Software Quality Assurance

Module 5

The RUP Test Discipline

Objectives

♦ **Introduce concepts and vocabulary used in this course:**

 ▪ The terminology of RUP

 ▪ The testing discipline in RUP

 ▪ The testing workflow structure

文档仅限个人使用，请勿上传至互联网中，违者必究！

What is the Rational Unified Process (RUP)?

The Rational Unified Process (RUP) is a software engineering process framework that provides a disciplined yet flexible approach to assigning tasks and responsibilities within a software development organization.

RUP's goal is to support the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.

The RUP supports many software engineering practices

◆ The dynamic structure (phases and iterations) of the Rational Unified Process creates a basis for iterative development.

◆ The Project Management discipline describes how to set up and execute a project using phases and iterations.

◆ The Use-Case Model and Risk List of the Requirements discipline help determine what functionality you implement in each iteration.

◆ The Workflow Details of the Requirements discipline show the activities and artifacts that make requirements management possible.

◆ The iterative approach allows you to progressively identify components, decide which ones to develop, which ones to reuse, and which ones to buy.

◆ The Unified Modeling Language (UML) used in the process represents the basis of Visual Modeling and has become the de facto modeling language standard.

◆ The focus on software architecture allows you to articulate the structure: the components and the ways in which they integrate, the fundamental mechanisms and patterns by which they interact

Module 5 - Agenda
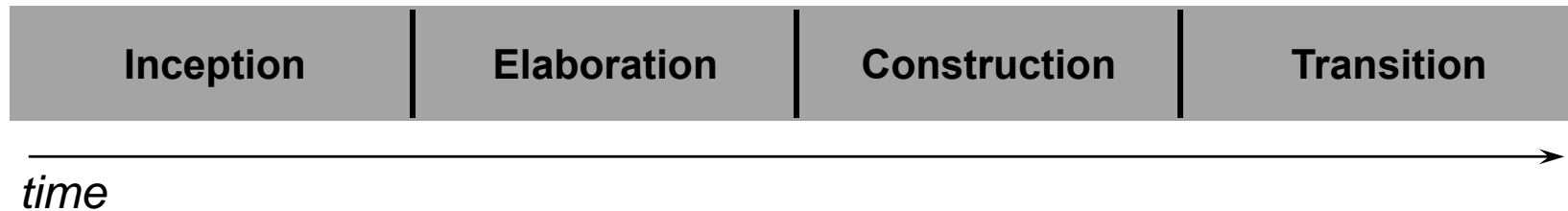
➔ **Overview of the software lifecycle in RUP**

◆ Overview of the building blocks of RUP

◆ Roles in the Test Discipline

◆ Workflow Details in the Test Discipline

RUP Process Architecture

Disciplines
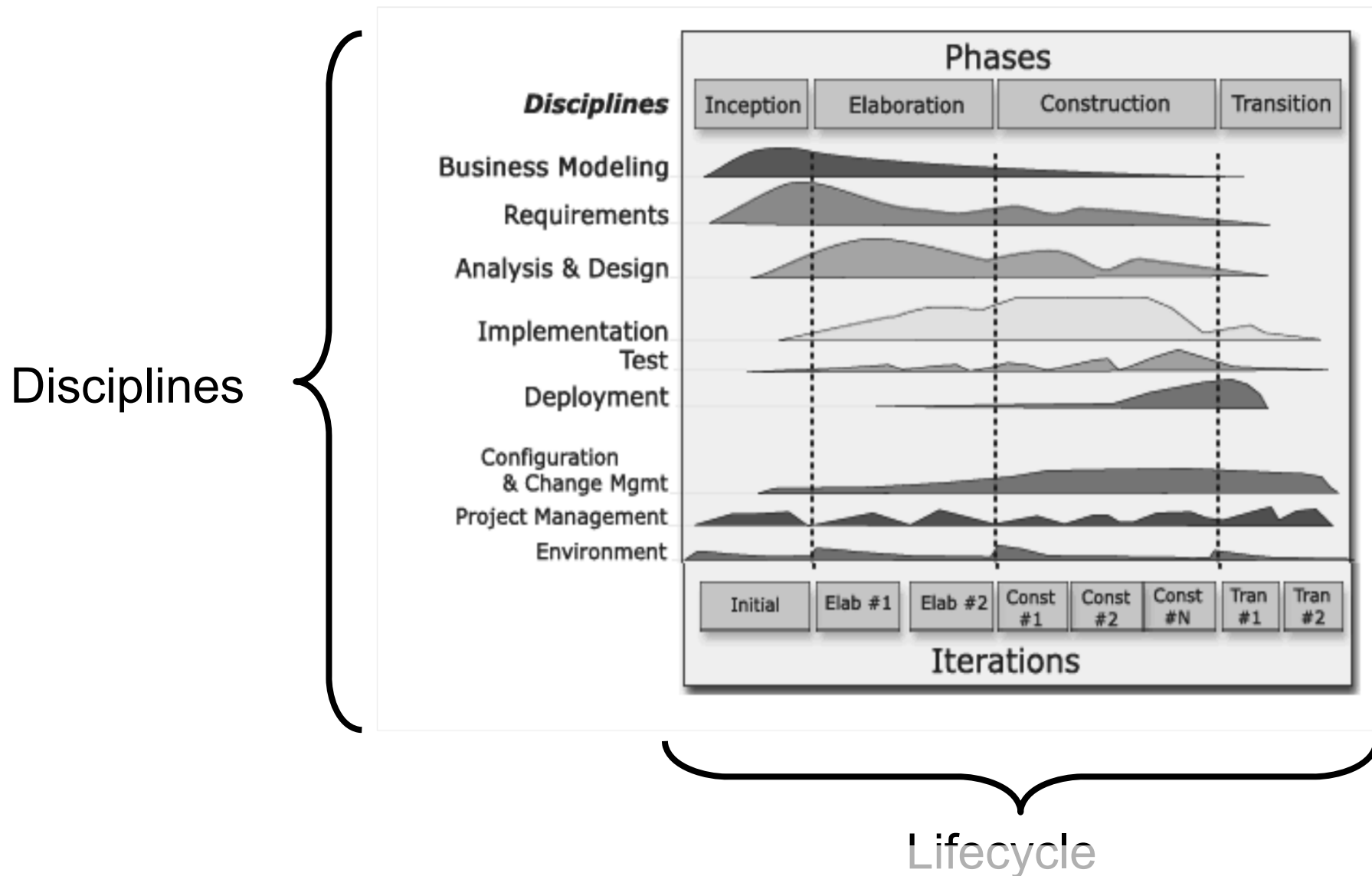
Lifecycle
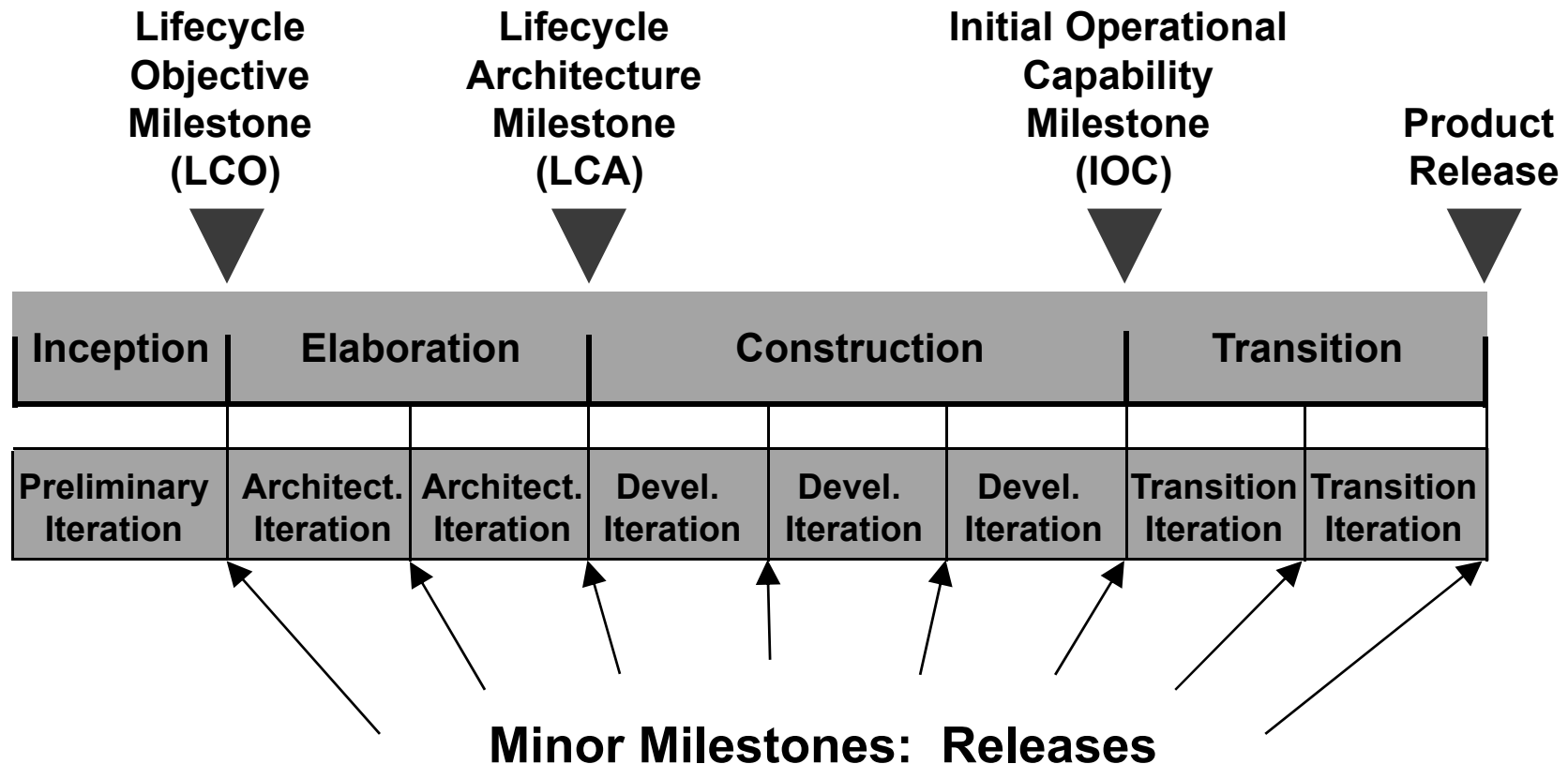
文档仅限个人使用，请勿上传至互联网中，违者必究！

Process Structure - Lifecycle Phases

Inception	Elaboration	Construction	Transition

time

## The Rational Unified Process has four phases:

- **Inception** - Define the project scope, gain agreement on project objectives, baseline the product Vision

- **Elaboration** - Address key technical risks, produce an evolutionary prototype, baseline the Architecture

- **Construction** - Iteratively and incrementally develop an operationally complete product

- **Transition** - Deliver the product into the live end-user environment

The Lifecycle Has Phases and Iterations

	Lifecycle Objective Milestone (LCO)		Lifecycle Architecture Milestone (LCA)			Initial Operational Capability Milestone (IOC)		Product Release
	▼		▼			▼		▼

Inception	Elaboration		Construction				Transition	
Preliminary Iteration	Architect. Iteration	Architect. Iteration	Devel. Iteration	Devel. Iteration	Devel. Iteration	Transition Iteration	Transition Iteration

Minor Milestones: Releases

Each iteration results in an executable release (internal or external). Iiterations are the "heartbeat" or rhythm of the project and a governing principle for testing in RUP.

# Bringing It All Together: The Iterative Approach

In an iteration, you typically address all disciplines

Disciplines group activities logically
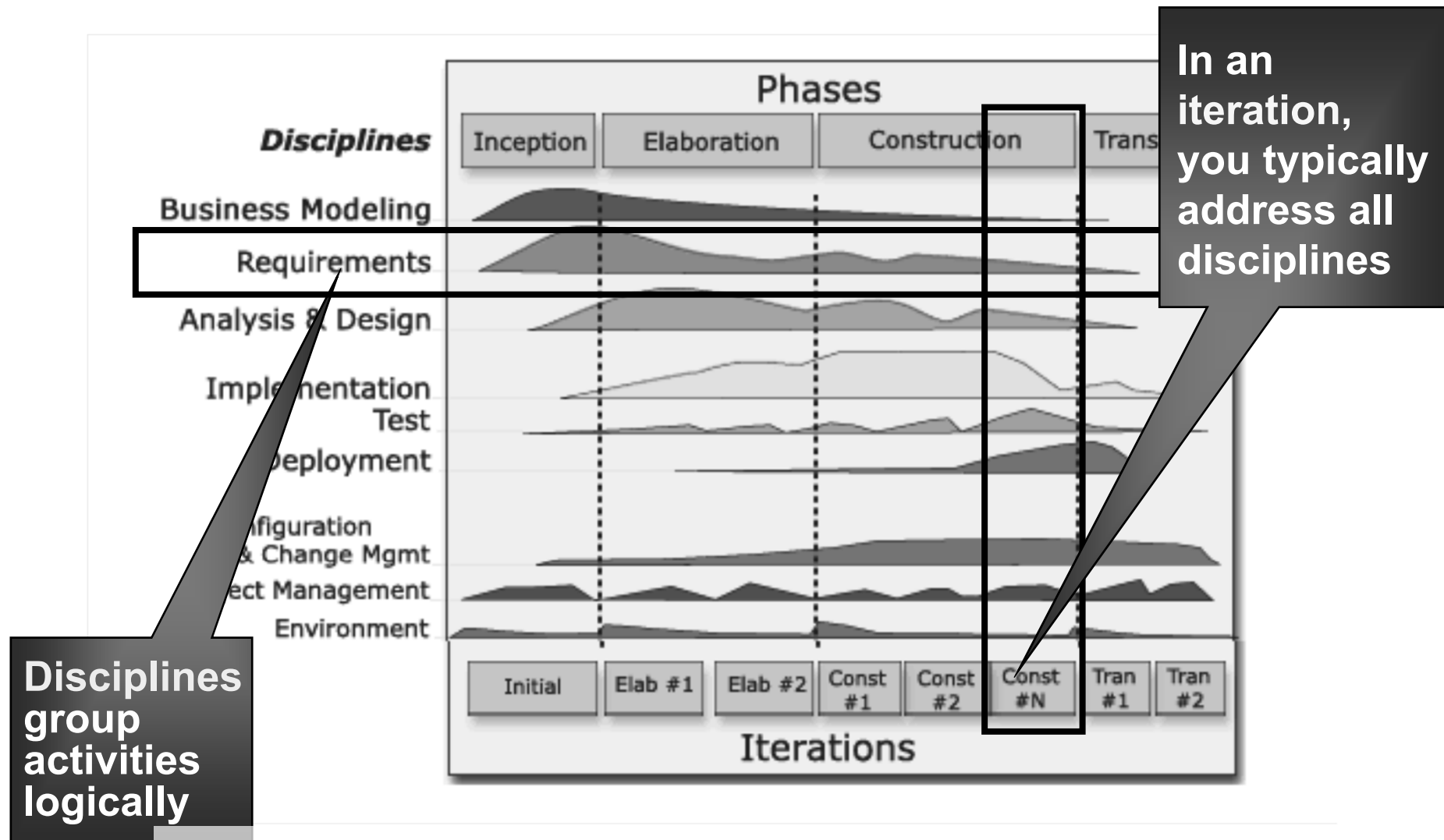
Business Modeling

Purpose

➢ To understand the structure and the dynamics of the organization in the target organization

➢ To understand current problems in the target organization and identify improvement potentials

➢ To ensure that customers, end users, and developers have a common understanding of the target organization

➢ To derive the system requirements needed to support the target organization
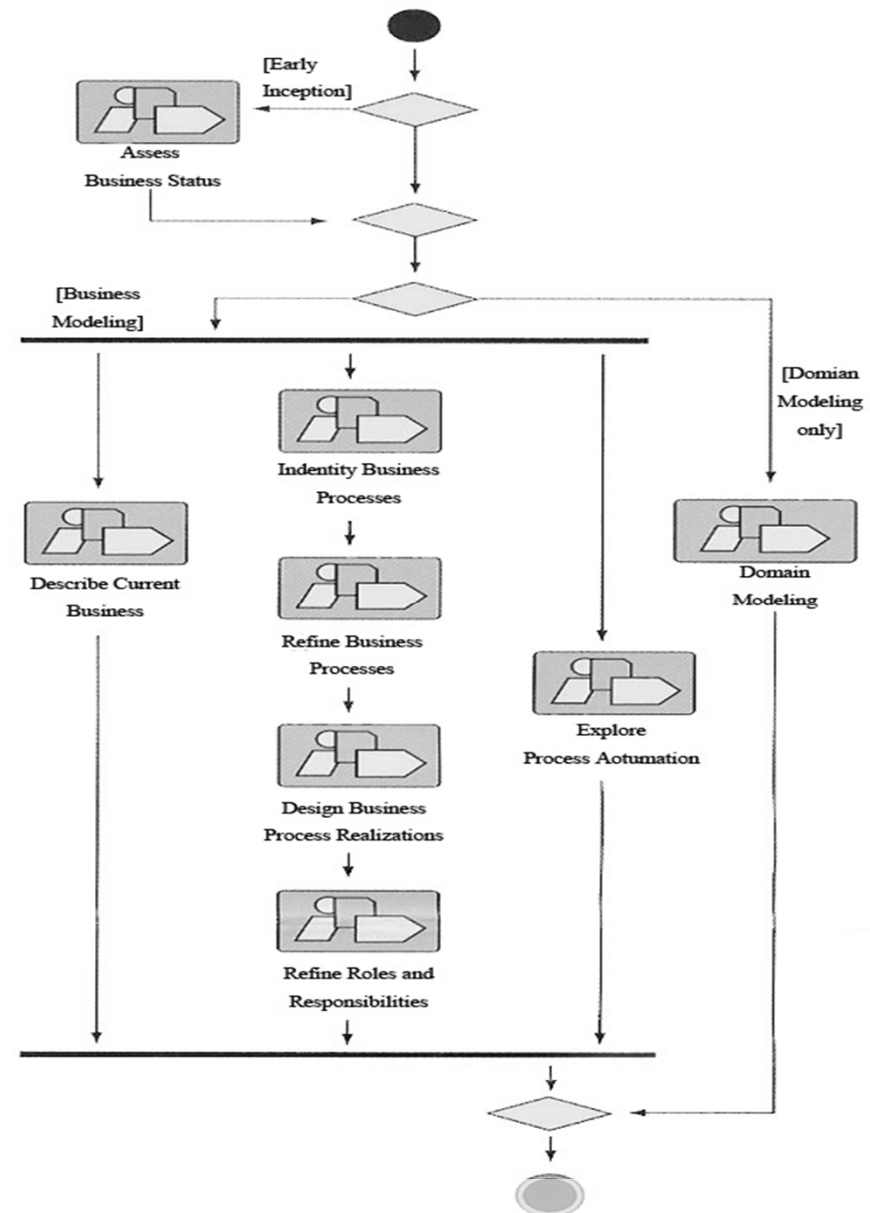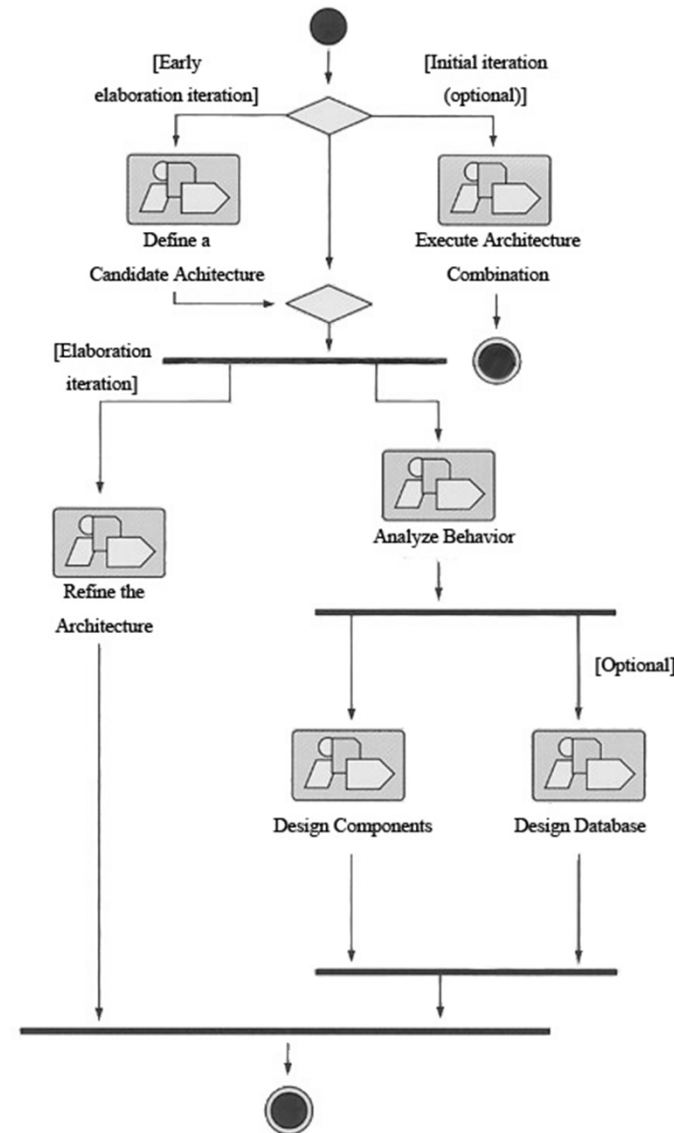
Requirements

Purpose

➢ To establish agreement with the customers and other stakeholders on what the system should do
➢ To provide system developers with a better understanding of the system requirements
➢ To define the boundaries of the system
➢ To provide a basis for estimating cost and time to develop the system
➢ To define a user-interface for the system, focusing on the needs and goals of the users
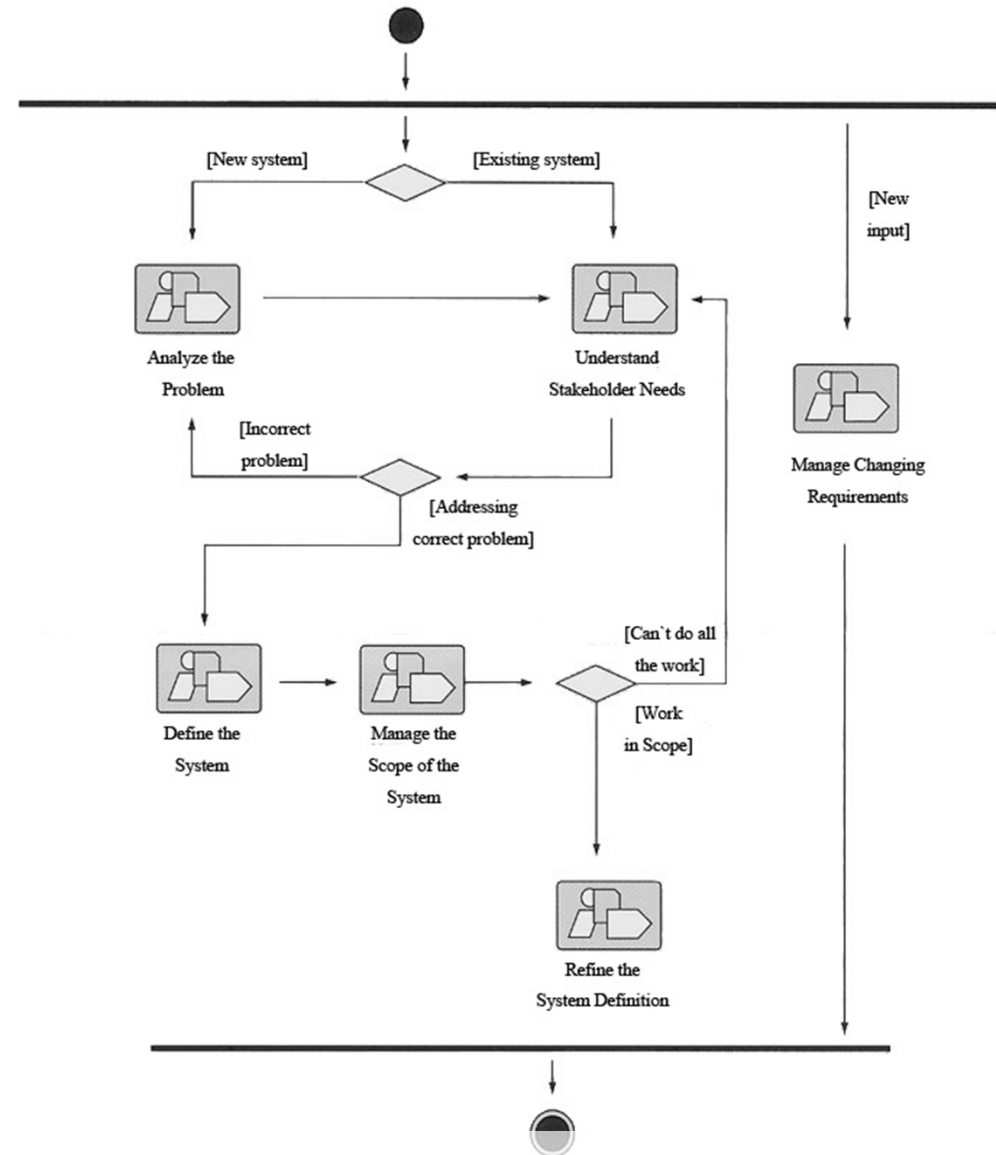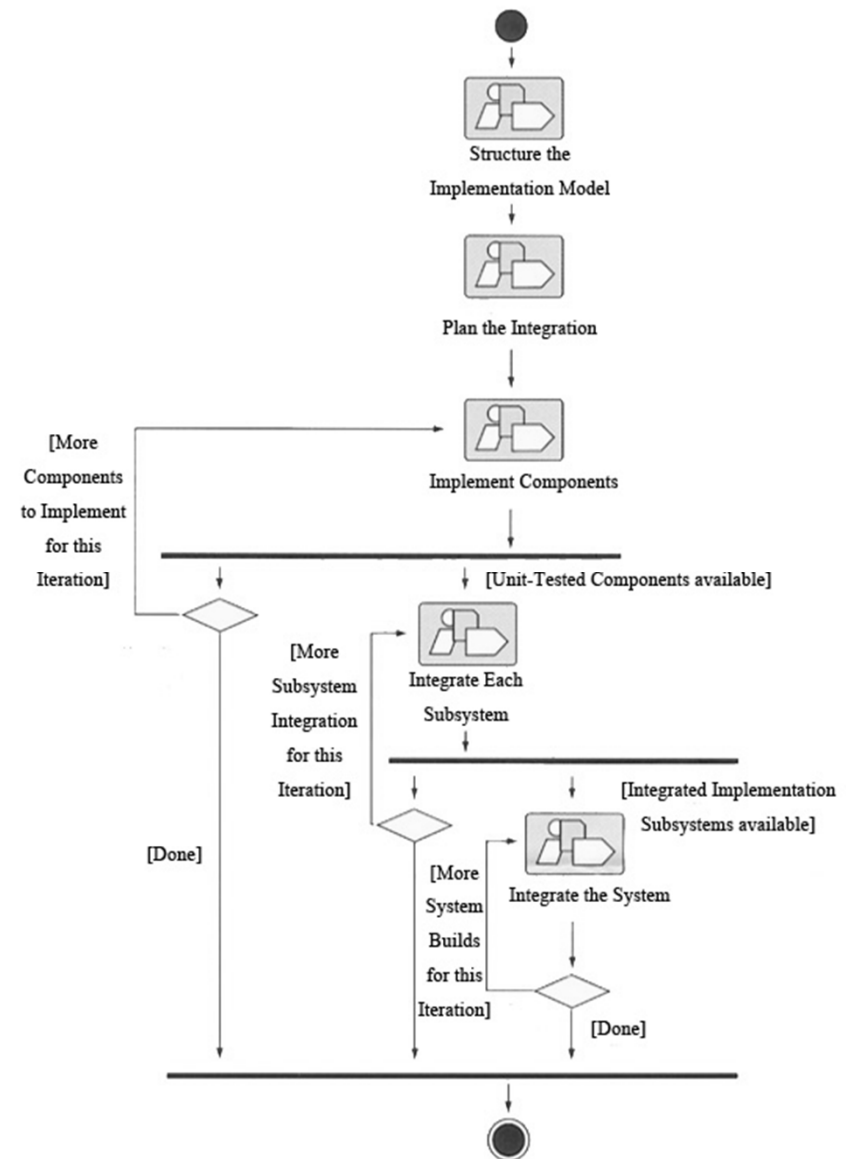
Analysis & Design

Purpose

➢ To turn the requirements into a design of the system-to-be

➢ To develop a comprehensive architecture for the system

➢ To adapt the design for performance

[Early elaboration iteration]

[Initial iteration (optional)]

Define a Candidate Achitecture

Execute Architecture Combination

[Elaboration iteration]

Refine the Architecture

Analyze Behavior

Design Components

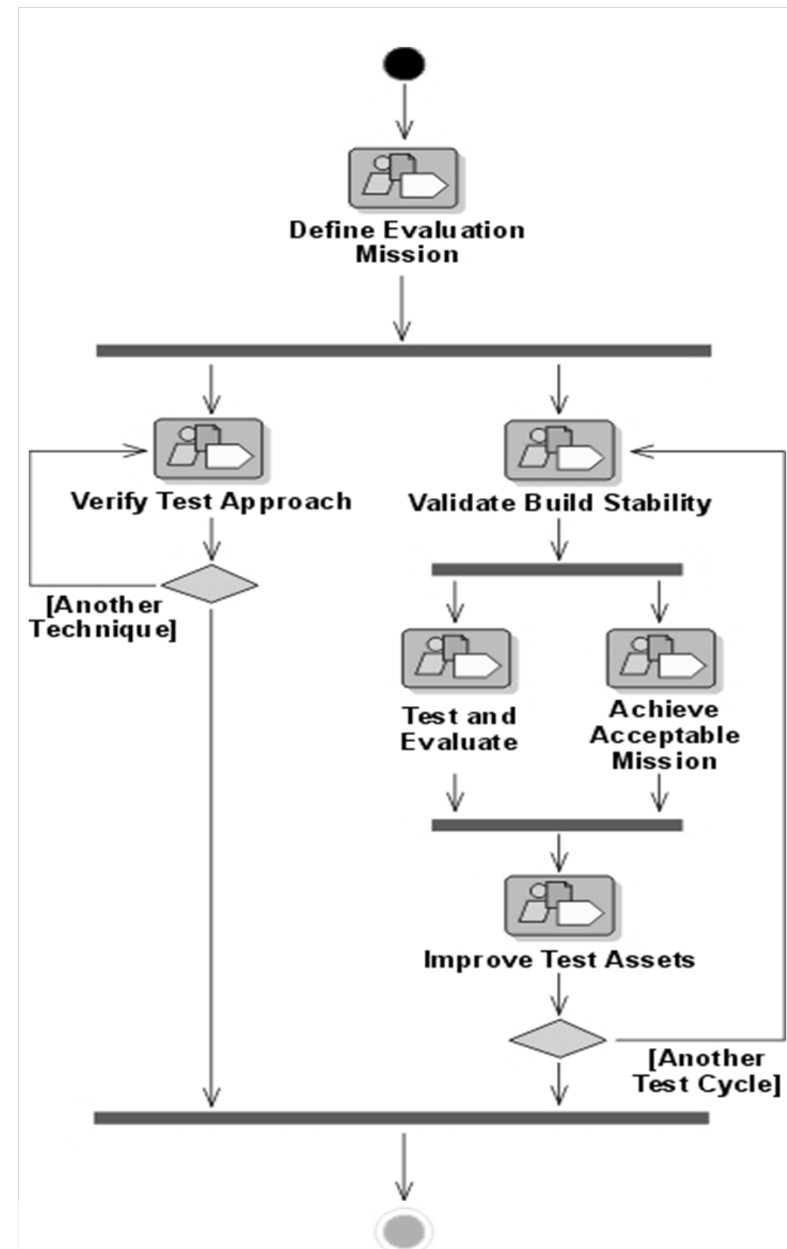Design Database

[Optional]

Implementation

Purpose
- To define the organization of the code, in terms of subsystems organized in layers
- To implement classes and objects in terms of components (source files, executables, and others),
- To test the developed components as units
- To integrate the results produced by individual developers (or teams), into an executable system

Structure the Implementation Model

Plan the Integration

Implement Components

[More Components to Implement for this Iteration]

[Unit-Tested Components available]

[More Subsystem Integration for this Iteration]

Integrate Each Subsystem

[Integrated Implementation Subsystems available]

[Done]

[More System Builds for this Iteration]

Integrate the System

[Done]

Test

Purpose
- ➢ To verify the interaction between objects
- ➢ To verify the proper integration of all components of the software
- ➢ To verify that all requirements have been correctly implemented
- ➢ To identify and ensure defects are addressed prior to the deployment of the software

Define Evaluation Mission
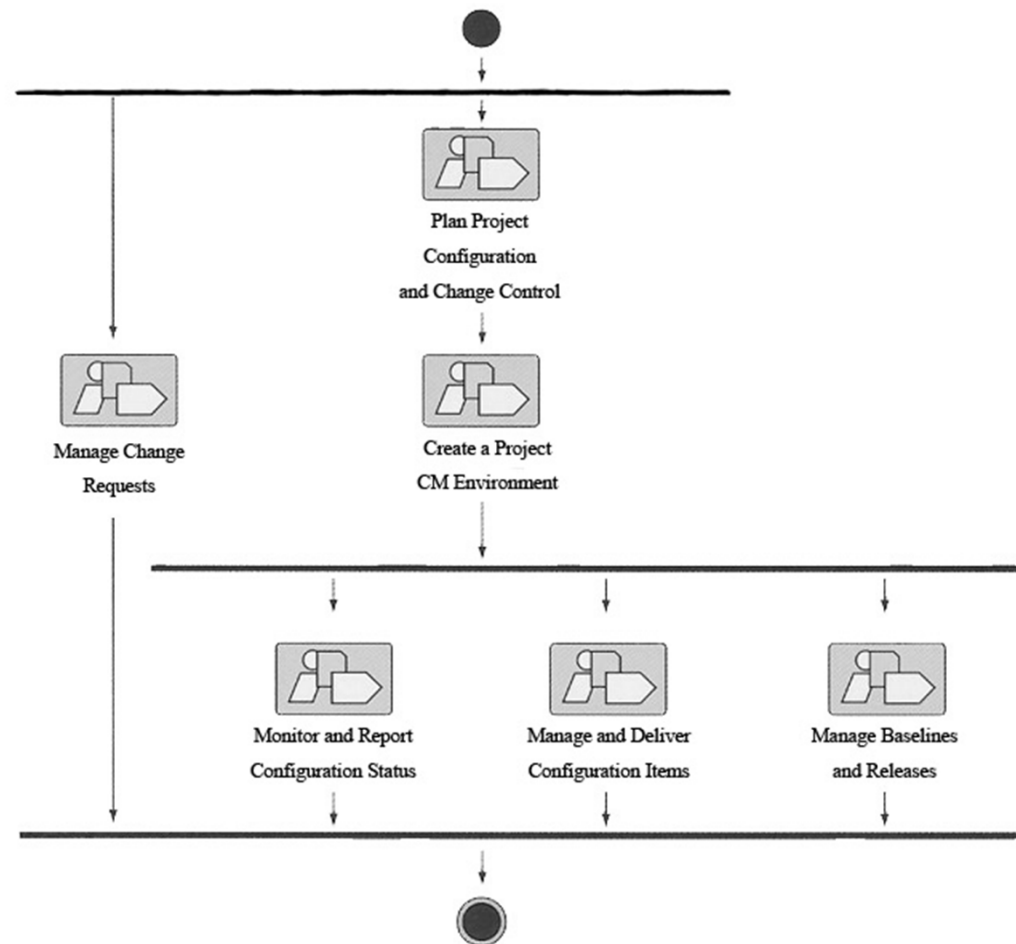
Verify Test Approach

Validate Build Stability

[Another Technique]

Test and Evaluate

Achieve Acceptable Mission

Improve Test Assets

[Another Test Cycle]

Deployment

Purpose

➢ To turn the finished software product over to its users

➢ three specific examples:
- ◆ Deployment of software in custom-built systems
- ◆ Deployment of shrink-wrapped software
- ◆ Deployment of software that is downloadable over the Internet
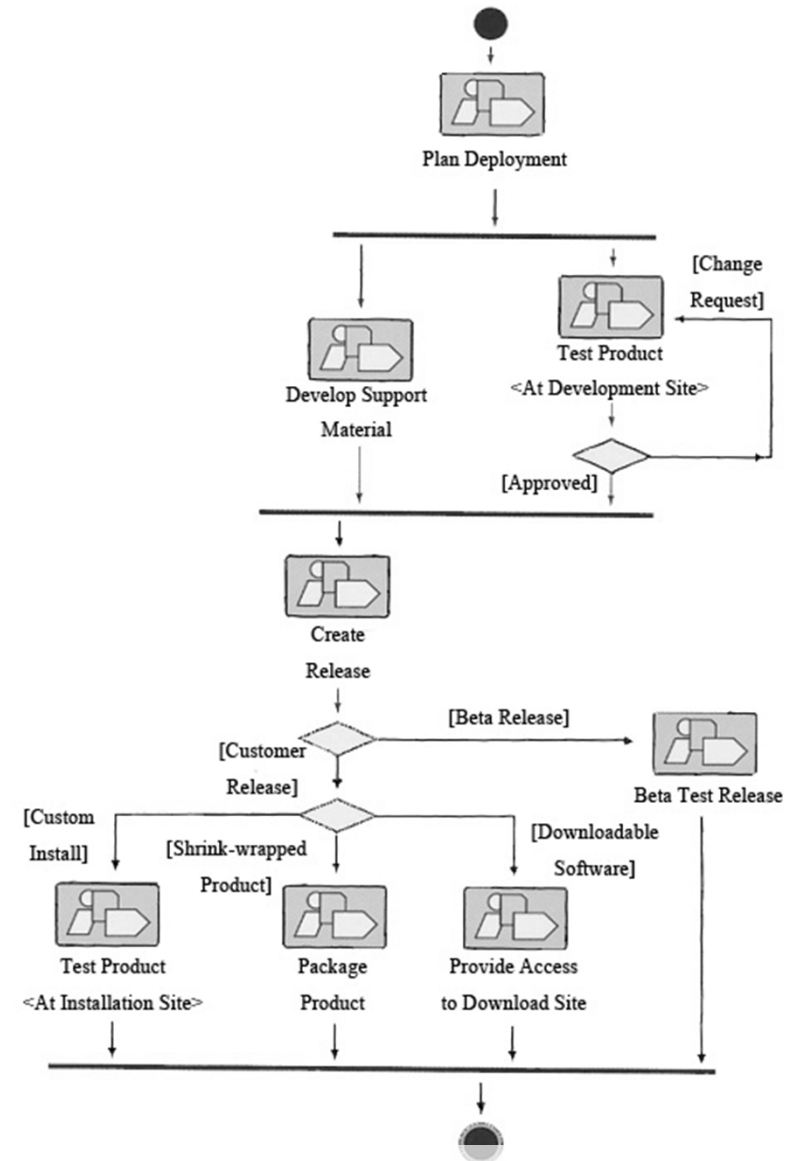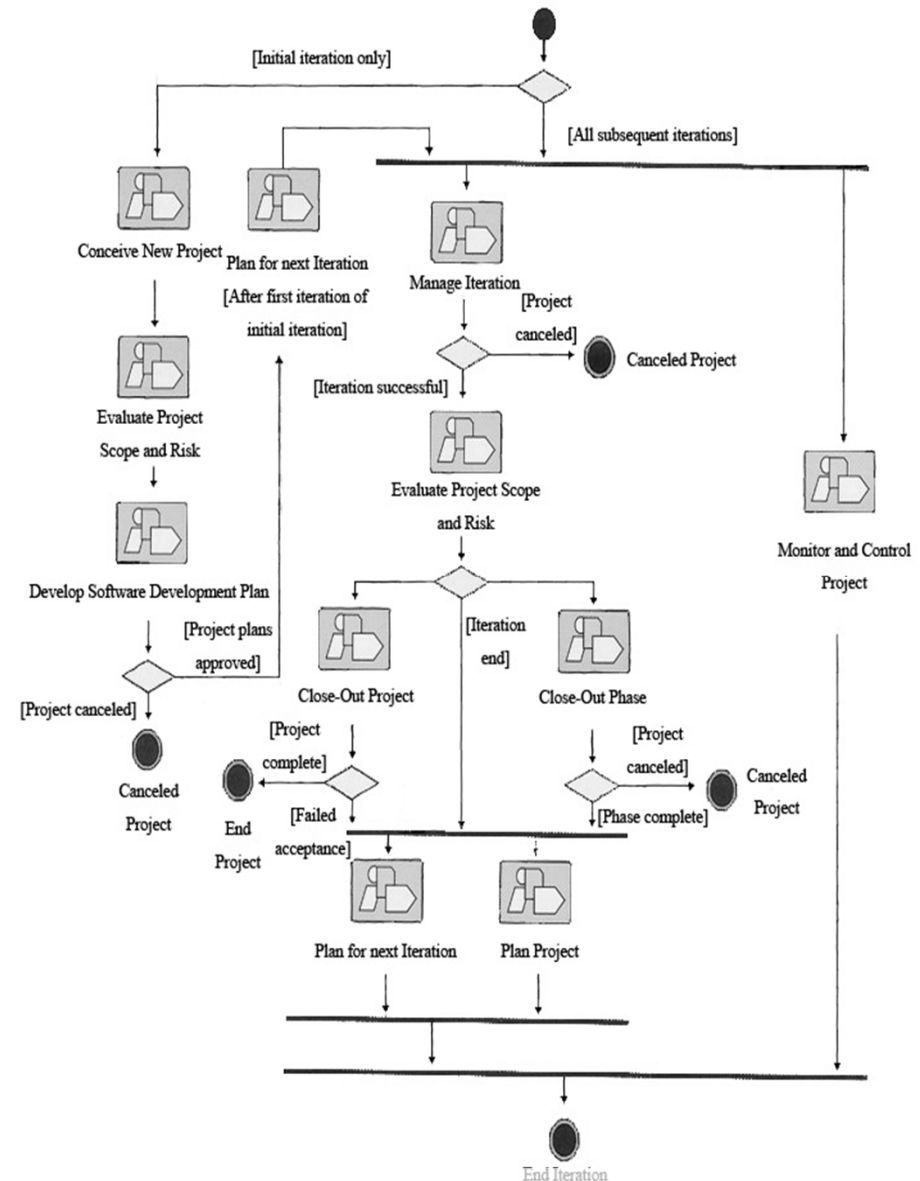
Configuration & Change Management

Purpose
- Identifying configuration items
- Restricting changes to those items
- Auditing changes made to those items
- Defining and managing configurations of those items
- Ensure completeness and correctness of the configured product
- Provide an audit trail on why, when and by whom any artifact was changed

Plan Project
Configuration
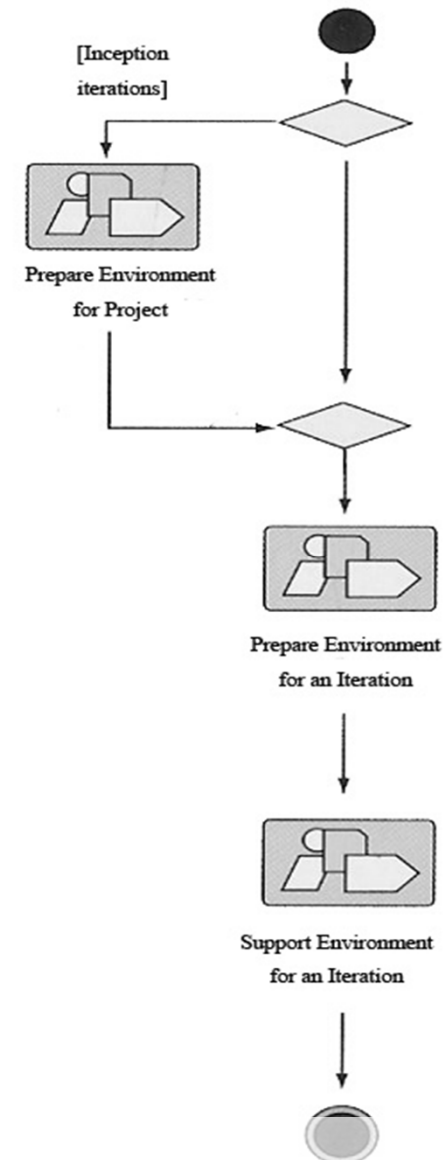and Change Control

Manage Change
Requests

Create a Project
CM Environment

Monitor and Report
Configuration Status

Manage and Deliver
Configuration Items

Manage Baselines
and Releases

Project Management

Purpose

➢ To provide a framework for managing software-intensive projects.

➢ To provide practical guidelines for planning, staffing, executing, and monitoring projects.

➢ To provide a framework for managing risk

➢ Risk management

➢ Planning an iterative project, through the lifecycle and for a particular iteration

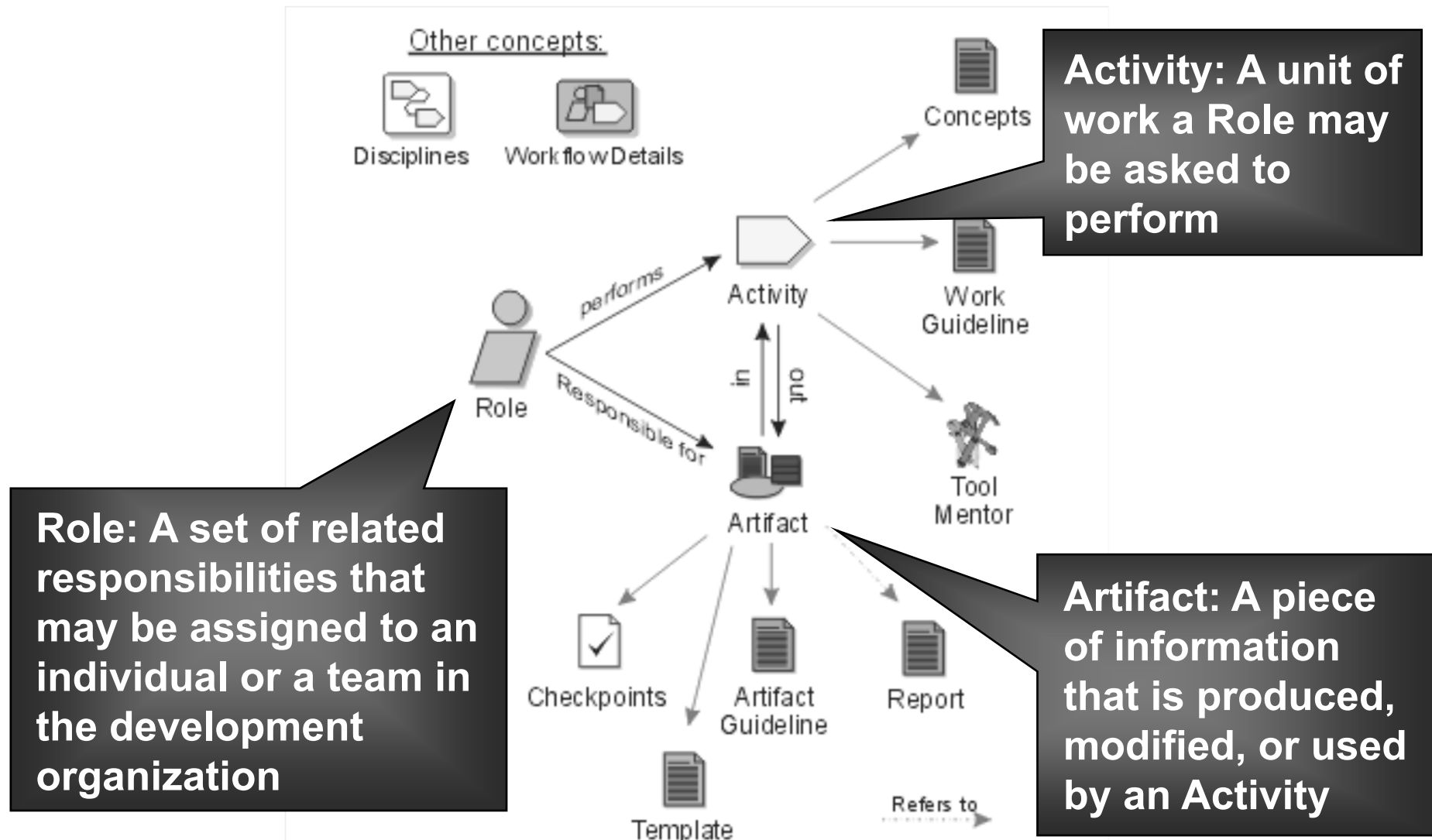➢ Monitoring progress of an iterative project, metrics

[Initial iteration only]

[All subsequent iterations]

Conceive New Project Plan for next Iteration

Manage Iteration

[After first iteration of initial iteration]

[Project canceled] Canceled Project

[Iteration successful]

Evaluate Project Scope and Risk

Evaluate Project Scope and Risk

Monitor and Control Project

Develop Software Development Plan

[Project plans approved]

[Iteration end]

[Project canceled]

Close-Out Project

Close-Out Phase

Canceled Project

[Project complete]

[Project canceled]

Canceled Project

End Project

[Failed acceptance]

[Phase complete]

Plan for next Iteration

Plan Project

End Iteration

Environment

Purpose

➢ To configure the process for a project

➢ to provide the software development organization with the software development processes and tools

[Inception iterations]

Prepare Environment for Project

Prepare Environment for an Iteration
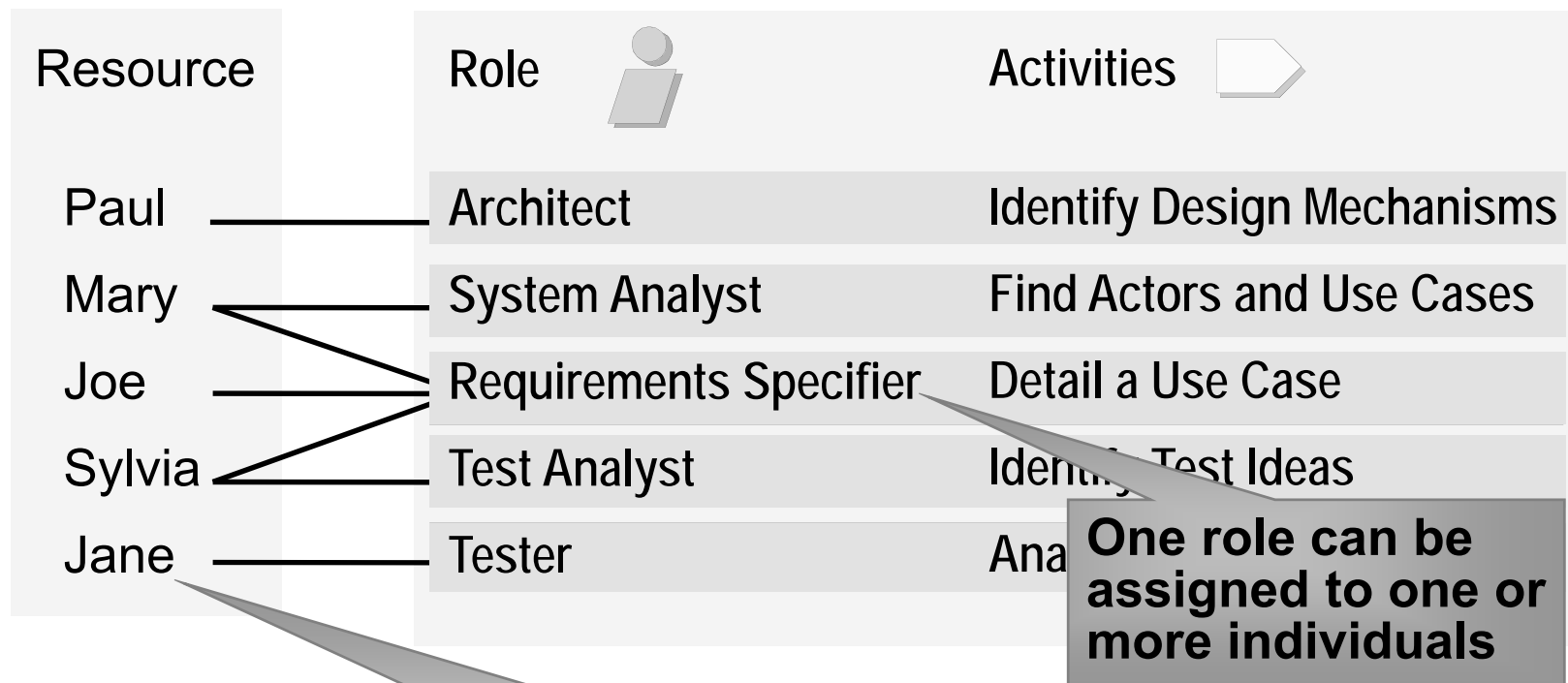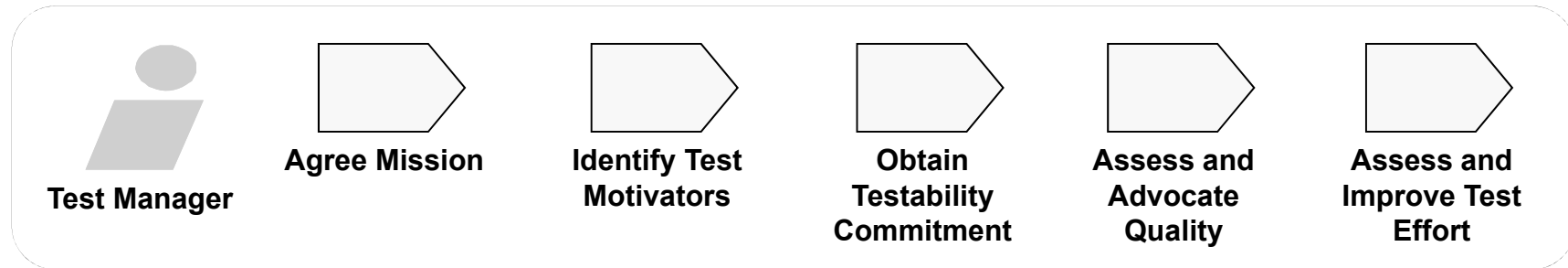
Support Environment for an Iteration

Module 5 - Agenda

- ◆ Overview of the software lifecycle in RUP
- ➜**Overview of the building blocks of RUP**
- ◆ Roles in the Test Discipline
- ◆ Workflow Details in the Test Discipline

Overview of Rational Unified Process Concepts

Activity: A unit of work a Role may be asked to perform

Role: A set of related responsibilities that may be assigned to an individual or a team in the development organization

Artifact: A piece of information that is produced, modified, or used by an Activity

Other concepts:

Disciplines   Workflow Details

Concepts

Activity

performs

Work Guideline

Role

Responsible for

in   out

Tool Mentor

Artifact

Checkpoints   Artifact Guideline   Report

Template   Refers to

Roles Are Used for Resource Planning

Resource	Role	Activities
Paul	Architect	Identify Design Mechanisms
Mary	System Analyst	Find Actors and Use Cases
Joe	Requirements Specifier	Detail a Use Case
Sylvia	Test Analyst	Identify Test Ideas
Jane	Tester	Analyze...

One role can be assigned to one or more individuals

**Each individual in the project is assigned to one or more roles**

文档仅限个人使用，请勿上传至互联网中，违者必究！

Module 5 - Agenda

- ◆ Overview of the software lifecycle in RUP
- ◆ Overview of the building blocks of RUP
- ➔ **Roles in the Test Discipline**
- ◆ Workflow Details in the Test Discipline

Roles in the Test Discipline

- **Test Manager**

- **Test Analyst**

- **Test Designer**

- **Tester**

RUP Test Manager Role, Activities, and Artifacts

Activities:

| Test Manager | Agree Mission | Identify Test Motivators | Obtain Testability Commitment | Assess and Advocate Quality | Assess and Improve Test Effort |

Artifacts:

Test Manager · Test Plan · Test Evaluation Summary
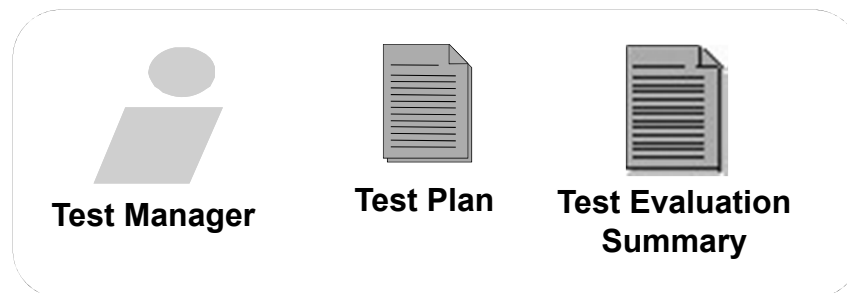
The **Test Manager** role is tasked with the overall responsibility for the test effort's success.

RUP Test Analyst Role, Activities, and Artifacts

Activities:

Test Analyst	Identify Targets of Test	Identify Test Ideas	Define Test Details	Define Assessment and Traceability Needs	Determine Test Results

Artifacts:
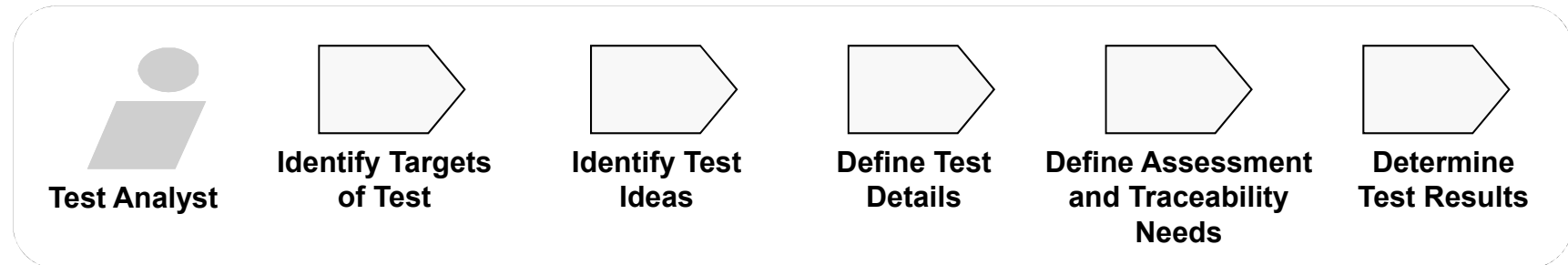
Test Analyst	Test Ideas List	Test Case	Workload Analysis Model	Test Data	Test Results
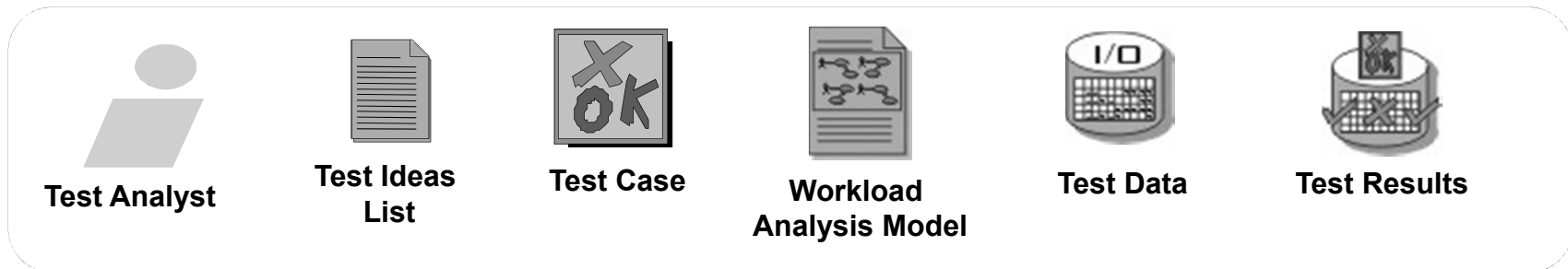
The **Test Analyst** role is responsible for initially identifying and defining the required tests, and subsequently evaluating the results of the test effort.

RUP Test Designer Role, Activities, and Artifacts

Activities:

Test Designer	Define Test Approach	Define Test Environment Configurations	Identify Testability Mechanisms	Structure the Test Implementation	Define Testability Elements	Develop Test Guidelines

Artifacts:
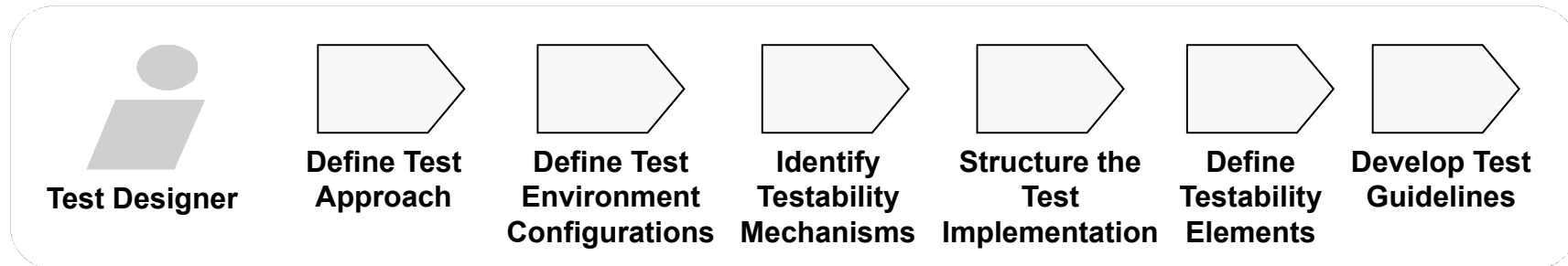
Test Designer	Test Automation Architecture	Test Interface Specification	Test Environment Configuration	Test Suite	Test Guidelines
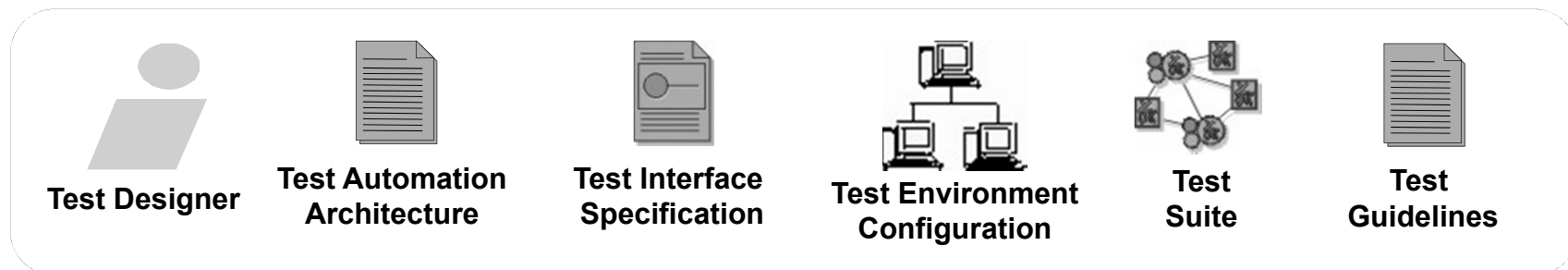
The **Test Designer** role is responsible for defining the test approach and ensuring its successful implementation.

RUP Tester Role, Activities, and Artifacts

Activities:

| Tester | Implement Test | Implement Test Suite | Execute Test Suite | Analyze Test Failure |

Artifacts:

Tester | Test Scripts | Test Log
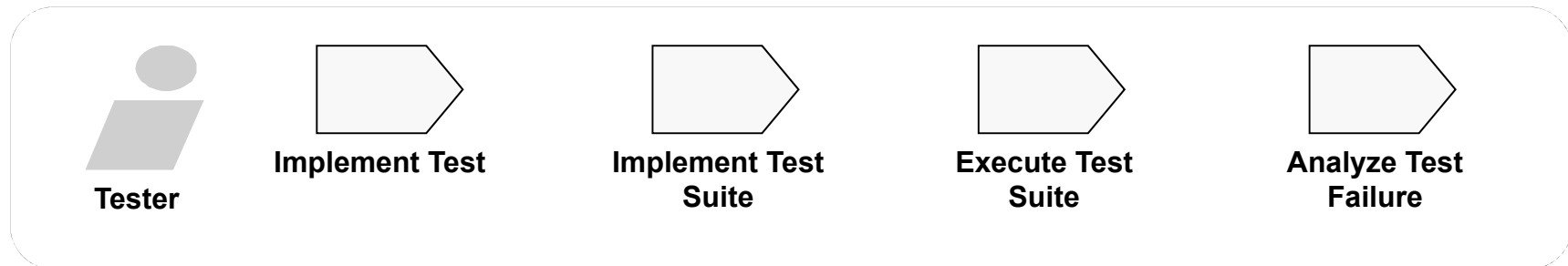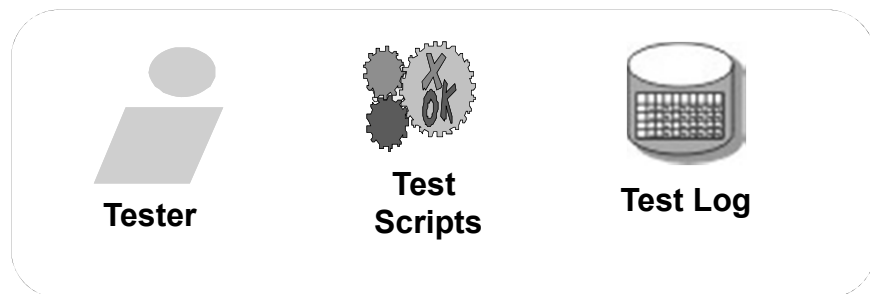
The **Tester** role is responsible for the core activities of the test effort, which involves conducting the necessary tests and logging the outcomes of that testing.

Module 5 - Agenda

- ◆ Overview of the software lifecycle in RUP
- ◆ Overview of the building blocks of RUP
- ◆ Roles in the Test Discipline
- ➔ **Workflow Details in the Test Discipline**

The RUP Test Discipline Workflow

- ◆ **Define Evaluation Mission**
  (module 6)

- ◆ **Test and Evaluate**
  (module 7 & module 8)

- ◆ **Achieve Acceptable Mission**
  (module 9)

- ◆ **Verify Test Approach**
  (module 10)

- ◆ **Validate Build Stability**
  (module 10)

- ◆ **Improve Test Assets**
  (module 10)
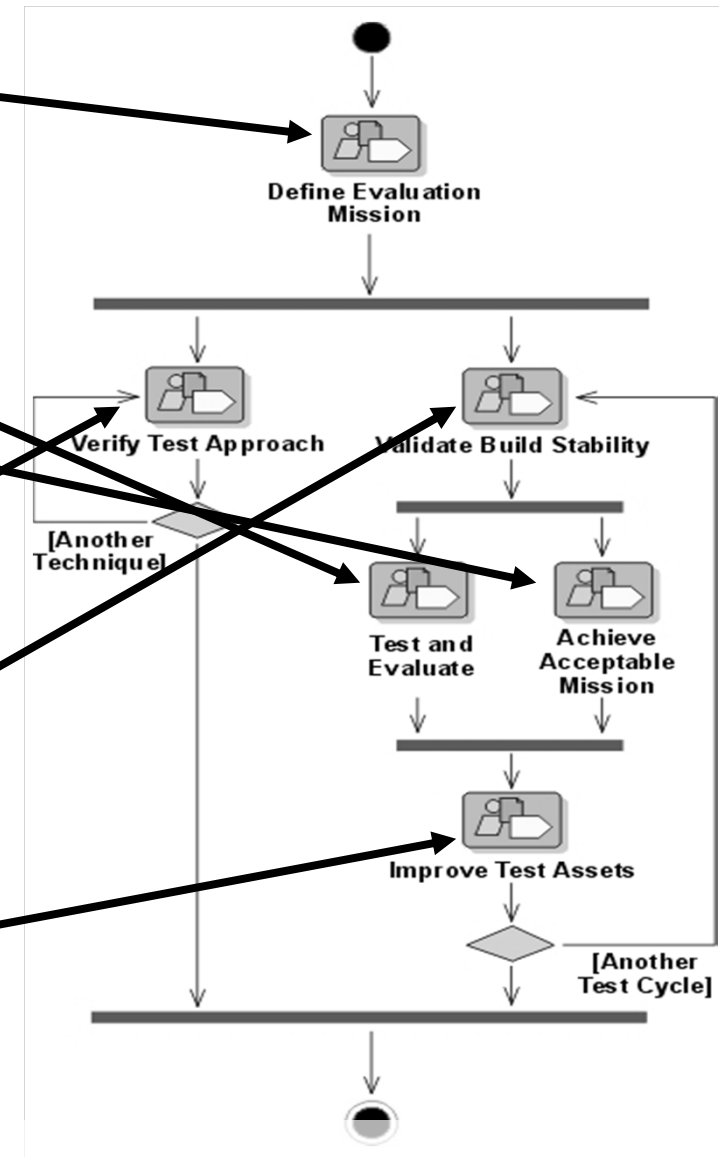
The RUP Test Discipline Workflow
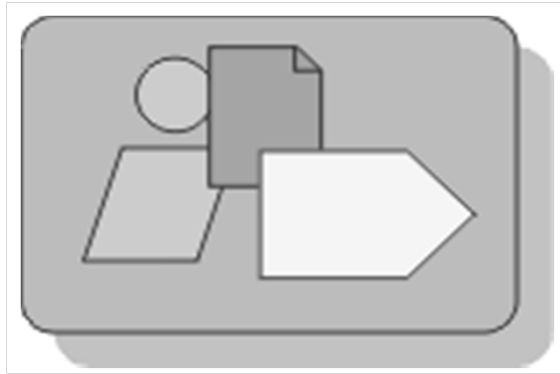
Define Evaluation Mission

- ◆ Identify the appropriate focus of the test effort for the iteration.

- ◆ Gain agreement with stakeholders on the corresponding goals that will direct the test effort.
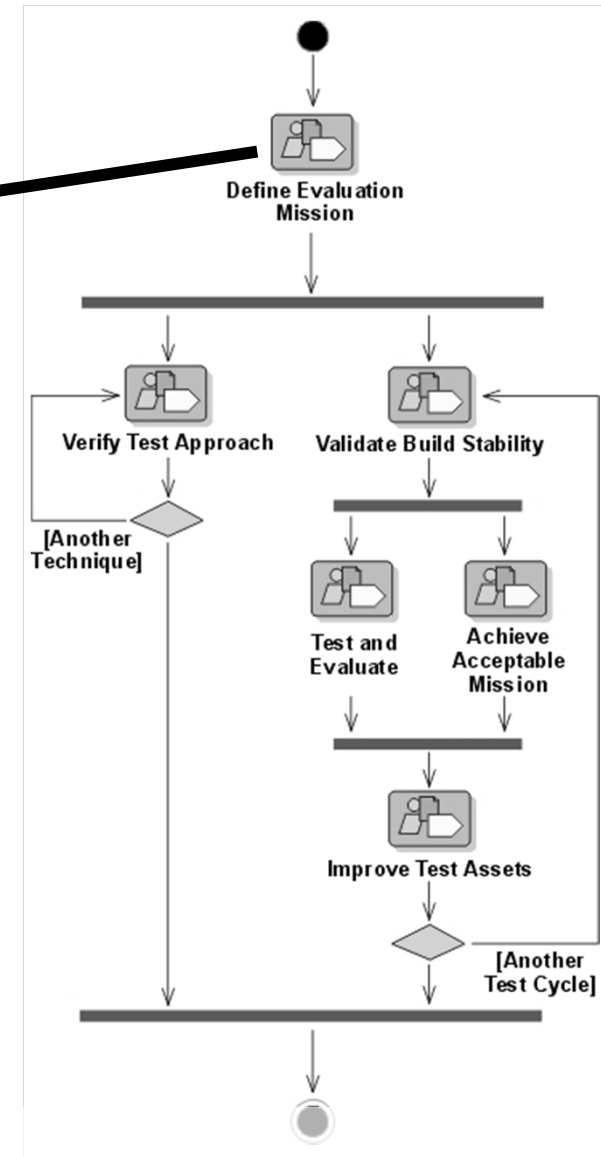
The RUP Test Discipline Workflow
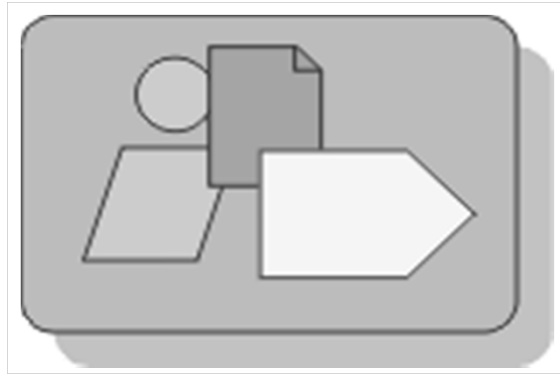
Test and Evaluate

◆ Achieve appropriate breadth and depth of testing to enable a sufficient evaluation of the targeted test items.
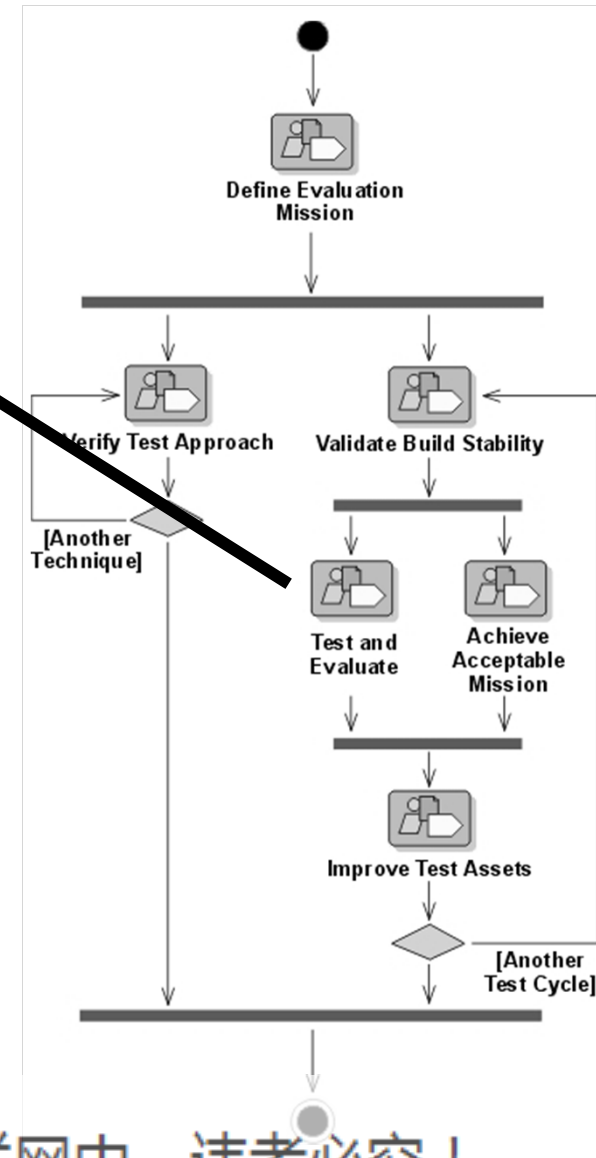
The RUP Test Discipline Workflow
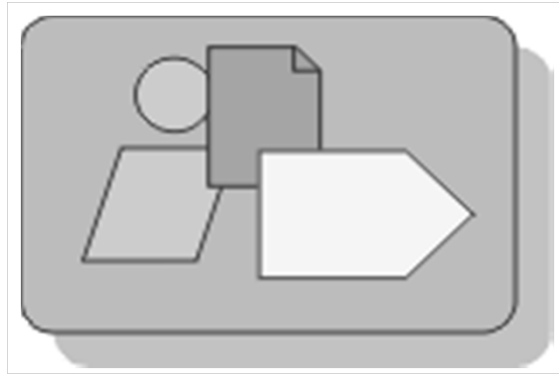
Achieve Acceptable Mission

- ◆ Deliver a useful evaluation result to the stakeholders of the test effort.
- ◆ Actively prioritize the test work that remains to be conducted.
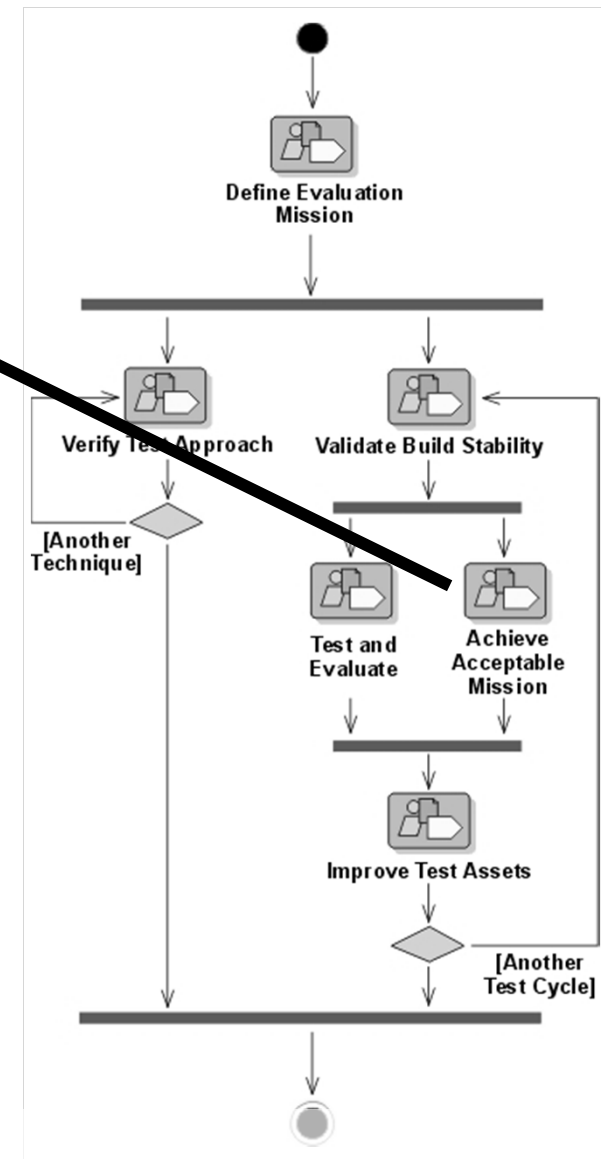
The RUP Test Discipline Workflow
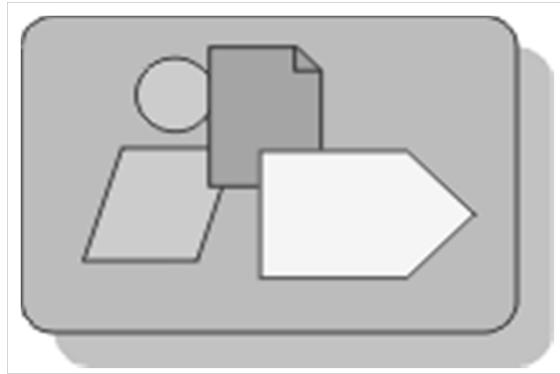
Verify Test Approach

- Demonstrate the techniques outlined in the Test Approach will support the required testing.

- Verify that the approach will work, produce accurate results and is appropriate for the available resources.
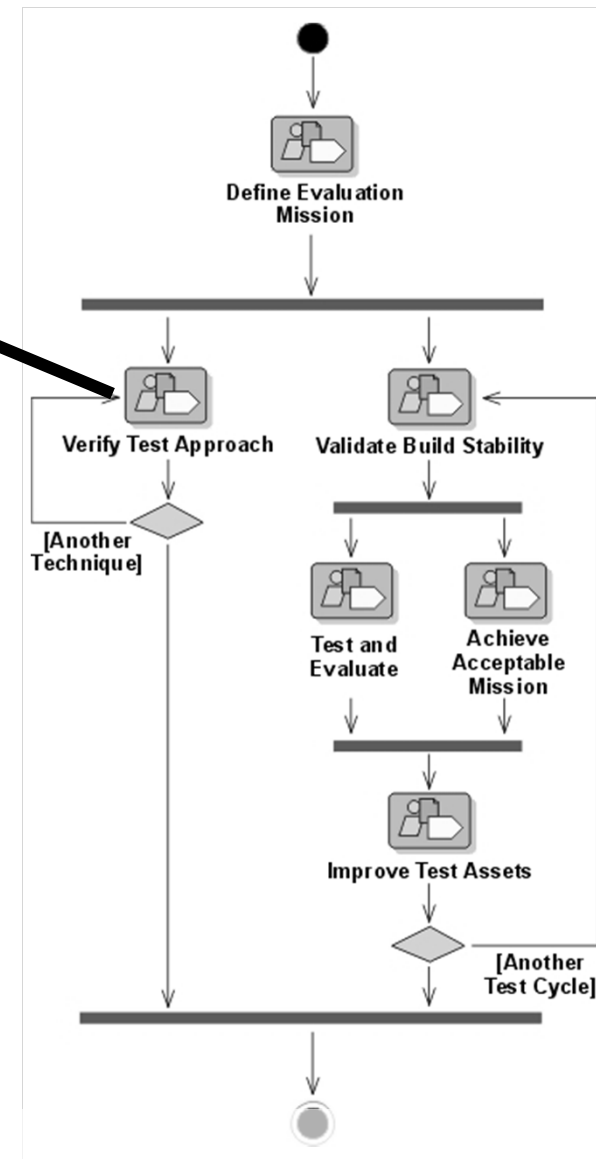
The RUP Test Discipline Workflow
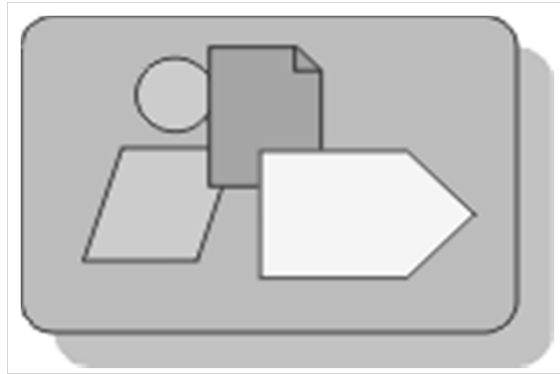
Validate Build Stability

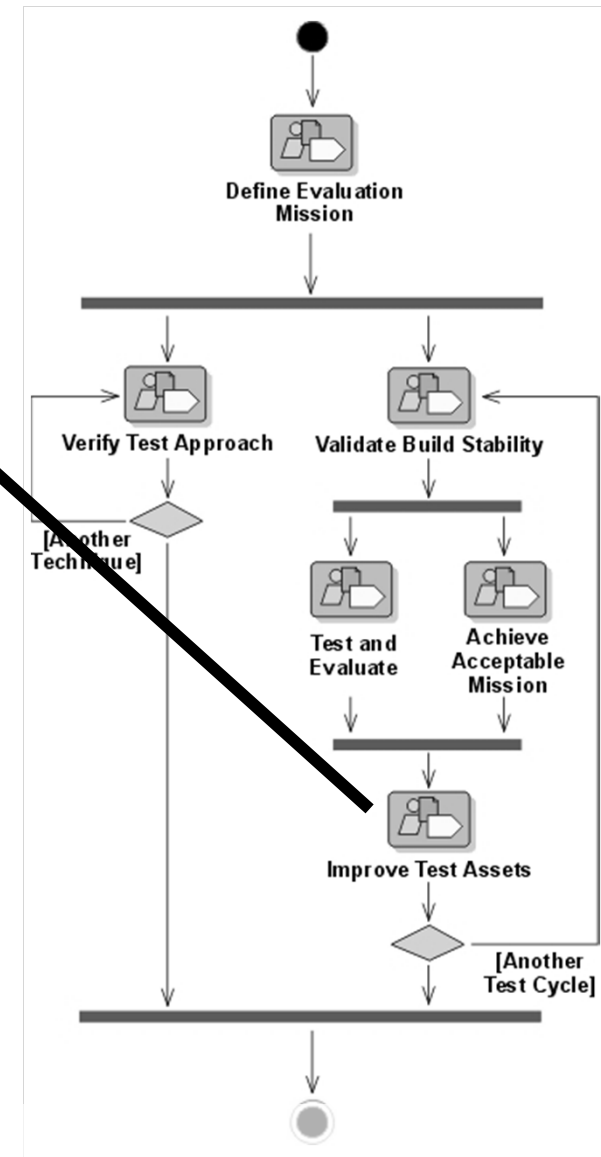◆ Validate that the build is stable enough for detailed test and evaluation work to begin.

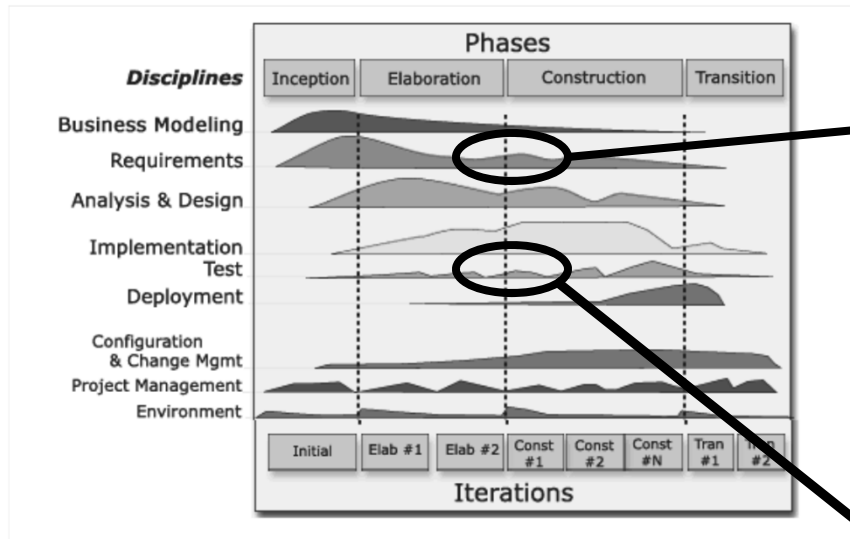文档仅限个人使用，请勿上传至互联网中，违者必究！
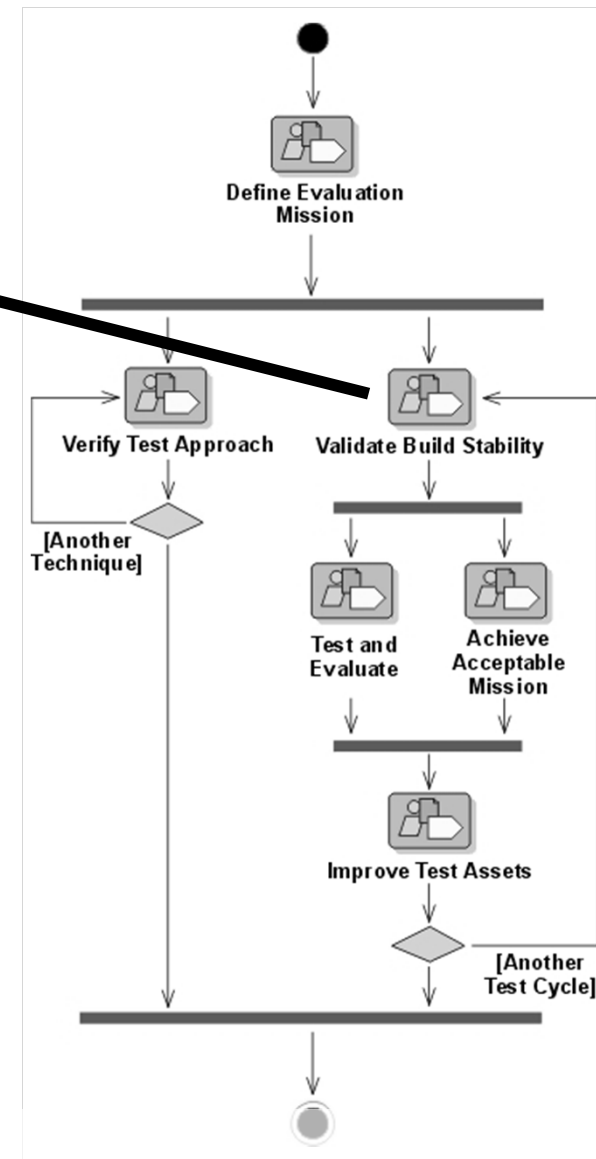
The RUP Test Discipline Workflow
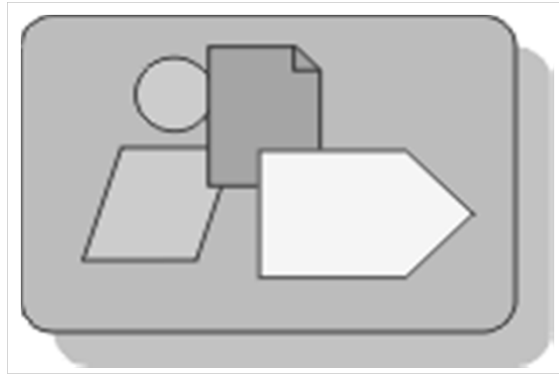
Improve Test Assets

◆ Maintain and improve the evolving test assets.

   ◆ (e.g. Maintain test suites and test data; harvest test-ideas into catalogs; clarify change request details)
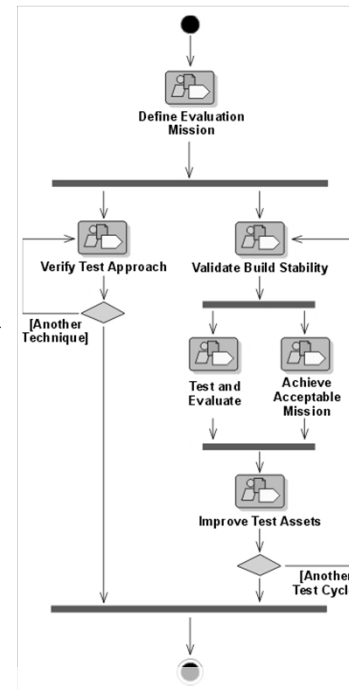
Discipline Workflows Guide Iterative Development
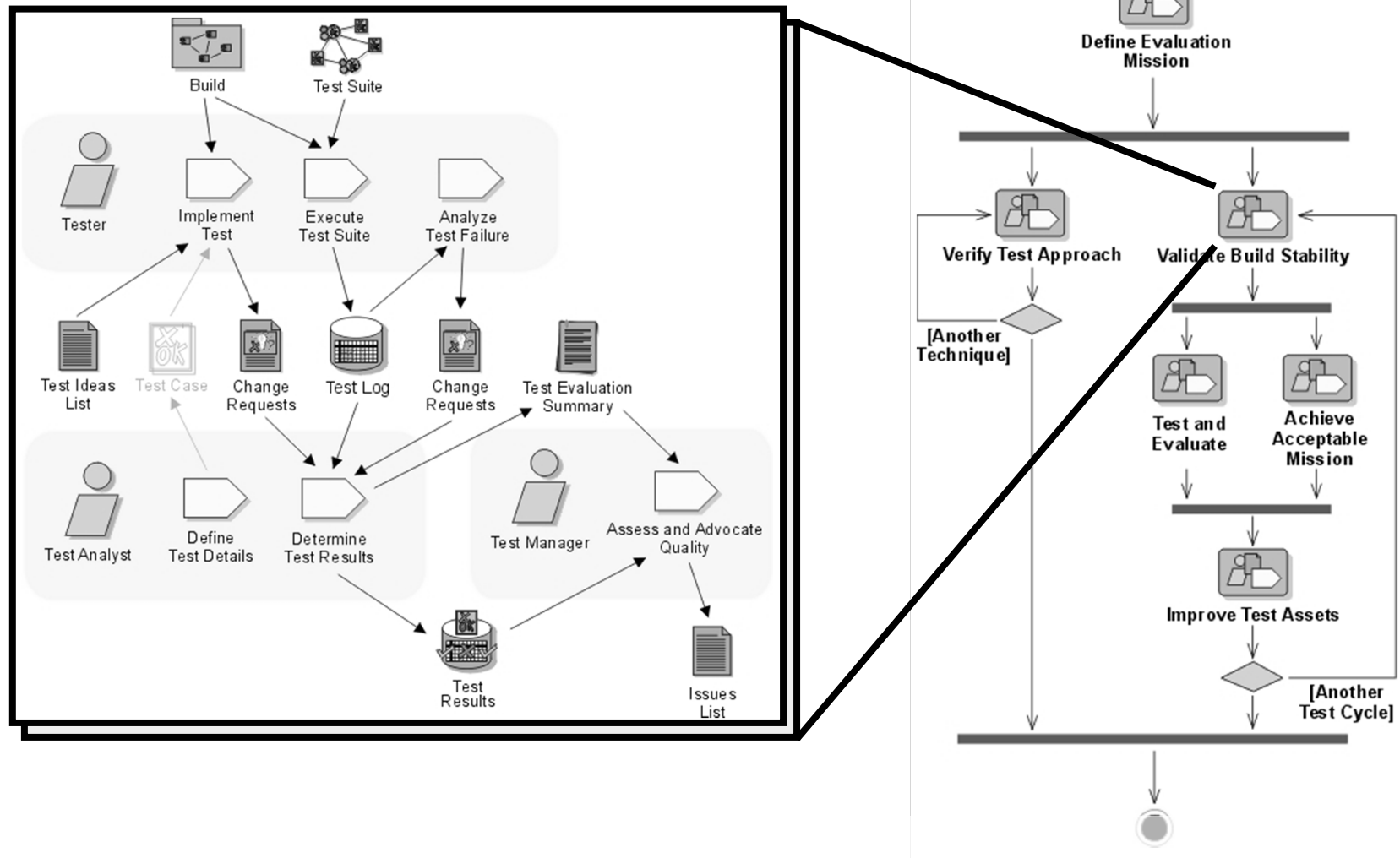

Example Workflow:
Requirements
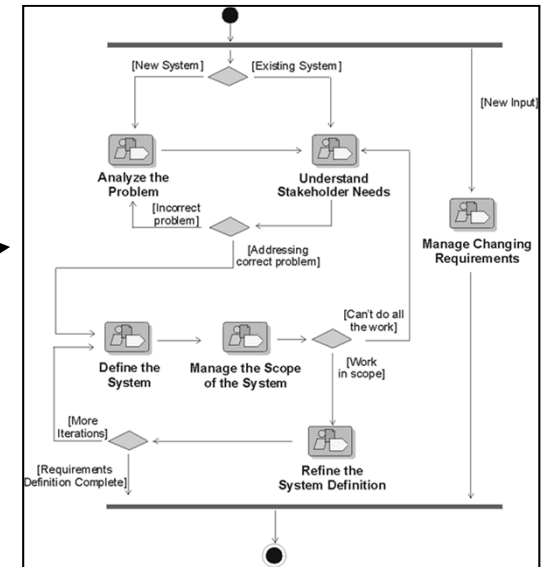
Example Workflow:
Test

Discipline Workflows Sequence the Workflow Details

Example Workflow Detail: Validate Build Stability

Module 5 - Review

The RUP Test Discipline:

- ◆ Presents an iterative testing process
- ◆ Is Scalable and Customizable
- ◆ Is designed for Flexibility

文档仅限个人使用，请勿上传至互联网中，违者必究！