Rational Unified Process

Computing Science 213

Jason Madar

Object Oriented Analysis and Design

- What is OOA and OOD?
- What is UML?
- What is Rational Unified Process
- Stages of RUP
- Static Structure of RUP
- RUP Workflow Example
- Activities in Rational Unified Process
- Effort distribution of activities
- Relative effort of activities

What is OOA and OOD?

- Analysis emphasizes an investigation of the problem and not on how solution is defined
- Design emphasizes a logical solution, how the system fulfills the requirements
- Object oriented analysis and design is to emphasize considering problem domain and logical solution from perspective of objects

What is OOA and OOD? (cont.)

- Object oriented analysis finds and describes the objects or concepts in problem domain
- Object oriented design defines logical software objects that will implement in an object-oriented programming language

What is UML

- UML stands for Unified Modeling Language
- UML is a language for specifying, visualizing and constructing the artifacts of software system
- UML is a modeling language, not a method
- Methods consist of both a modeling language and a process

What is UML (cont.)

- A process describes who is doing what, how, and when
- Modeling language is the notation that methods use to express designs
- It is a notational system with semantics defined aimed at modeling systems using object-oriented concepts

What is UML (cont.)

- It is an industry standard for object oriented modeling approved by OMG (Object Management Group)
- The UML is the emerging effort of:
  - Grady Booch (creator of Booch modeling technique)
  - Jim Rumbaugh (creator of OMT, Object Modeling technique)
  - Ivar Jacobson (creator of OOSE, Object-Oriented Software Engineering)

What is UML? (cont.)

- However, UML does not include description of a process
- Grady Booch, Jim Rumbaugh, Ivar Jacobson from Rational also form a software engineering process called Rational Unified Process
- The UML is used throughout the Rational Unified Process

What is Rational Unified Process

- An **iterative and incremental** approach allows an increasing understanding of the problem through successive refinements
- An **architecture-centric** approach
- A **use-case driven** approach
- Manages **risk**
- Manages **change**
- Can be tailored to different situations (flexible)

What is RUP (cont.)

- RUP incrementally grows an effective solution over multiple iterations
- RUP consists of 4 phases
  - Inception
  - Elaboration
  - Construction
  - Transition

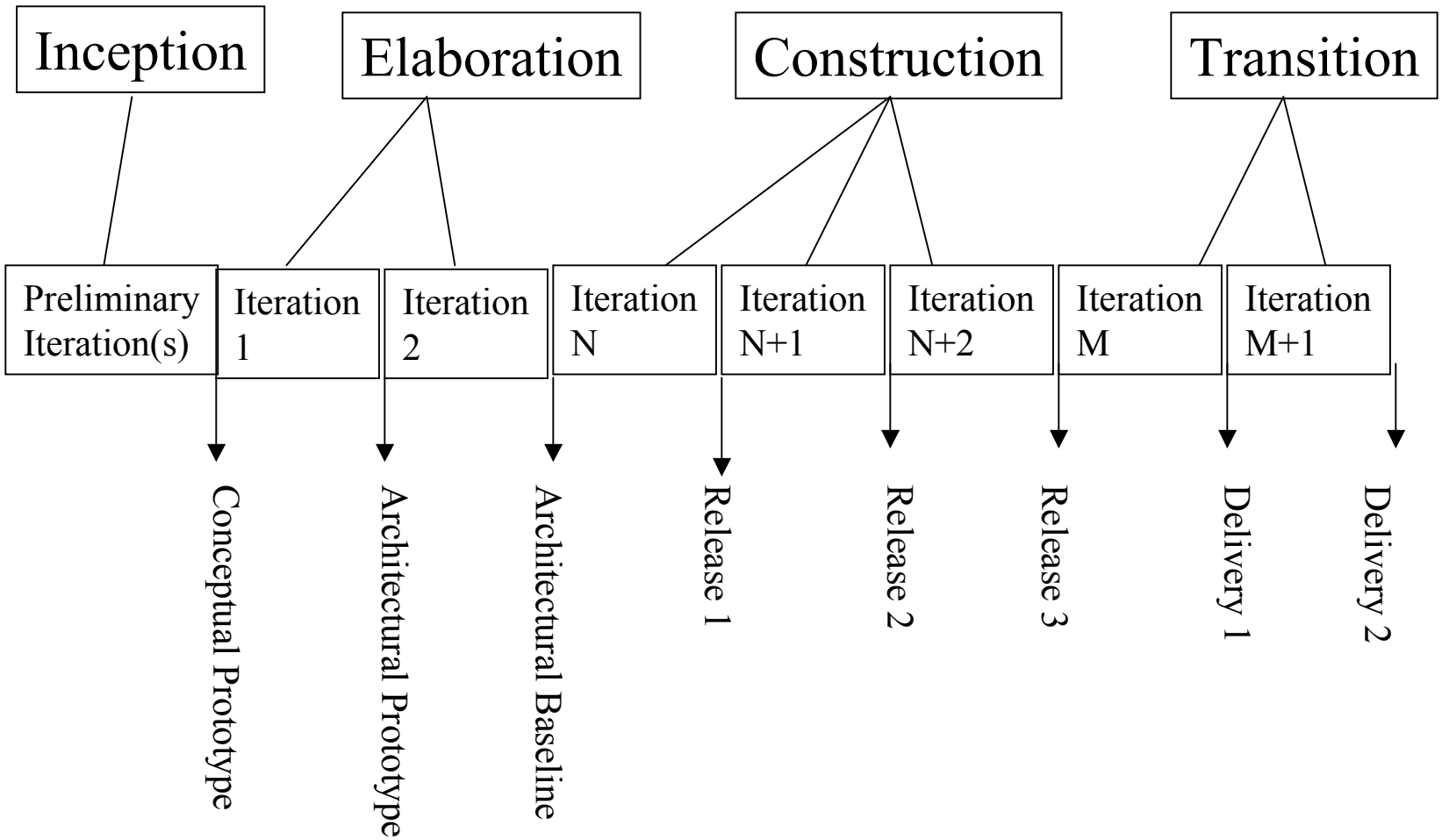
What is RUP (cont.)

- Inception phase establishes the business rationale for the project, delimits the project scope and a conceptual prototype
- Elaboration phase collects more detailed requirements, performs high-level analysis and design to establish an architecture baseline, and create a plan for construction

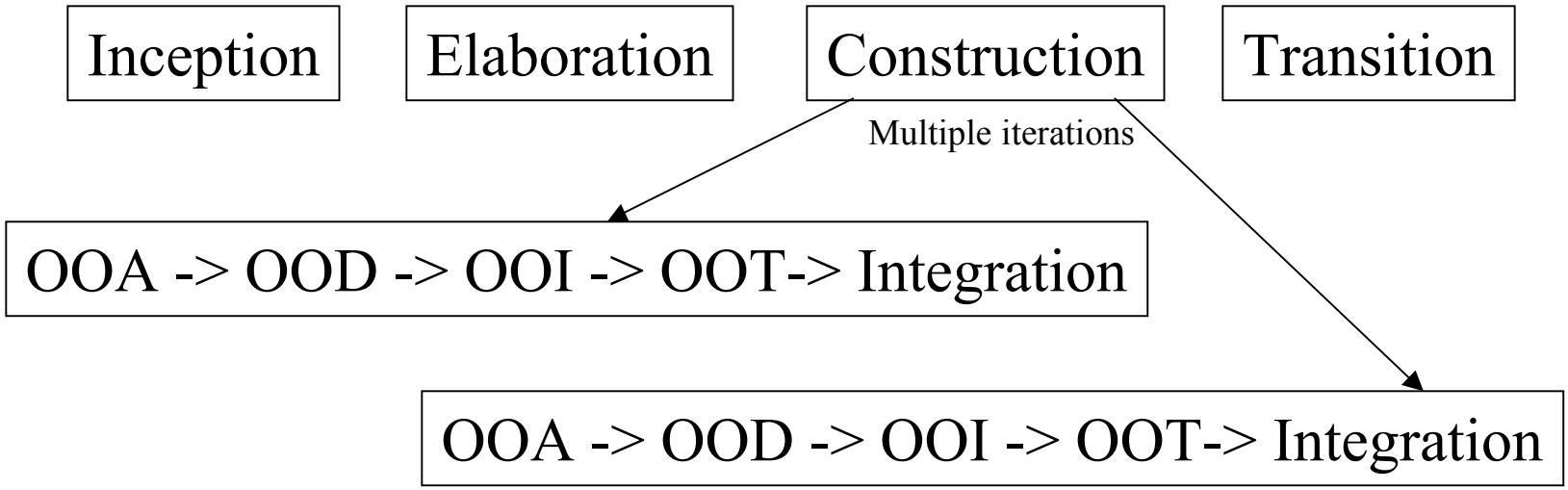
What is RUP (cont.)

- Construction phase consists of many iterations, in which each iteration analyzes, designs, builds, tests, and integrates a subset of requirements of a project
- Transition phase includes beta testing, packaging, performance tuning, and user training

What is RUP (cont.)



What is RUP (cont.)



Inception

- Investigation on the problem to be solved and analysis of the benefits of using such a system
- Work out business cases for a project
- Cost of the project
- Revenue the project will bring (ROI)
- Size of the project
- Time spent in this phase should be few days for a few months project

Elaboration

- At this stage, requirements for the project is usually quite vague
- To better understand requirement, ask the following questions:
  - What is actually going to be built?
  - How are you going to built it?
  - What technology are you going to use?

Elaboration (cont.)

- The issues addressing in this stage can be found by identifying the risks in your project:
  - Requirement risk: What is the chance of building the wrong system
  - Technological risk: Assume the use of OO and Java technology, how much is known about OO design? Will Java do the job?

Elaboration (cont.)

- Skill risk: Staff has the necessary expertise?
- Political risk: Any political influences that may affect the project?

Elaboration (cont.)

- To address requirement risk:
  - Use cases drive the elaboration phase and form the foundation of planning for the construction phase
  - A use case is a typical sequence that a user has with the system in order to achieve some goal
  - A skeleton of conceptual model of the problem domain is provided

Elaboration (cont.)

- Explore the vocabulary of the domain
- UML class diagrams capture the conceptual perspective of the business requirement
- UML sequence diagrams explore how various roles interact in the business
- One may build a prototype of any tricky parts of the use cases

Elaboration (cont.)

- To address the technological risk:
  - Try out pieces of potential technology
  - Integrate test of the pieces of technology
  - Architectural design decisions can be illustrated by the following UML diagrams:
    - Package diagram: A high level picture of the components at this stage
    - Deployment diagram: An overview of how pieces are distributed from system architecture perspectives
    - Sequence diagram: How components are communicated

Elaboration (cont.)

- To address the skill risk:
  - Train staff before the project started
  - Mentoring: Have an experience developer work with your project or have him/her review your project from stage to stage

Elaboration (cont.)

- Result of the elaboration is to have a baseline architecture for the system, including the following:
  - A list of use cases
  - Conceptual model
  - Technology platform

Elaboration (cont.)

- Plan for the construction phase includes
  - Prioritization of use cases
  - Assignment of use cases into iterations of the construction phase
- A good rule of thumb is that elaboration should take about a fifth of the total length of the project

Construction

- Builds system in a series of iterations
- Each iteration which is based on one or subset of a use case, including
  - Detail analysis
  - Detail design
  - Coding
  - Testing
  - Integration of the use cases from previous iterations

Construction (cont.)

- Iteration is incremental in function. Each iteration builds on the use cases developed in the previous iteration
- Iteration is iterative in terms of code base and rewrite some existing code to make it more flexible
- Class diagram roughs out concepts for the use case and see how to fit into the software from previous iterations

Construction (cont.)

- If the use case contains significant workflow element, use sequence diagram to help
- If a class has complex dynamic behaviour (many state changes in response of events), use state diagram
- Use package diagram to help visualize the logical pieces of the system

Construction (cont.)

- Use patterns if possible to address common problem
- Patterns are well known model developed and collected by those with experiences to a set of common problems

Transition

- Optimization of code to improve system performance can be addressed in this stage
- Bug fixing
- No additional functionalities
- This is the time between beta release to customer and final release of a product
- User training for beta users
- Packaging of software

Static Structure of RUP

- A process describes **who** (the worker) is doing **what** (artifacts), **how** (activities), and **when** (workflows)
- RUP has 4 primary modeling elements:
  - A **worker** defines the behaviour and responsibilities of an individual, or group of individuals working together in a team

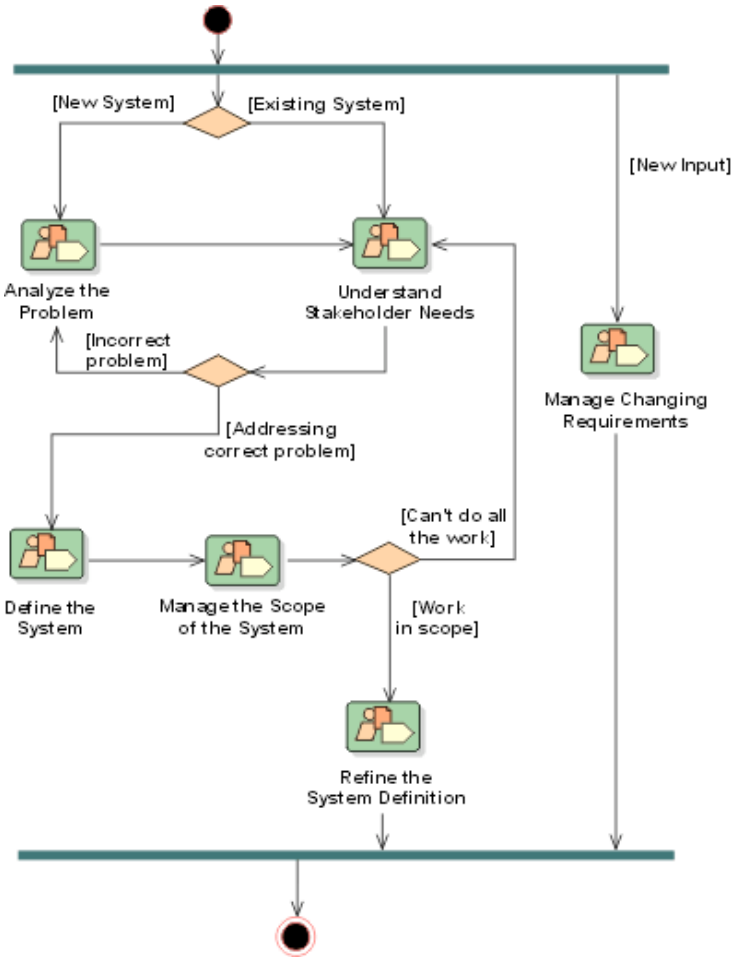
Static Structure of RUP (cont.)

- An **activity** of a specific worker is a unit of work that an individual in that role may be asked to perform
- An **artifact** is a piece of information that is produced, modified, or used by a process
- A **workflow** is a sequence of activities that show interactions between activities

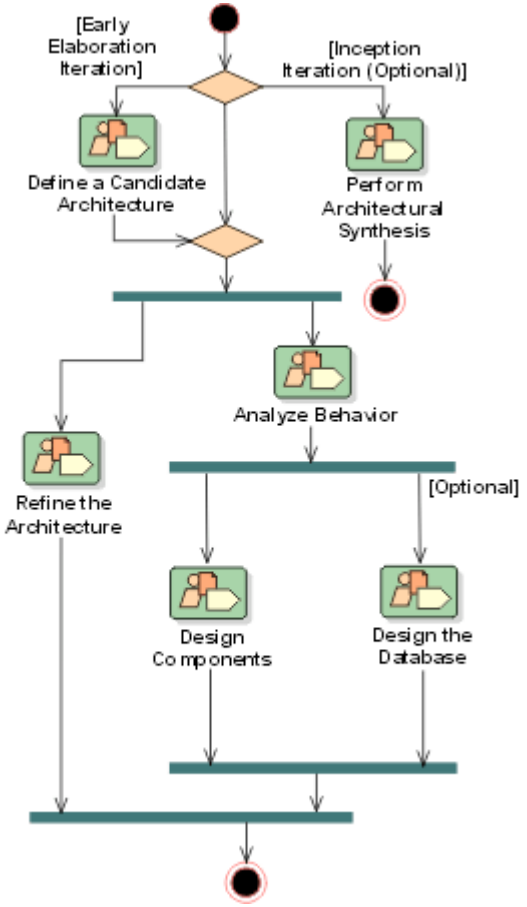
- RUP provides workflow samples for each worker

RUP Workflows

Requirements

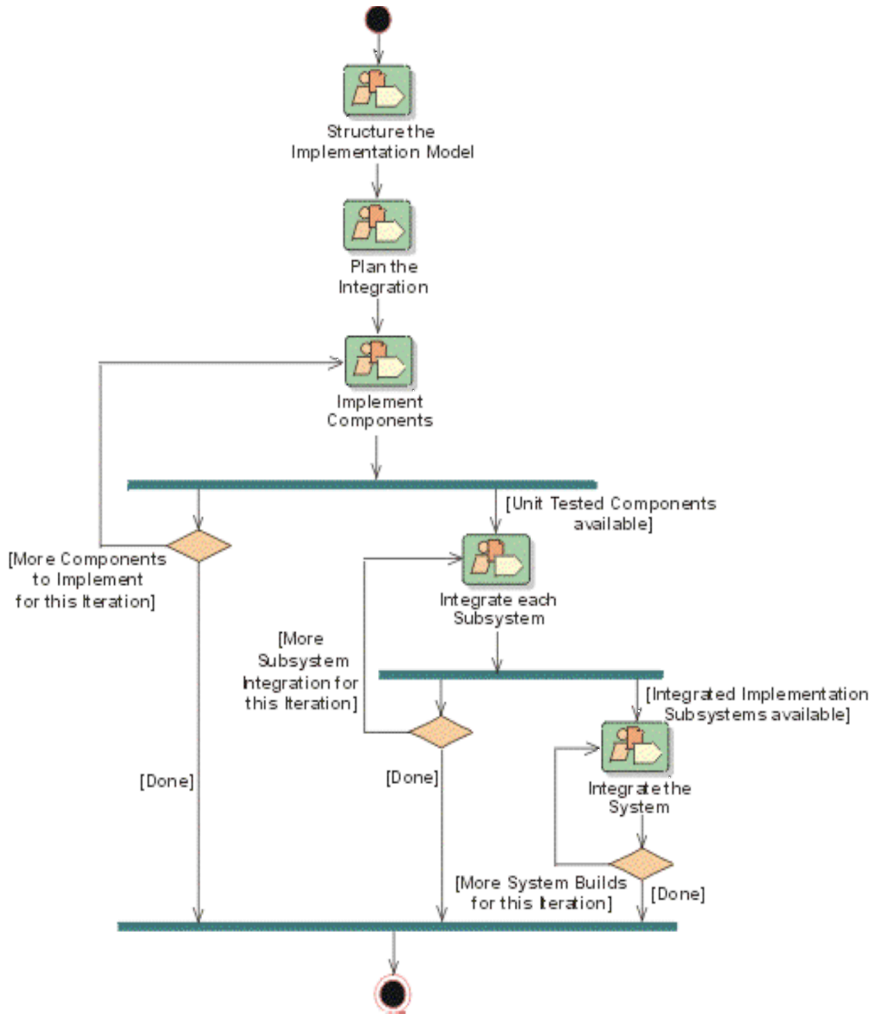


Analysis and Design

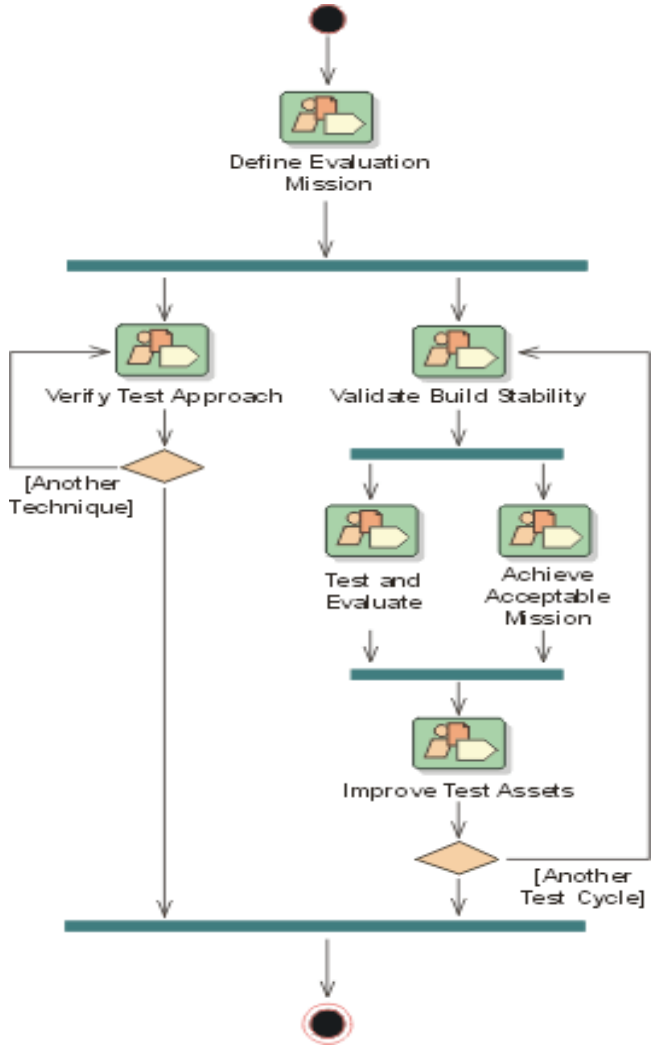


RUP Workflows

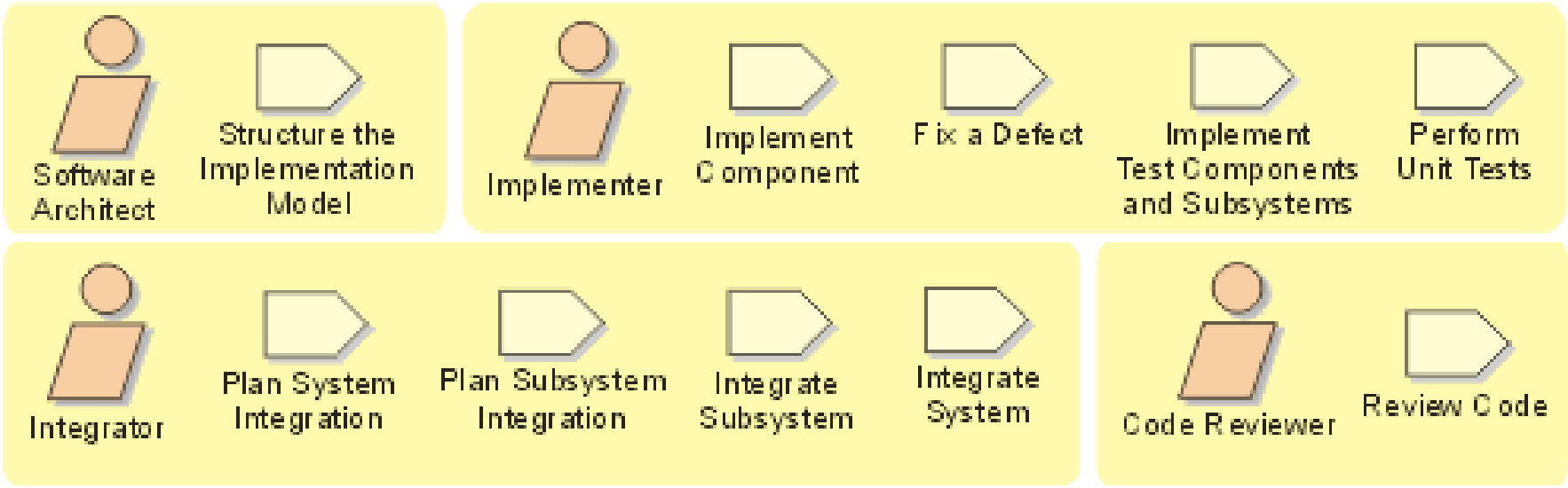
Implementation



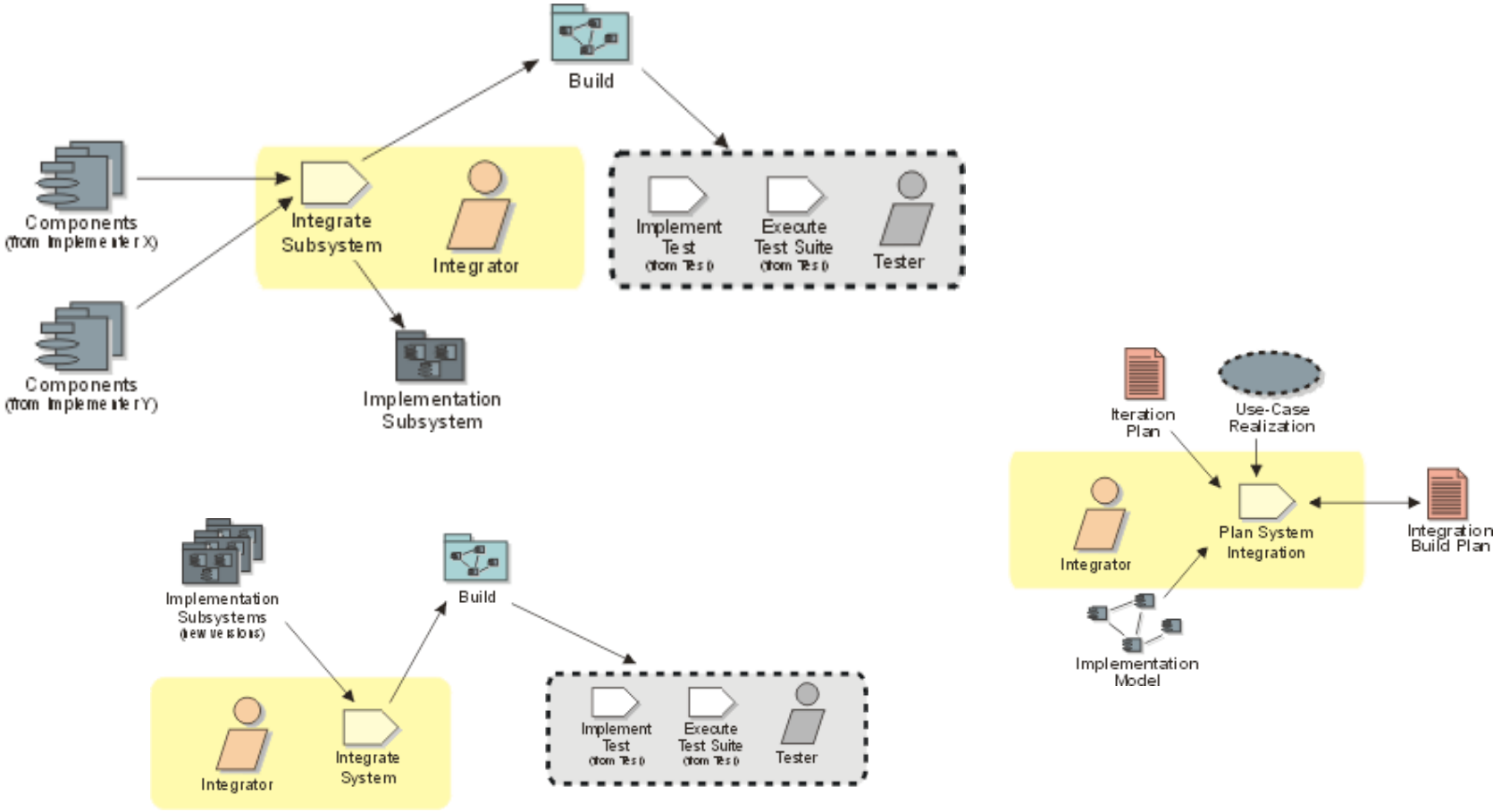
Testing



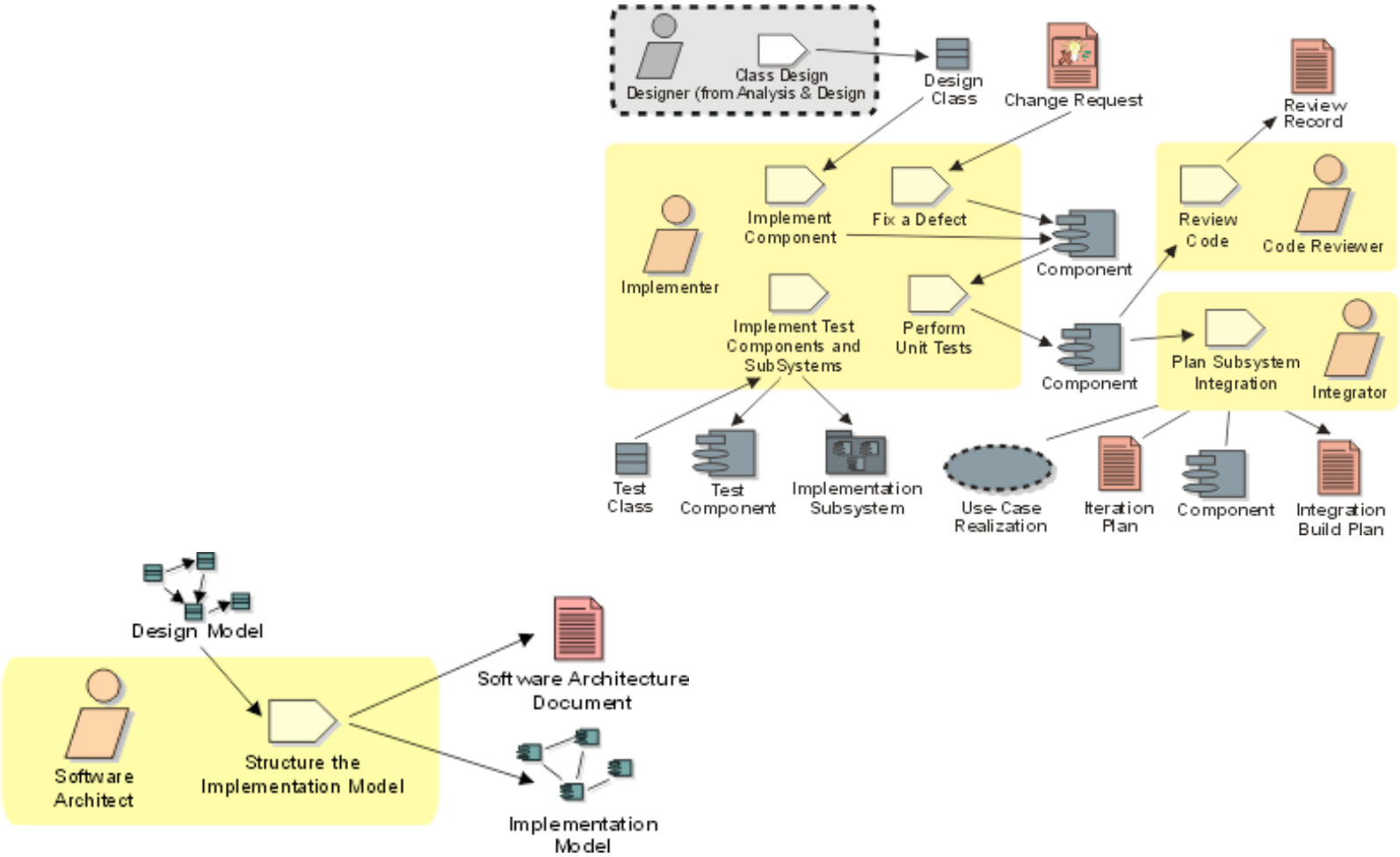
Activities vs Roles (implementation workflow)



Activities vs Roles vs Artifacts (implementation workflow)



Activities vs Roles vs Artifacts (implementation workflow)



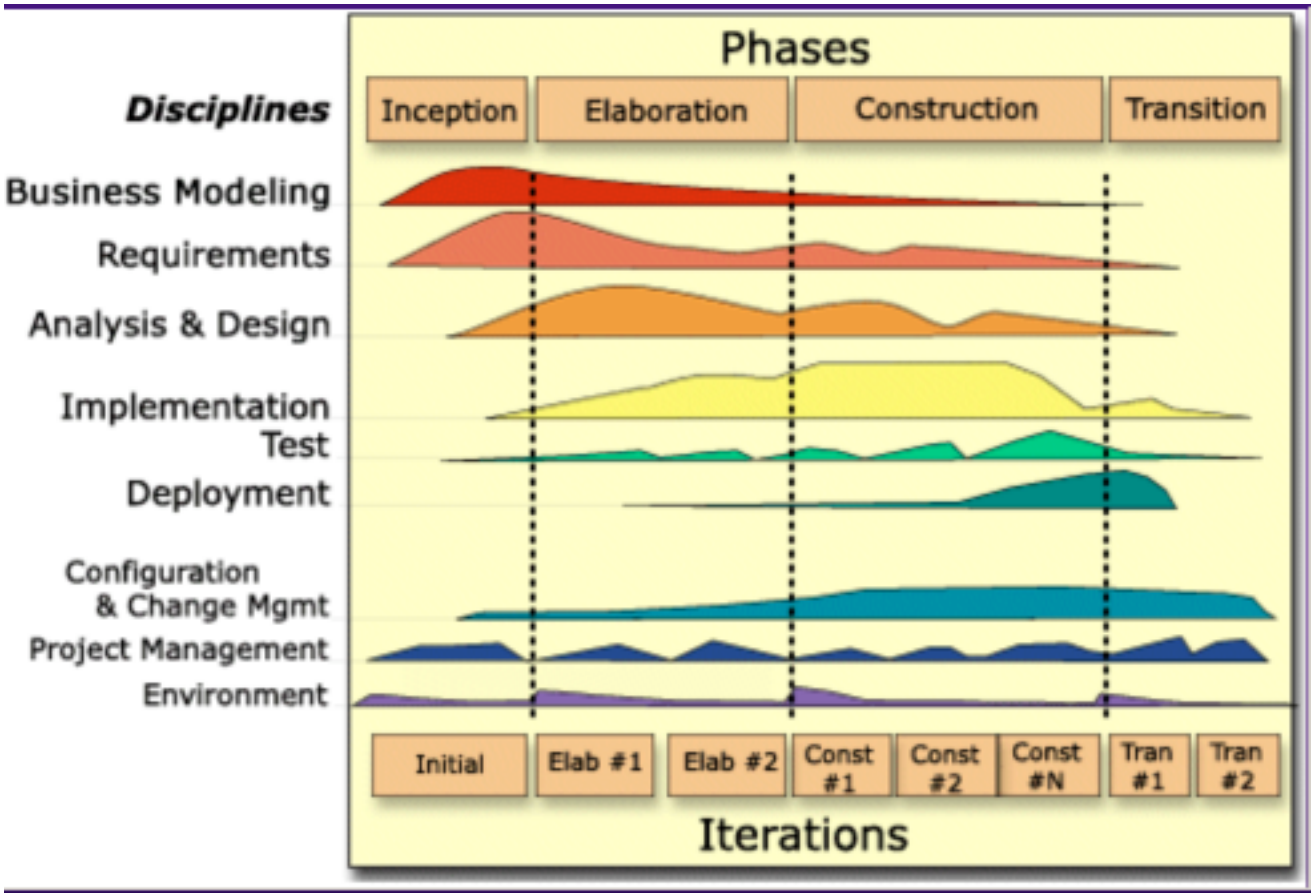
Activities in RUP

- Activities in RUP can be divided into the following major categories:
  - Planning
  - Analysis
  - Architecture/Change Management/Tools
  - Design
  - Implementation
  - Integration
  - Test/Assessment

Activities in RUP

- Activities can run in parallel
- The effort distribution of each activity depends on the locality of the RUP phase(s)
- A sample of relative effort % for each RUP activity for a medium size project is provided later

Effort Distribution of Activities



Relative Effort of Activities

- Planning 15%
- Analysis 10%
- Architectural/Management 10%
- Design/Integration 15%
- Implementation 30%
- Maintenance 5%
- Test/Assessment 15%