CP317 Software Engineering

week 9-2: Iterative models – part 2

Shaun Gao, Ph.D., P.Eng.

Agenda

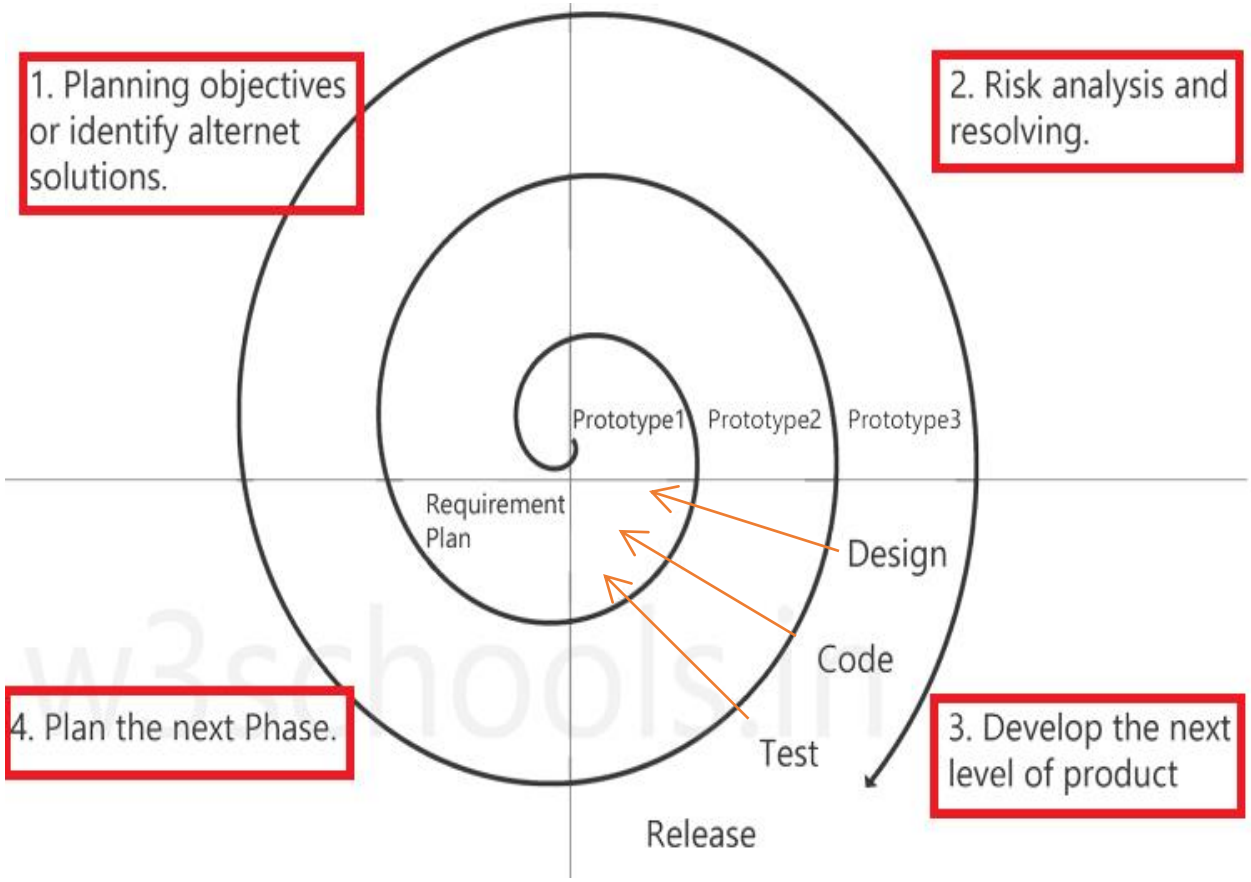
- Review part 1
- Spiral model
  - Concept
  - Four phases
  - Advantages and disadvantages
- Spiral model vs. waterfall model
- Spiral model vs. prototype
- Unified process model
  - Concept, four phases
  - Advantages and disadvantages
- Rational unified process
- Summary

Review part 1

- Iterative model
  - Concept
  - Advantages and disadvantages
- comparisons
  - Iterative model vs. waterfall model
  - Iterative model vs. incremental model
- Prototype model
  - Prototype model vs. iterative model

Spiral model

- A spiral model is a risk-driven process model of software engineering to help project teams decide on what development approach to take for various parts of a project.
  - Spiral model is similar to iterative model with more emphasize on risk analysis
- Four phases:
  - Planning phase
  - Risk analysis phase
  - Engineering phase
  - Evaluation phase



Spiral model – cont.

- Phases of the spiral model
  - Planning phase
    - Determine the objectives of the current cycle. For example, resources, timelines
  - Risk analysis phase
    - Perform a risk analysis to determine what the biggest risk factors are. For example, technical risks, management risks
  - Engineering phase
    - Design and development, developing planed features
  - Evaluation phase
    - Make sure that stakeholders agree that your solution is correct.

Spiral model – cont.

- Clarifications from the inventor (Barry Boehm in 1988) of Spiral model
  - Spiral model is not simply a series of waterfall models
  - The activities need not follow a single spiral sequence
  - Flexibility, add items or remove items from a spiral model
- Six characteristics
  - Define tasks concurrently
  - Perform the four tasks (goals, alternative approaches, resolve risks, stakeholders agree) in each cycle
  - Use risk to determine the level of effort
  - Use risk to determine the level of detail
  - Use anchor milestones
  - Focus on the system and its life cycle

Spiral model – cont.

- Advantages
  - Highly flexible model - spiral structure gives many points for review
  - Emphasizing risk analysis - resolve risks, lead to success
  - Accommodating changes (requirements & design) well
  - Time and effort estimation more accurate
- Disadvantages
  - It is complicated
  - Expensive to implement (Risk analysis can be difficult)
  - Stakeholders must have the skills needed to review the project periodically
  - Project success may depend greatly on the risk analysis

Spiral model vs. waterfall model

Spiral Model	Waterfall model
❑ Risk factors are considered.	❑ Risk factors are not considered.
❑ The requirements are not freeze.	❑ The requirements are freeze.
❑ works in loop.	❑ Is linear sequential model.
❑ costly as Risk factor is covered.	❑ Not much costly.
❑ Better communication between developer and customer.	❑ Communication level is not high

Spiral model vs. prototype model

Spiral Model

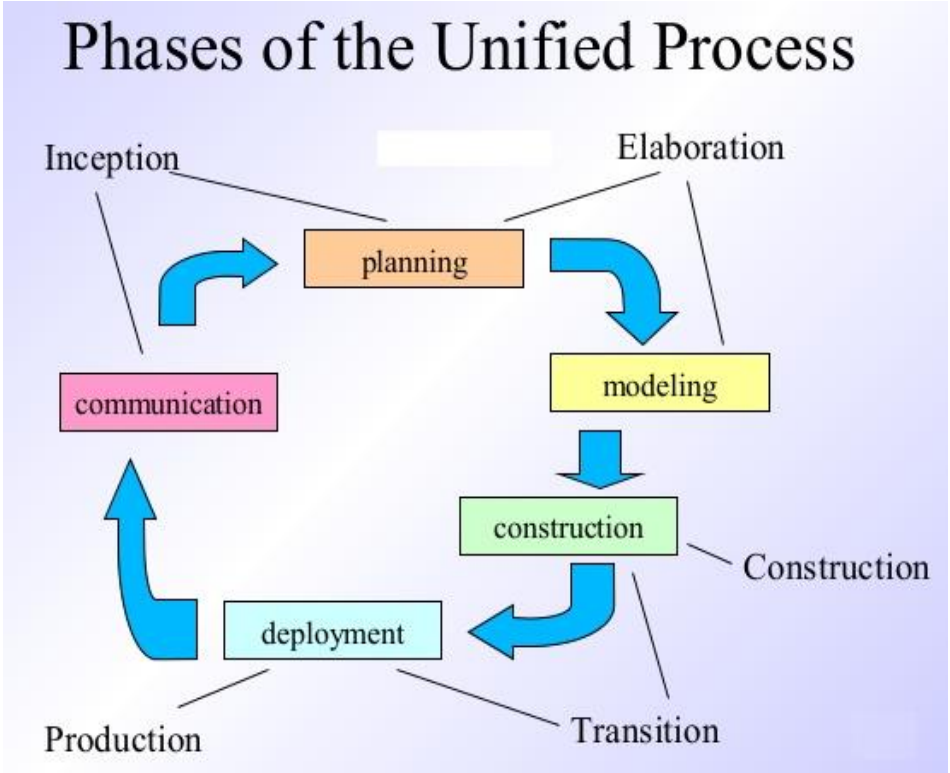
- Used when requirement is not clear and needs confirmation
- customer interaction continuous to move together.
- Risk factor is considered

PROTOTYPE MODEL

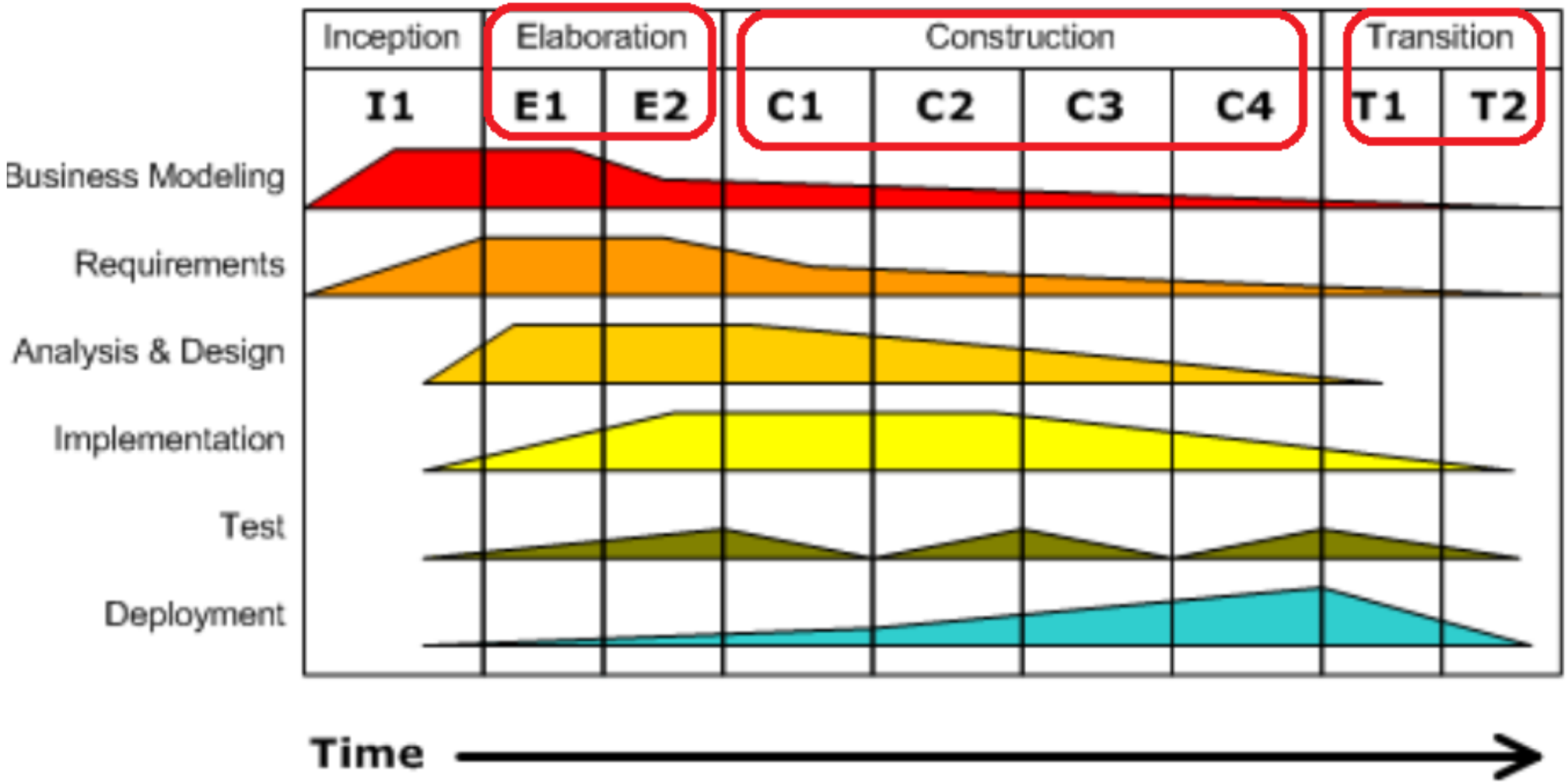
- Requirement are clear but complex.
- Customer interaction needs till the prototype is app.
- Risk factor are not considered

Unified process model

- A unified process model is a process model of software engineering that is a process framework of iterative model and incremental model that can be customized to fit business and project needs.
- Four phases
  - Inception: create project ideas
  - Elaboration: generate project requirements
  - Construction: perform coding, testing
  - Transition: provide product and maintenance
- Use-case driven approach



Unified process model

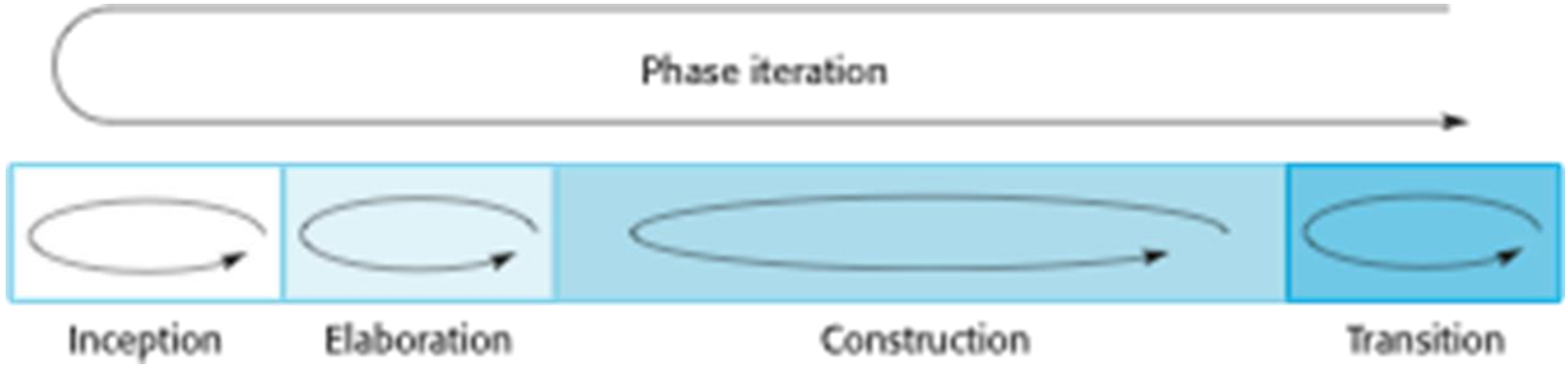


Unified process model

- Pros
  - The iterative approach enables a team to incrementally define requirements
  - The elaboration iterations focus on risks and risk mitigation to increase the project success chance
  - Flexibility, accommodate different models
- Cons
  - It is complicated
  - Require more resources because it is complicated
  - Risk analysis can be difficult

Rational Unified Process (RUP) model

- Rational Unified Process an iterative software development process framework created by Rational Software Corporation, a division of IBM since 2002. – a real product
- It has an underlying object-oriented model, using Unified Modeling Language (UML)
- Four phases and their iteration



RUP model – cont.

- Inception
  - Establish the business case for the system.
- Elaboration
  - Develop an understanding of the problem domain and the system architecture.
- Construction
  - System design, programming and testing.
- Transition
  - Deploy the system in its operating environment.

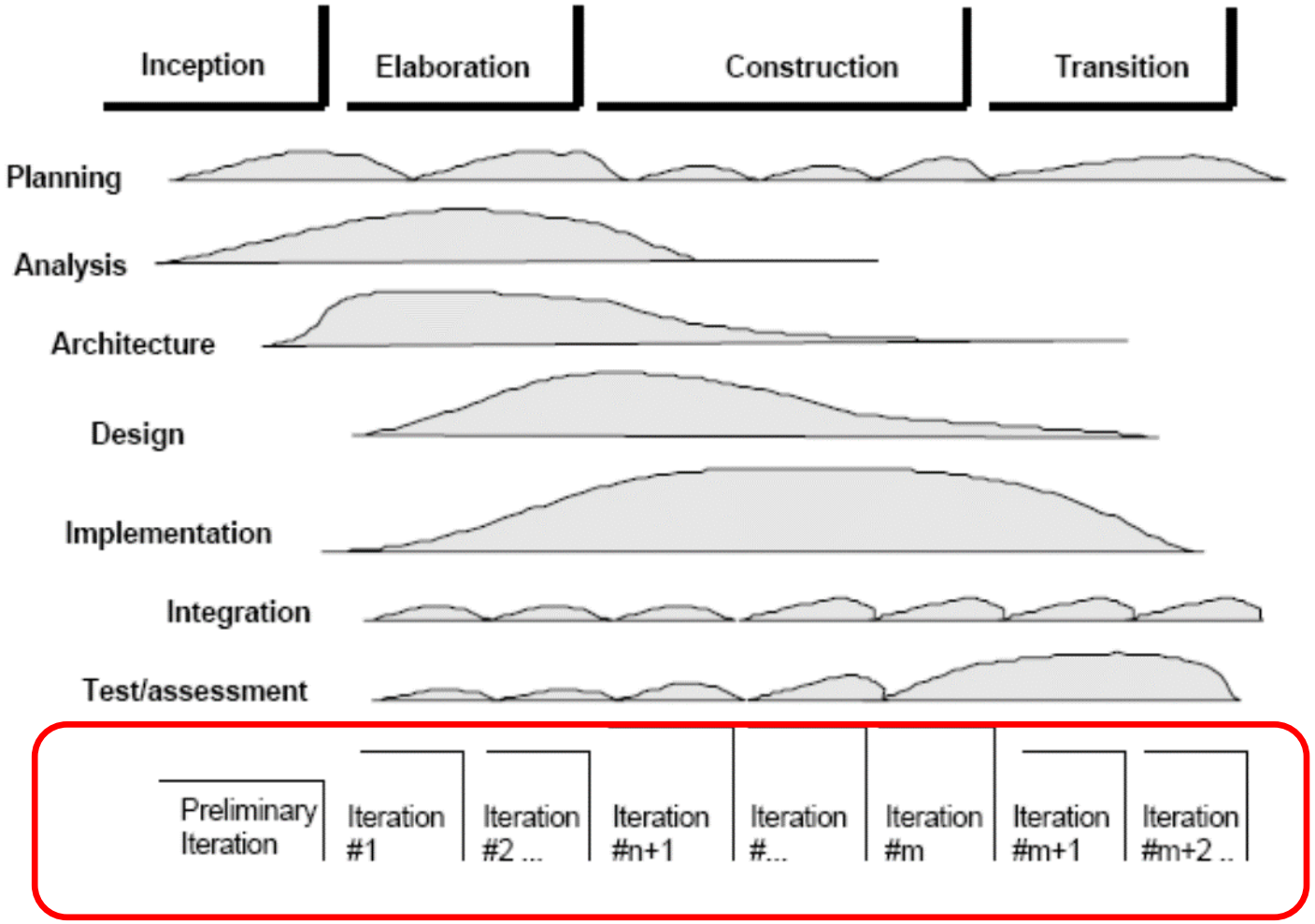
Static workflows in RUP model

- Business modelling workflow:
  - The business processes are modelled using business use cases.
- Requirements workflow:
  - Actors who interact with the system are identified and use cases are developed to model the system requirements. – **use case diagram**
- Analysis and design workflow:
  - A design model is created and documented using architectural models, component models, object models and sequence models.
- Implementation workflow:
  - The components in the system are implemented and structured into implementation sub-systems. **Automatic code generation** from design models helps accelerate this process.

Static workflows in RUP model – cont.

- Testing workflow:
  - Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
- Deployment workflow:
  - A product release is created, distributed to users and installed in their workplace.
- Project management workflow:
  - This supporting workflow manages the system development.
- Configuration and change management workflow:
  - This supporting workflow managed changes to the system.

RUP model overview



Rational Unified Process – cont.

Advantages/Disadvantages of RUP

- Advantages
  - Iterative
  - Use Case Driven
  - Architecture Centric
  - Fully Utilizes UML
- Disadvantages
  - Lack of Structure

Rapid Application Development (RAD)

- Software engineering differs from traditional engineering
  - Design or blueprint is rigidity, for example, buildings or bridges
  - Software development changes the design easily compare to others
- The problems
  - Problem1: Software took so long to build that requirements had changed before the system was complete, resulting in inadequate or even unusable systems.
  - Problem2: the assumption that a methodical requirements analysis phase alone would identify all the critical requirements

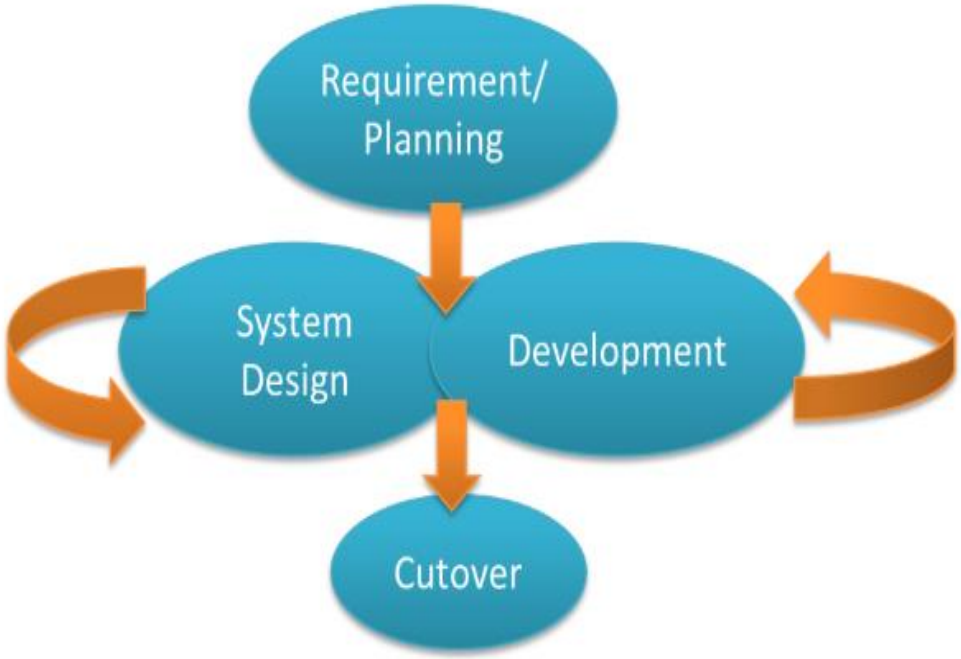
Introduction to RAD – cont.

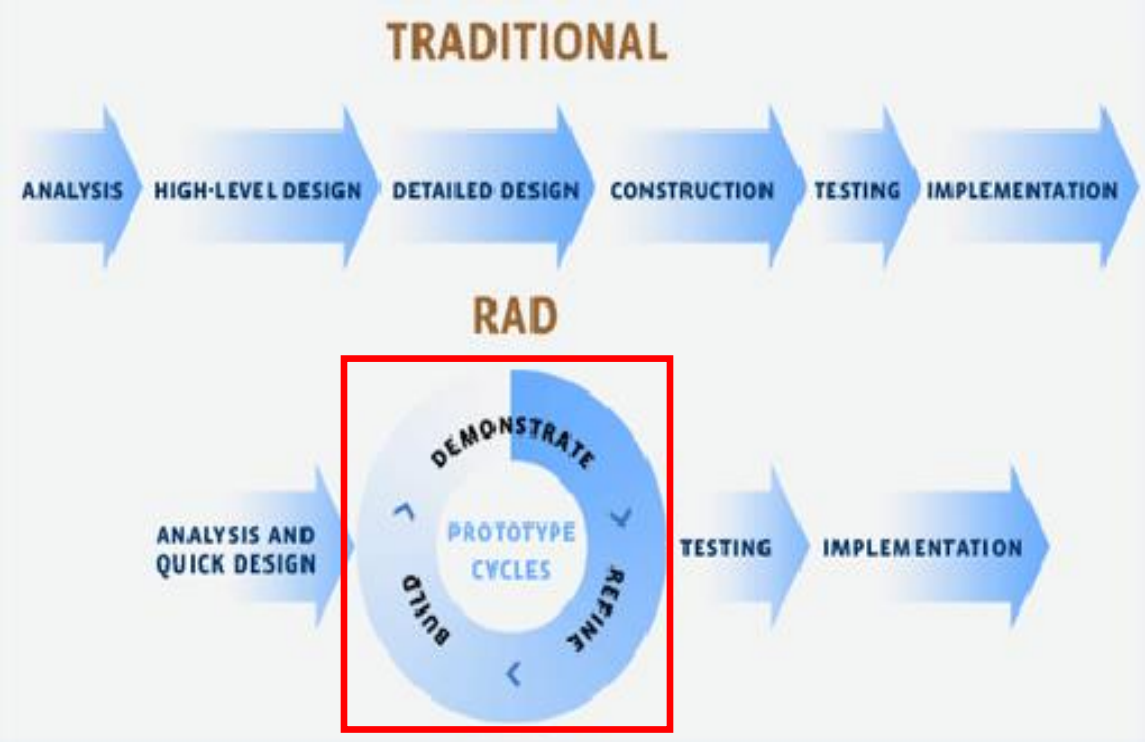
- Software becomes common
  - Tools for software developments
  - software frameworks, web application frameworks
- Based on the above points
  - Brian Gallagher, Alex Balchin, Barry Boehm and Scott Shultz, James Martin developed the rapid application development (RAD) approach in 1991

Rapid application development

- RAD is a software development methodology that enables organizations to develop strategically important software systems faster while reducing development costs and maintaining quality. This is achieved by using a series of proven application development techniques and computer-assisted software engineering (CASE) tools.
- Key elements: (1) proven technologies (2) CASE
- The key objectives of RAD
  - High speed
  - High quality
  - Low cost

RAD phases


Requirement/
Planning
System
Design
Development
Cutover


TRADITIONAL
ANALYSIS
HIGH-LEVEL DESIGN
DETAILED DESIGN
CONSTRUCTION
TESTING
IMPLEMENTATION
RAD
ANALYSIS AND
QUICK DESIGN
DEMONSTRATE
REFINE
BUILD
PROTOTYPE
CYCLES
TESTING
IMPLEMENTATION

Summary

- Spiral model
  - Concept
  - Four phases
  - Advantages and disadvantages
- Spiral model vs. waterfall model
- Spiral model vs. prototype
- Unified process model
  - Concept, four phases
  - Advantages and disadvantages
- Rational unified process
- RAD introduction

Announcement

- **Group project presentation schedule:**
  - Nov. 21 (Thursday) group numbers 9, 13, 18, 2
  - Nov. 26 (Tuesday) group numbers 6, 11, 1, 21, 22
  - Nov. 28 (Thursday) group numbers 10, 8, 5, 20, 7
  - Dec. 3 (Tuesday) group numbers 12, 15, 24, 19, 3
  - Dec. 5 (Thursday) group numbers 16, 23, 14, 17, 4
- Group project status
  - Software design document – project report due date Nov. 22