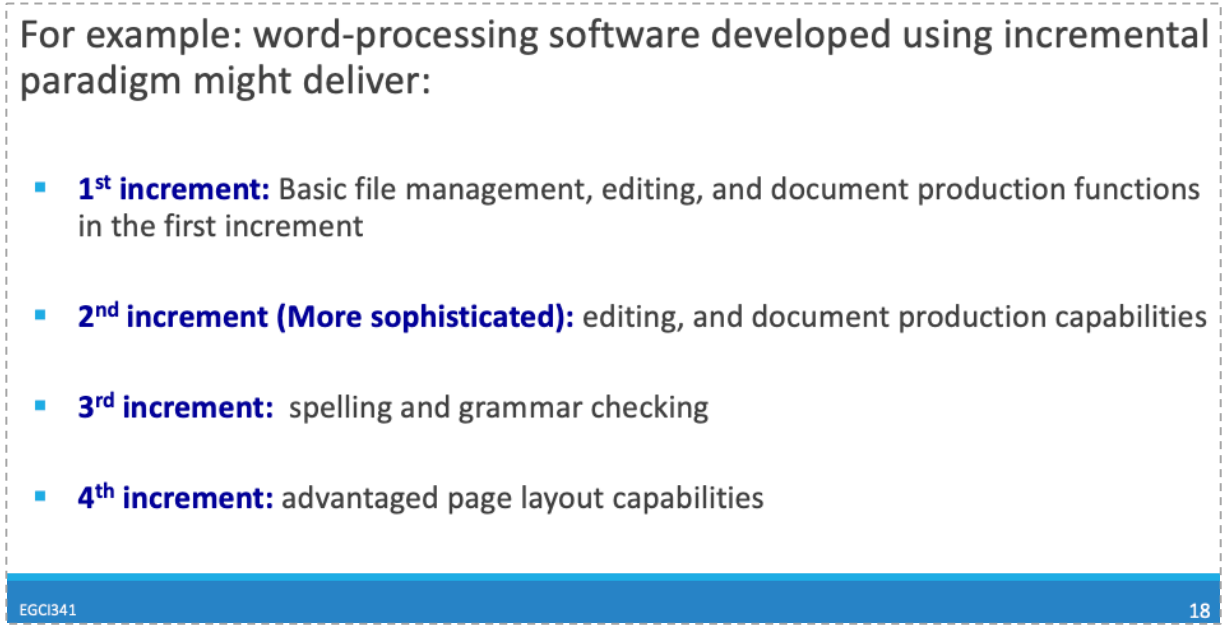
3. Lower risk of overall project failure

4. Highest priority system services tend to receive the most testing

In-class Assignment (1)

Write the list of features in each increment for your selected application

- 5 Groups (~6-7 students/ group)
- Max. no. of increments = 3
- See the example on page 18
- 10-15 Minutes



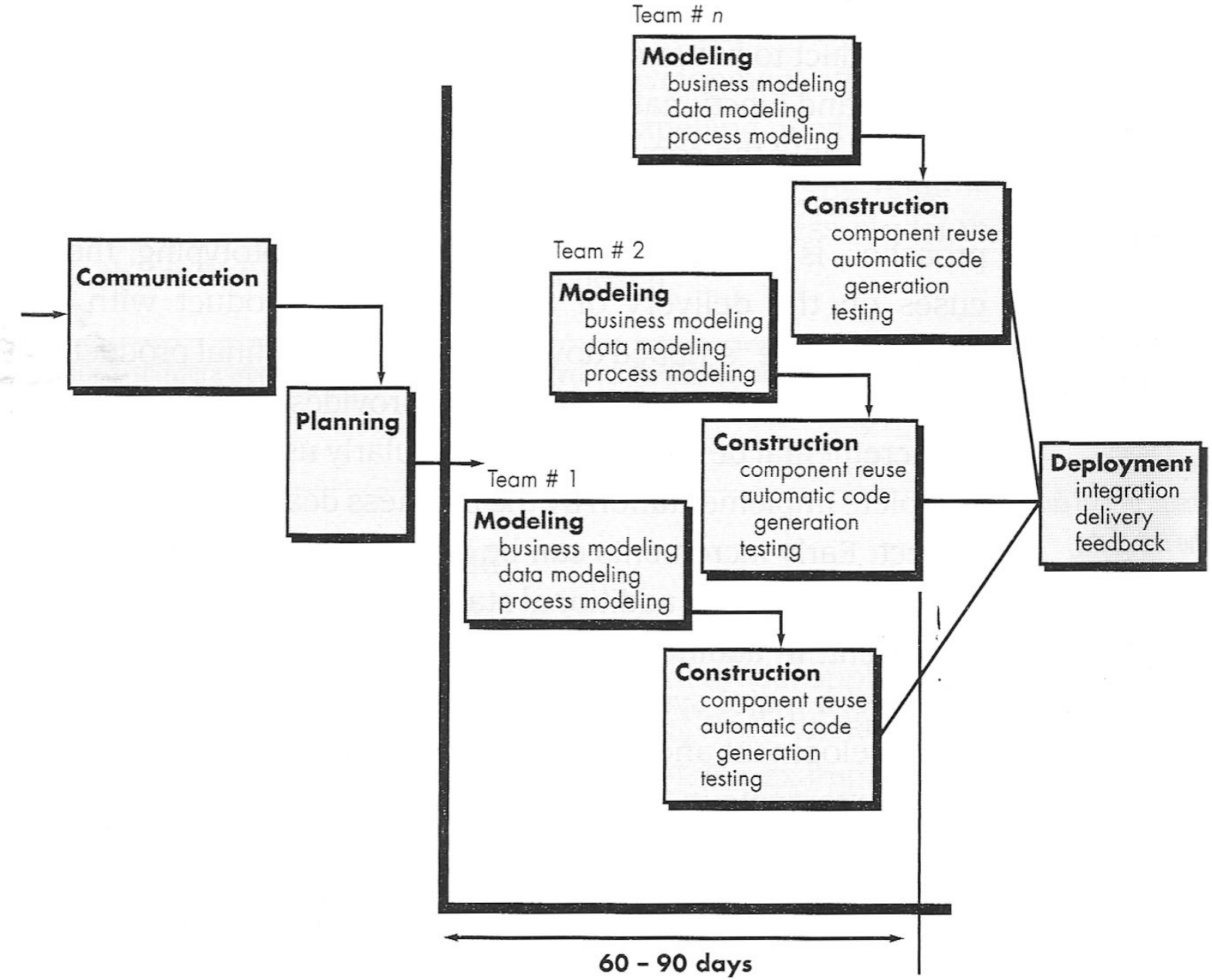
Rapid Application Development (RAD) Model

An incremental software process model that *emphasizes a short development cycle*

The RAD model is a “high-speed” adaptation of the waterfall model

- Using *a component-based construction* approach

RAD Model (Cont.)



Evolutionary Process Model

Evolutionary models are iterative

- They are characterized in a manner that enables software engineers to develop increasingly more complete versions of the software

Objective is:

- To work with ***customers***
- To evolve a final system from an initial outline specification
  - Start with well-understood requirements and,
  - Add new features as proposed by the customer

(1) Prototyping

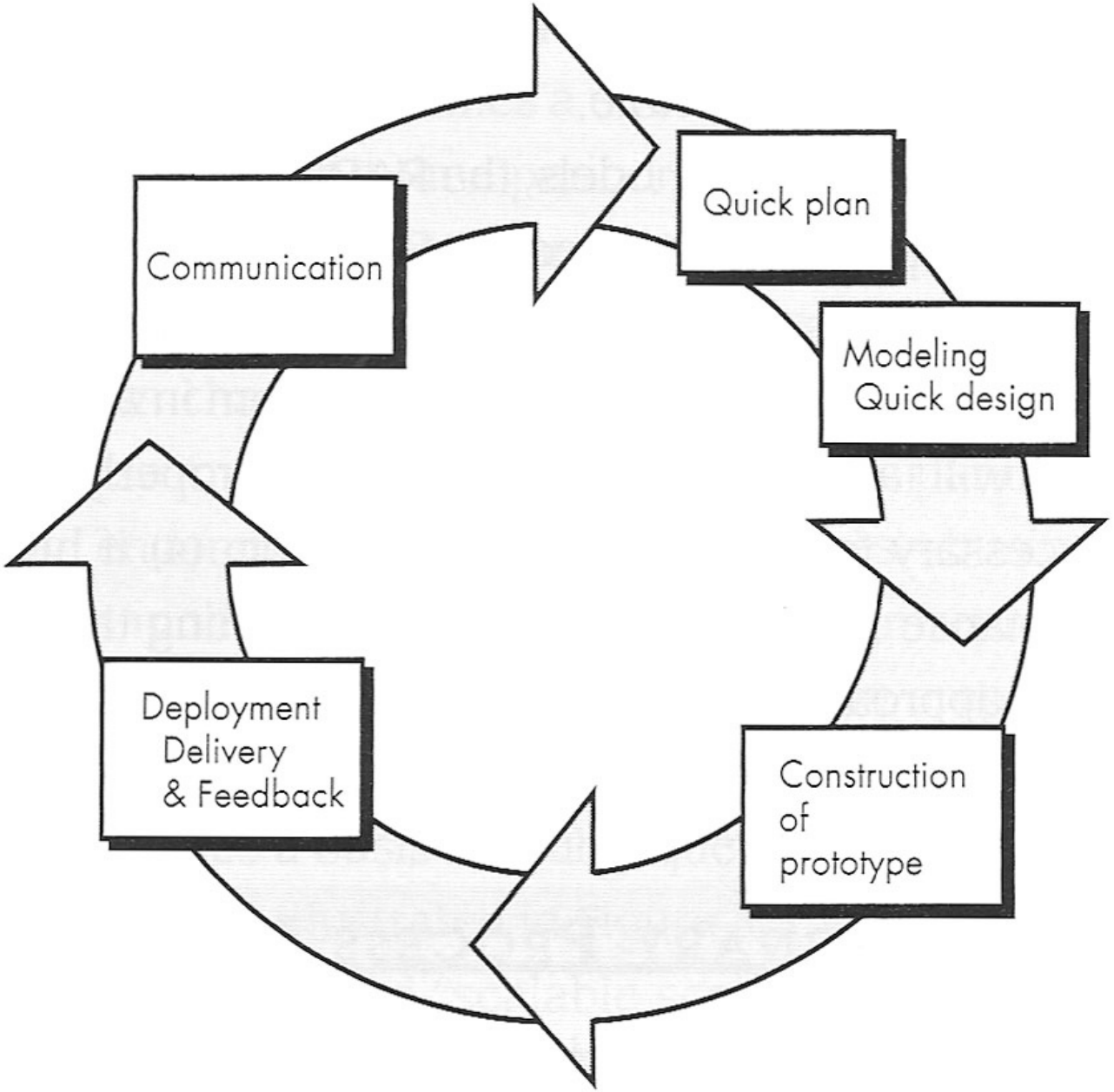
Customer defines a set of general objectives for software,

- But does not identify detailed input, processing, or output requirements

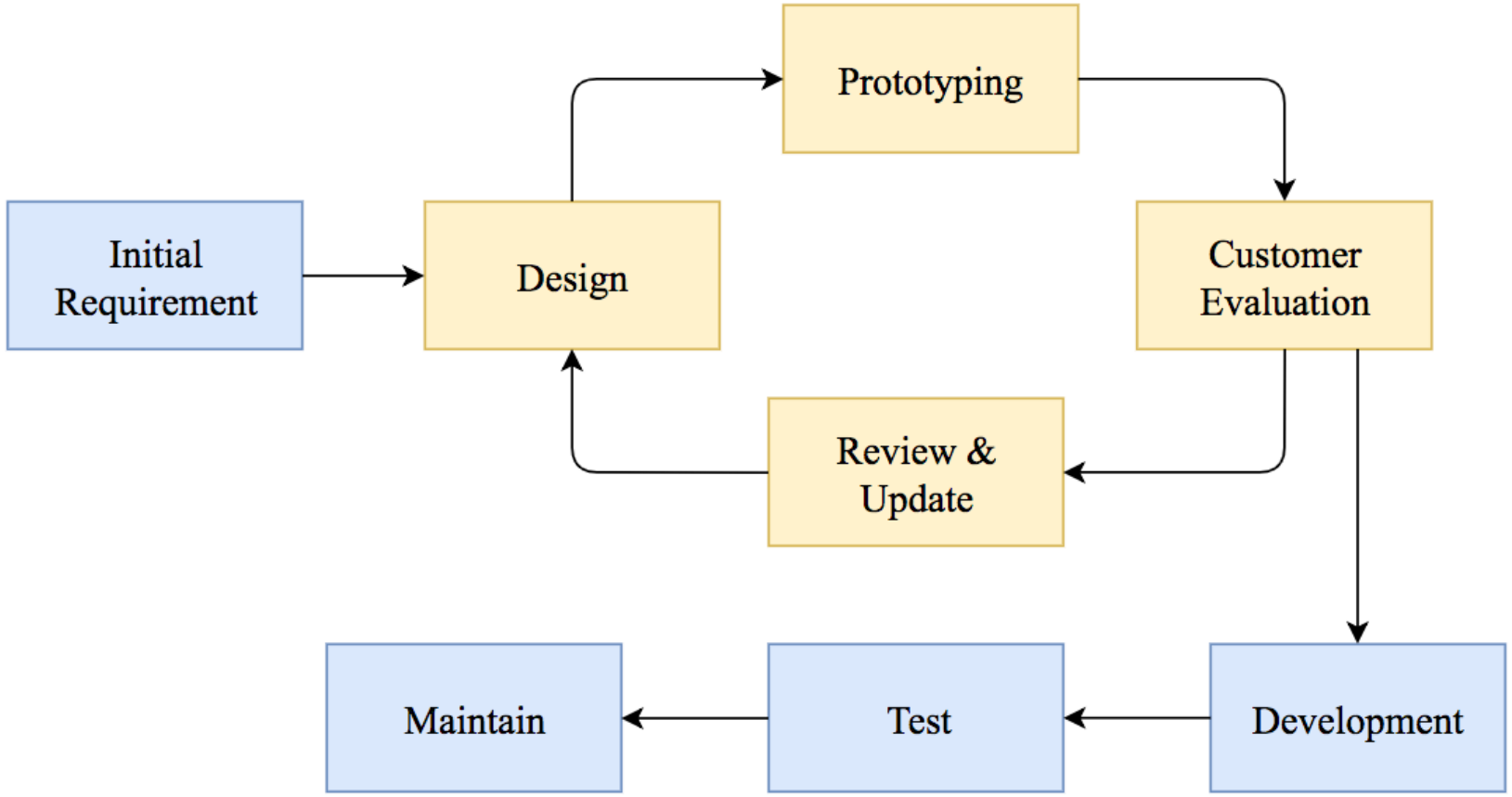
In other cases, the developer may be ***unsure of the efficiency of an algorithm***,

- The adaptability of an operating system, or the form that human-machine interaction should take

Prototyping (Cont.) [1]



Prototyping (Cont.)

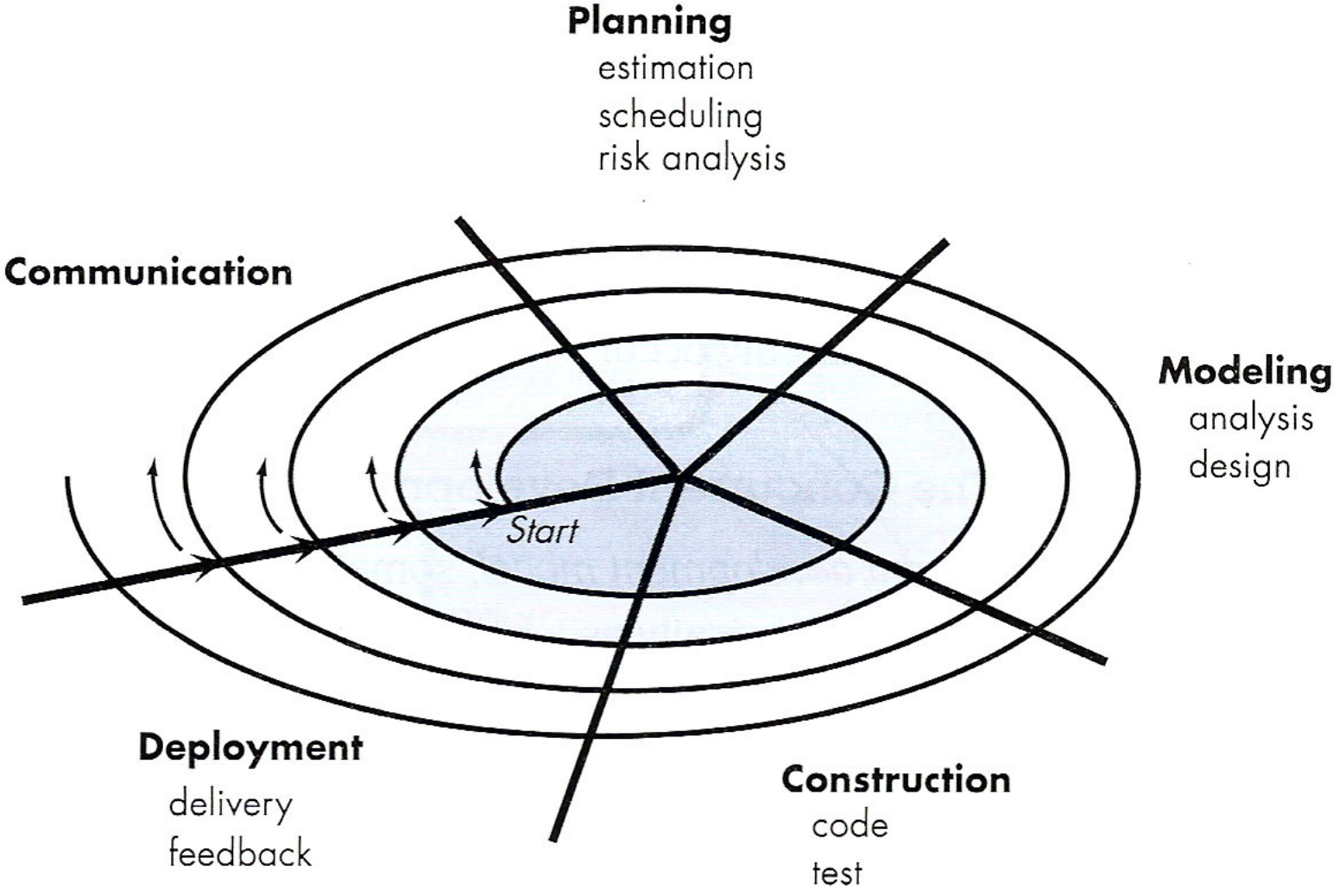


Prototyping Model

(2) Spiral Model

- The model that couples the *iterative nature of prototyping* with the controlled and systematic *aspects of the waterfall model*
- It provides the potential for rapid development of increasingly more complete versions of the software
- A spiral model is divided into a set of framework activities defined by the software engineering team
- Each of the framework activities represent one segment of the spiral path illustrated in next page

Spiral Model (Cont.) [1]



Spiral Model (Cont.)

- Software team performs activities that are implied by a circuit around the spiral in a clockwise direction, beginning at the center
- Risk is considered as each revolution is made
- Anchor point milestones are noted for each evolutionary pass
  - Anchor point milestones: a combination of work products and conditions that are attained along the path of the spiral

Component-based software engineering (CBSE)

Process stages

- Component analysis
- Requirements modification
- System design with reuse
- Development and integration

This approach is becoming increasingly used as component standards have emerged

Review

- What are the differences between Incremental Model and Spiral Model
- About Prototyping Model, how do the development team know that they can start to build the real software instead of another prototype.
- What are the differences between Rapid Application Development (RAD) and Incremental Model.

What is Agility?

Agile software engineering combines:

- A philosophy and a set of development guidelines

The philosophy encourages customer satisfaction and early incremental delivery of

- Software
- Small, highly motivated project team
- Minimal software engineering work products
- Overall development simplicity

What is Agile Process?

3 key assumptions about the majority of software projects:

1. It is difficult to predict in advance which software requirements will persist and which will change.
   - It is equally difficult to predict how customer priorities will change as a project proceeds

2. For many types of software, design and construction are interleaved
   - That is both activities should be performed in sequence so that design models are proven as they are created.
   - It is difficult to predict how much design is necessary before construction is used to prove the design.

3. Analysis, design, construction, and testing are not as predictable as we might like

Extreme Programming Model (XP)

Planning

- The planning activity begins with the creation of a set of ***stories*** called ***User Stories***
- The stories that describe required features and functionality for software to be build

Design

- XP design rigorously that keep its simple)

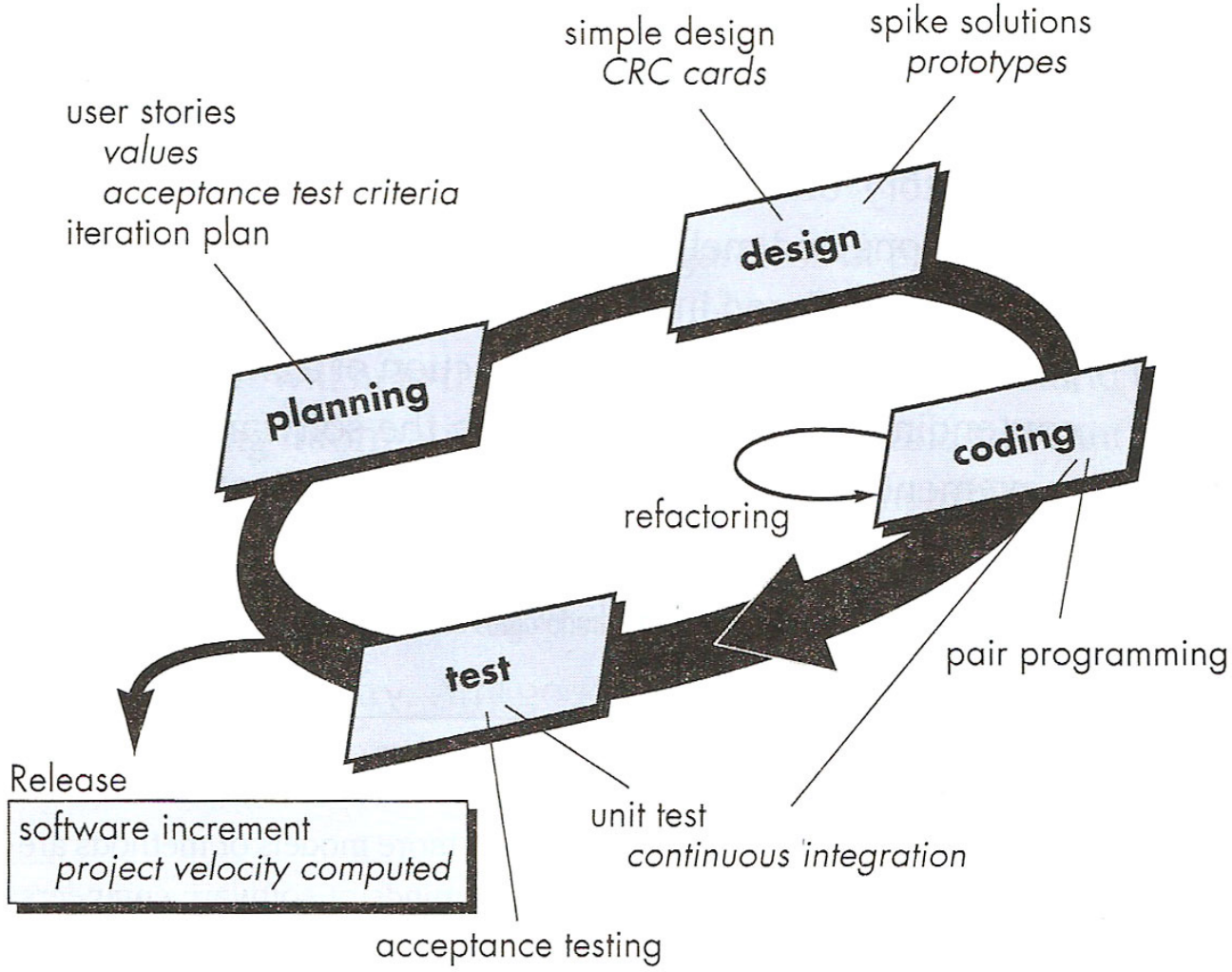
Coding

- Coding activity is ***pair programming.***

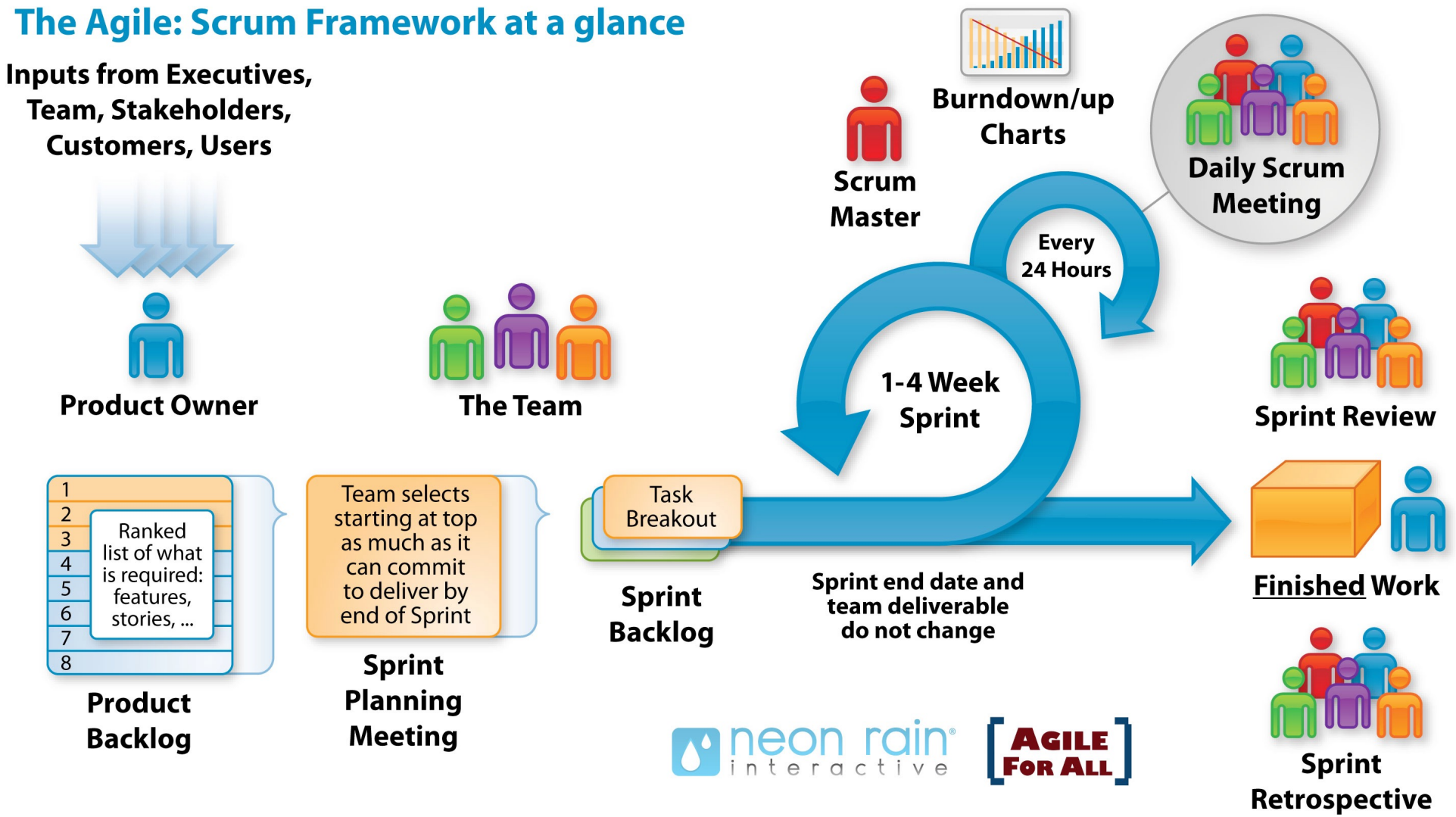
Test

- The unit tests are organized into a ***universal testing suite***, integration and validation testing of the system can occur on a daily basis.
- XP acceptance test (customer tests) are specified by the customer and focus on overall system features and functionality that are visible and reviewable by the customer.

Extreme Programming (XP) (Cont.) [1]



Srcum Model



[6]

Reference

1. Ian Sommerville, Software Engineering 10th Edition, Pearson, April 2015

Any Questions?

:O)

Thank you

Test your understanding

What is the oldest software process model?

Test your understanding (2)

What is the software process model for the software development situation below:

- In the ACM-computer programming competition, each group will have to solve 5 programming problems within 3 hours.
  - Answer:_____
- If you have to do the software project “Online meeting schedule planning” with the features that multiusers can collaborate to schedule several courses.
  - Answer:_____
- Mobile application for the social network communication i.e. Line
  - Answer:_____
- Basic unit converter application
  - Answer:_____