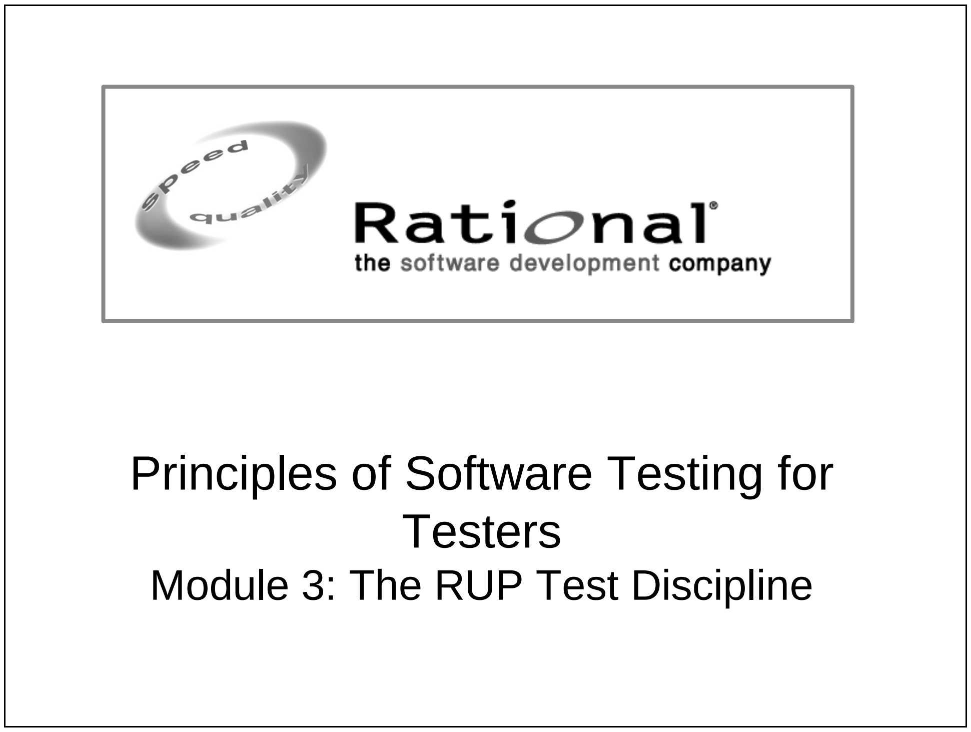
# Module 3
# The RUP Test Discipline



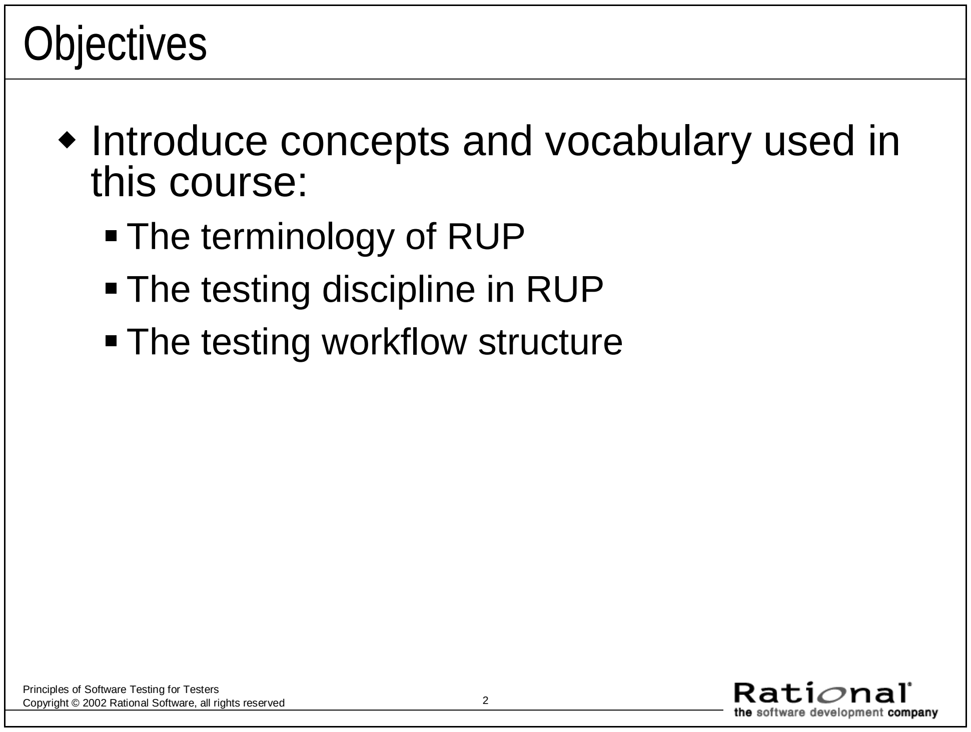
Rational® the software development company

Principles of Software Testing for Testers

Module 3: The RUP Test Discipline

## Topics

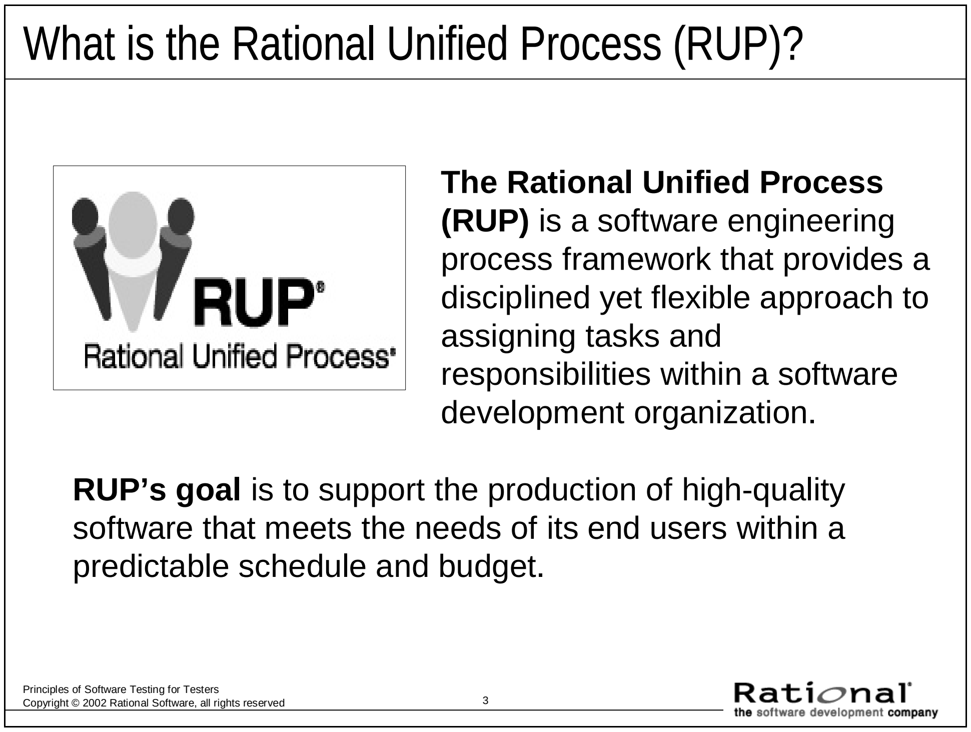
## Objectives



Objectives

- Introduce concepts and vocabulary used in this course:
  - The terminology of RUP
  - The testing discipline in RUP
  - The testing workflow structure

## What is the Rational Unified Process (RUP)?



### 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.

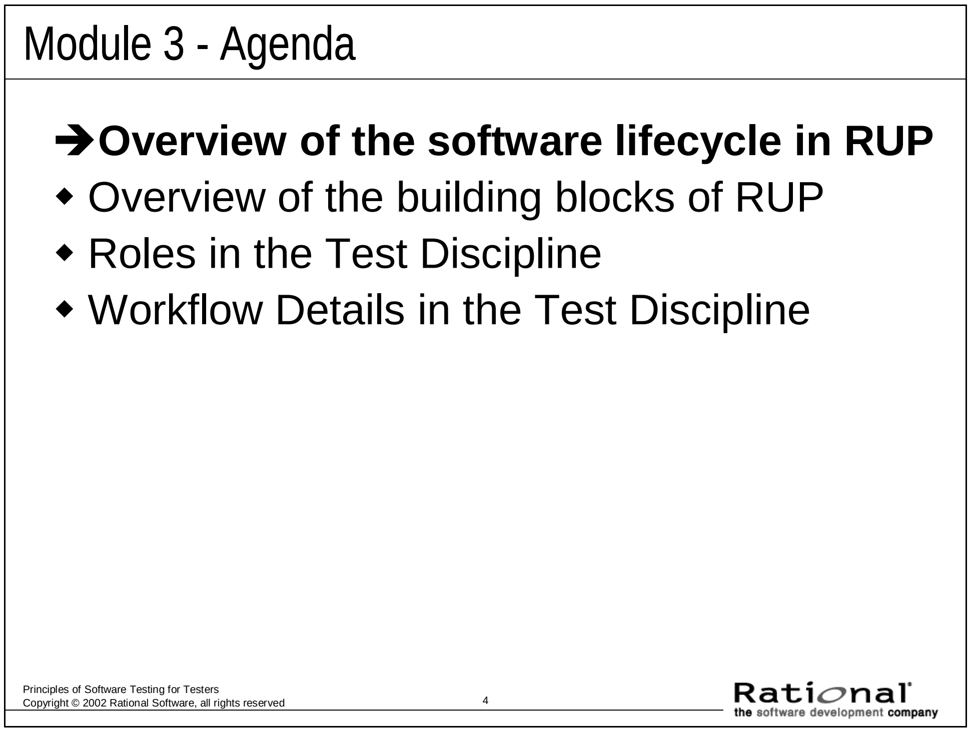
**RUP's goal** is to support the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.

The Rational Unified Process is a generic process framework for conducting object-oriented software engineering projects. It describes a family of related software engineering practices sharing a common structure and a common process architecture. The Rational Unified Process captures the proven practices in modern software development in a form that can be adapted for a wide range of projects and organizations.

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.

## Overview of the Software Lifecycle in RUP



Module 3 - 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

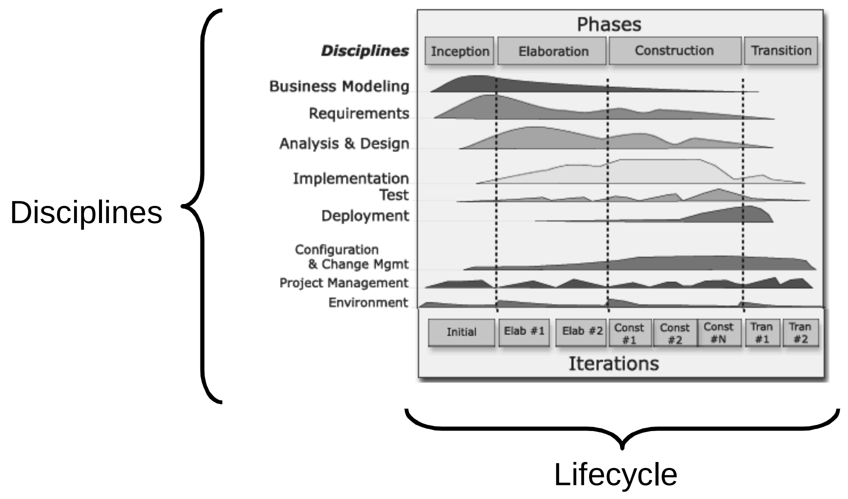
Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 4

Rational
the software development company

In this section, we introduce the structural elements of the Rational Unified process that are referred to as the software lifecycle in the RUP.

## RUP Process Architecture

### RUP Process Architecture



The figure at the top shows the overall architecture of the RUP.

The RUP has two dimensions:

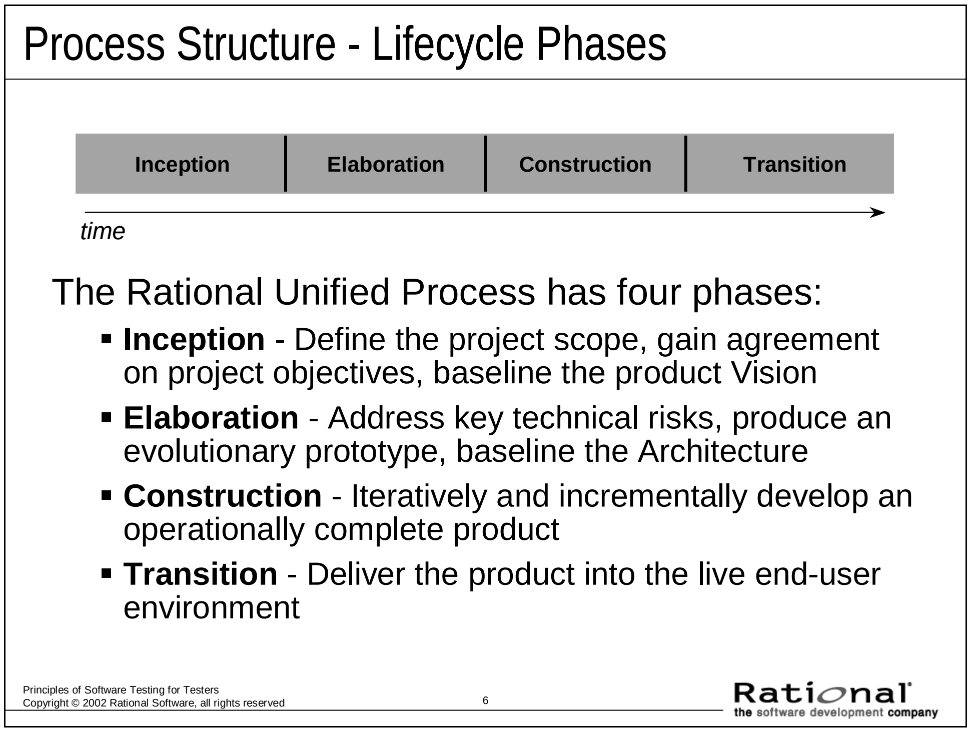
- The horizontal axis represents time and shows the lifecycle aspects of the process as it unfolds.
- The vertical axis represents disciplines, which group activities logically by nature.

The first dimension represents the "dynamic" aspect of the process as it is enacted, and it is expressed in terms of phases, iterations, and associated milestones.

The second dimension represents the "static" aspects of the process as it is described in terms of process components – the roles, activities, artifacts, and their related disciplines.

The graph shows how the emphasis varies over time. For example, in early iterations, we spend more time on requirements, and in later iterations we spend more time on implementation.

## Process Structure - Lifecycle Phases



### 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

During Inception, we define the scope of the project based on our initial understanding, identifying what is included, and what is not. We do this by identifying actors and use cases, and by outlining the most essential use cases (typically approximately 20% of the complete model). A business plan and a vision of the product are developed and assessed to determine whether resources should be committed to the project.

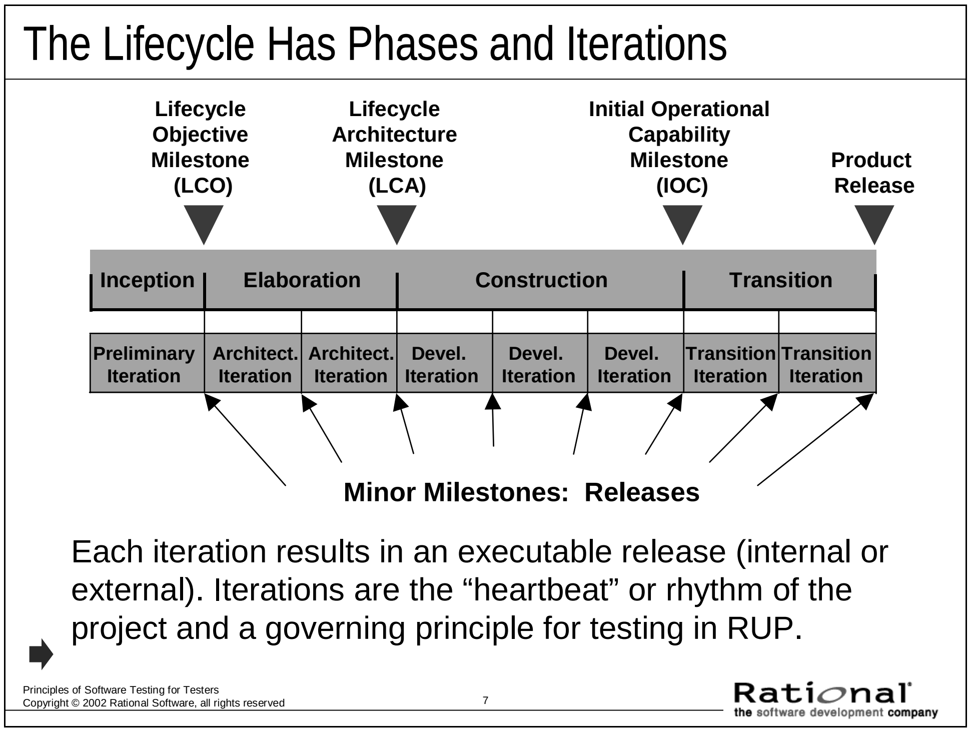
During Elaboration, we focus on three things: getting a good grasp of the requirements (80% complete), addressing key technical risks and establishing an architectural baseline that proves the key concepts of the solution. If we have a good grasp of the requirements and the architecture, we can eliminate a lot of the risk inherent in software development. This gives us a much better idea for what amount of work remains to be done. We can make detailed cost/resource estimations at the end of Elaboration with much more confidence.

During Construction, we build the product in several iterations, evolving it progressively into a complete operationally capable system. We might include a beta product release during or at the end of this phase.

During Transition, we stabilize the product and transition it into the end user's environment(s). We also focus on end user training, installation, and support.

The amount of time spent in each phase varies. For a very complex project with a lot of technical unknowns and unclear requirements, Elaboration will involve more iterations (e.g. 3-5). For a very simple project, where requirements are known and the architecture is simple, Elaboration may include only a single iteration.

## The Lifecycle Has Phases and Iterations



### The Lifecycle Has Phases and Iterations

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

At each of the major milestones, we review the project and decide whether to proceed with the project as planned, to abort the project, or to revise it. The criteria used to make this decision vary by phase.

*Definitions:*

*LCO: scope agreed upon and risks understood and reasonable*

*LCA: key risks addressed and architecture stable*

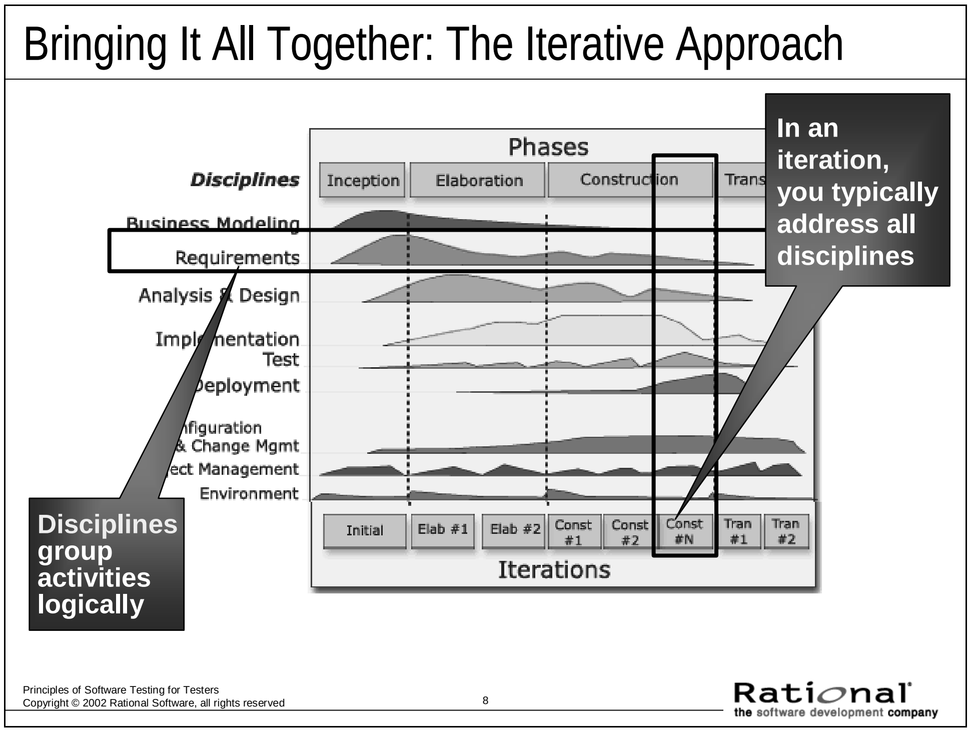
*IOC: product is operationally complete and quality acceptable*

Within each phase, there is a series of iterations. The number of iterations per phase will vary. Each iteration results in an executable release (either internal or external) encompassing larger and larger subsets of the final application.

An internal release is kept within the development environment and ideally demonstrated to a representative portion of the stakeholder community. More significant external releases, typically for installation in the end-user environment, are also provided. External releases are much more expensive (they usually involve more ceremony and therefore more resources) and thus typically occur at important milestones.

The end of an iteration marks a minor milestone. At this point, we assess technical results and revise future plans as necessary.

## Bringing It All Together: The Iterative Approach



Bringing It All Together: The Iterative Approach

In an iteration, you typically address all disciplines

Disciplines group activities logically

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved
8

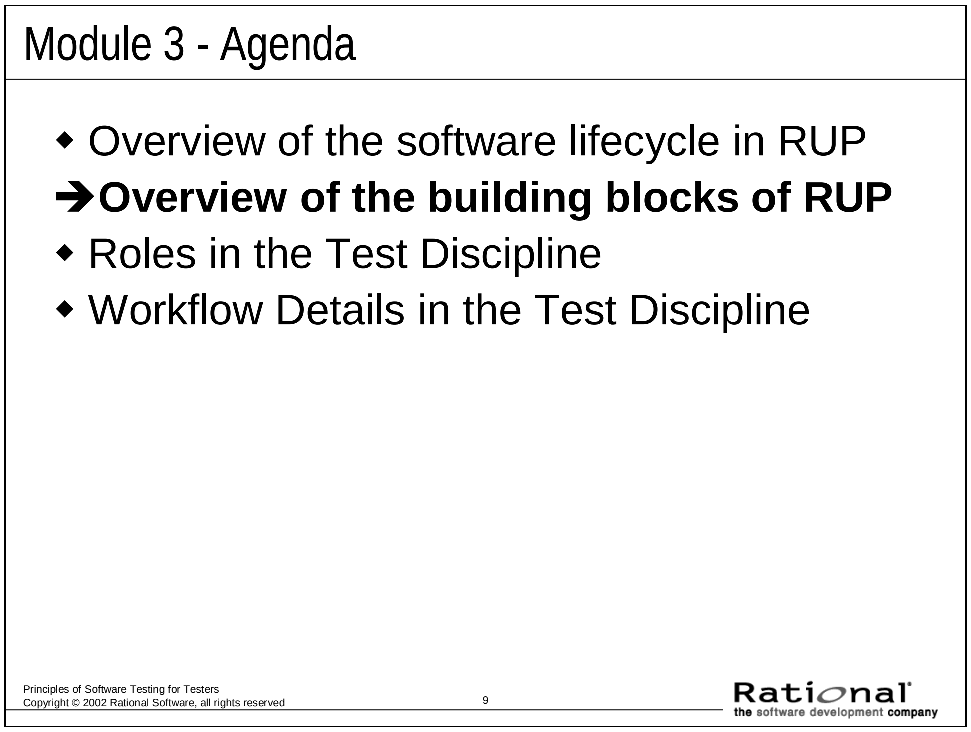
Rational the software development company

This graphic illustrates how phases and iterations (the “dynamic” or time dimension of RUP) relate to the development activities described in the disciplines (the “static” dimension). Note that while the graphic simply provides an example of how RUP might be enacted, the relative size of the colored graphs gives a general indication of how much relative effort is spent for each process discipline in each phase/ iteration.

Notice that with a few exceptions, each iteration involves activity in all disciplines, and that the relative amount of effort expended in each discipline changes between iterations. For instance, during late Construction, the main effort is related to Implementation, Test and Deployment with minimal effort expended on Requirements and Environment work.

Note that in an iterative development process, requirements work is typically not “complete” early in the project lifecycle – requirements effort typically continues into late Construction. It is also common for the final analysis and design work for well-understood portions of the system to be delayed until Construction – this can be supported because this incomplete design represents minimal unaddressed risk.

## Overview of the Building Blocks of RUP



### Module 3 - 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

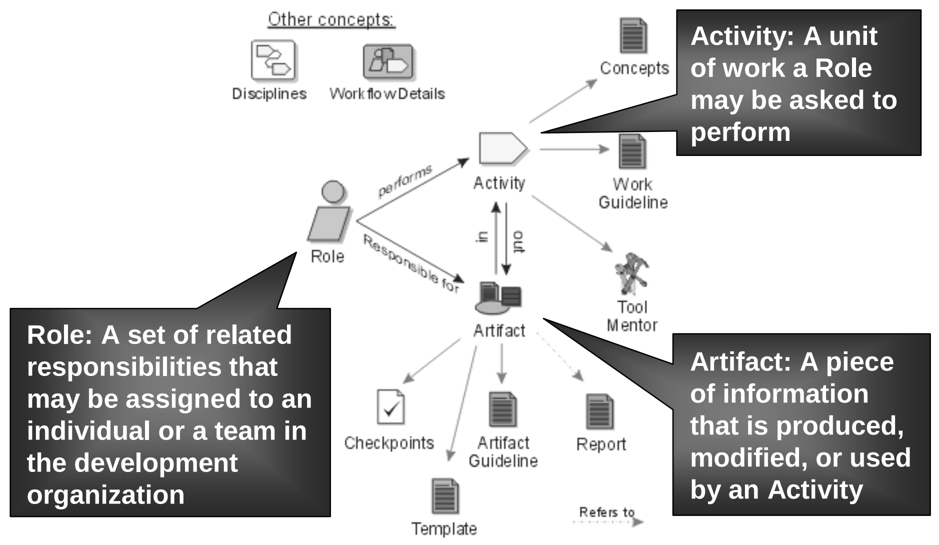
Principles of Software Testing for Testers

9
Rational
the software development company

Next, we'll introduce the major static structural elements of RUP that are used to define the detailed and unique process elements.

## Overview of Rational Unified Process Concepts

### Overview of Rational Unified Process Concepts


Other concepts:
Disciplines
WorkflowDetails
Concepts
Activity: A unit of work a Role may be asked to perform
performs
Activity
Work Guideline
Role
Responsible for
in
out
Tool Mentor
Artifact
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
Checkpoints
Artifact Guideline
Report
Template
Refers to

10

Rational
the software development company

A **Role** is an abstract definition of a set of related behavior and responsibilities that will be fulfilled by an individual, or a set of individuals working together as a team.

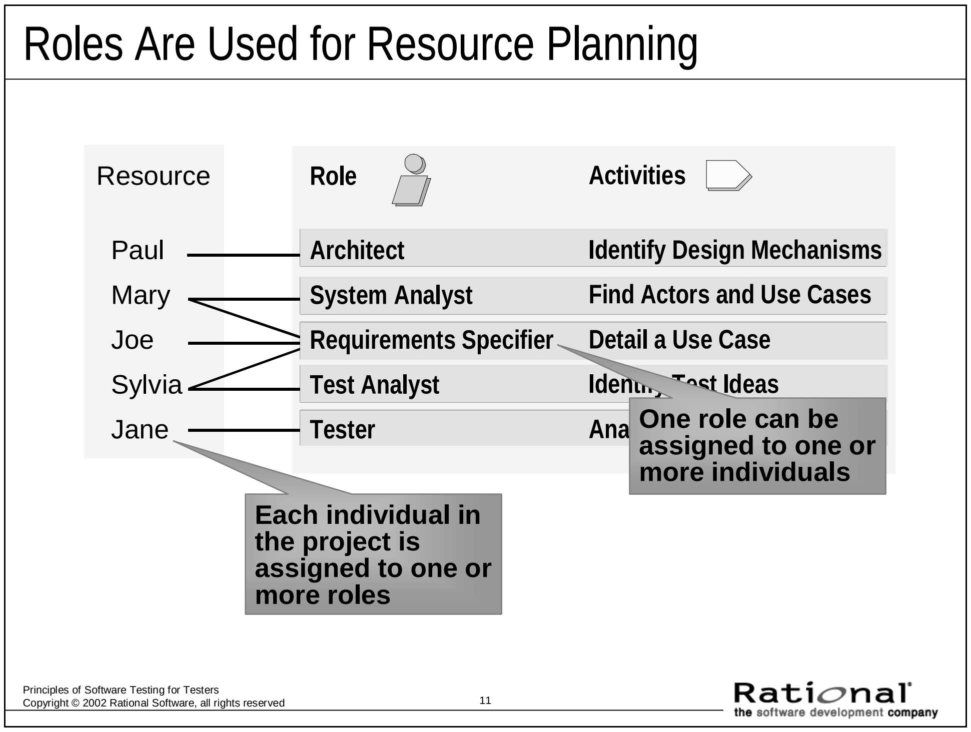
An **Activity** is the smallest piece of work that is useful to define in terms of repeatable process. Dividing the work in this manner makes it easier to monitor development. A specific role is responsible for one or more Activities.

**Artifacts** are the work products of enacting the process. They are produced in the course of developing the software product –Activities evolve, maintain, or make use of Artifacts as input. This includes the source code itself, as well as the models, documents, and other products of the lifecycle. The UML provides notation for representing many of the artifacts of the development process.

**Some other basic terminology in the Rational Unified Process:**

- **Concepts** – provide information which is important for understanding the workflow.
- **Guidelines** – provides artifact guidelines with descriptive information about an artifact type, and work guidelines containing practical information about how to perform certain tasks.
- **Tool Mentors** – offer support for software-engineering tools.
- **Checkpoints** – provide a quick reference to help you assess the quality and completeness of an artifact.
- **Templates** – a number of ready-to-use templates for certain artifacts are provided.

## Roles Are Used for Resource Planning



### 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 | Ana |

One role can be assigned to one or more individuals

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

Rational the software development company

In developing a project plan, a project manager assigns the available individuals to roles according to their skills and abilities. The project manager assigns each individual on the project to one or more roles. The association of individuals to roles is dynamic over time.

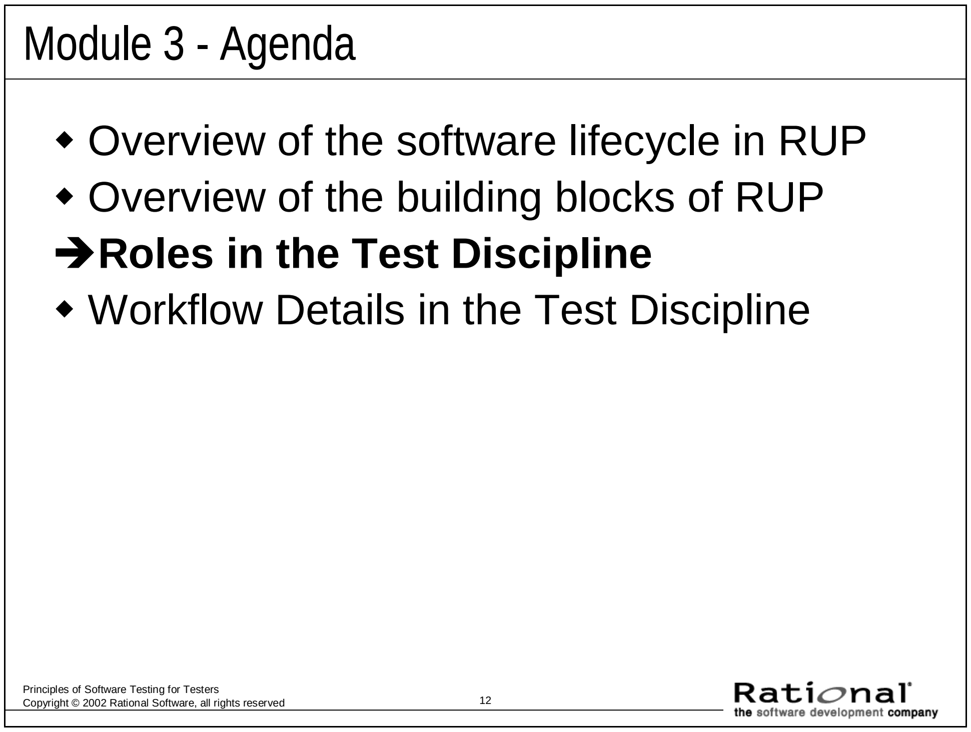
A project team member often fulfills many different roles. Roles are not individuals; instead, they describe how individuals behave in the business and what responsibilities these individuals have.

An individual may act as several different roles during the same day. We can informally call this “wearing several hats.” For example, Sylvia may be both a Requirements Specifier and a Test Analyst.

Several individuals may act in the same role to perform a certain activity as a team. For example, Mary, Joe and Sylvia may all serve as Use-Case Specifiers.

Artifacts are the responsibility of a single role, to help manage accountability. However, even though one role "owns" the artifact, different roles usually collaborate in evolving the artifact throughout its life.

## Roles in the Test Discipline



Module 3 - Agenda

- Overview of the software lifecycle in RUP
- Overview of the building blocks of RUP

➔**Roles in the Test Discipline**

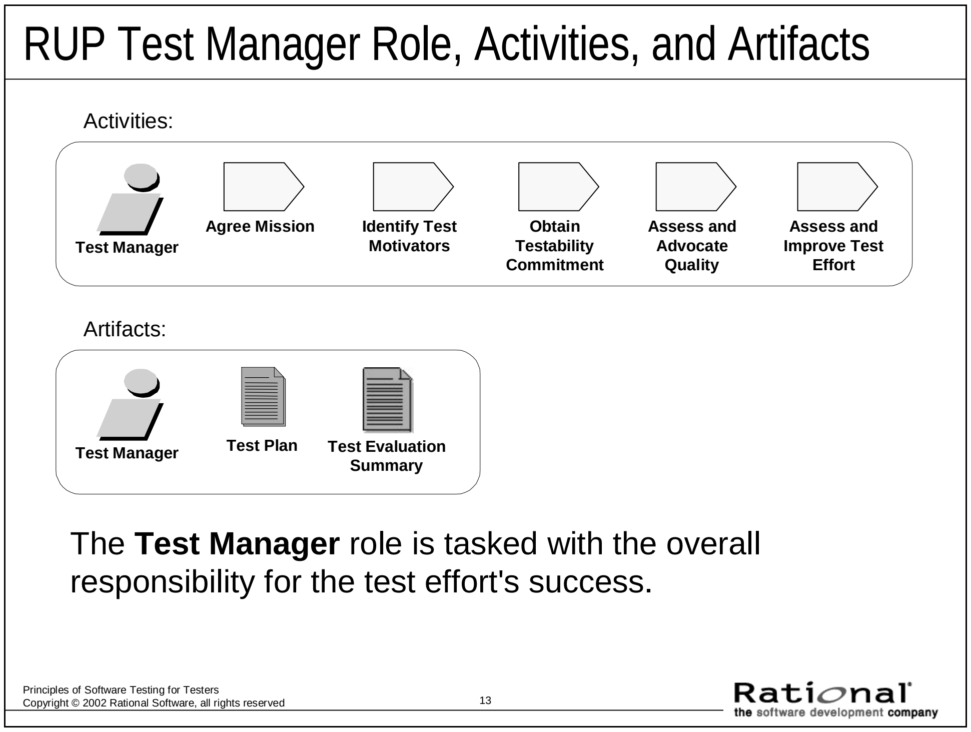
- Workflow Details in the Test Discipline

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 12

Rational
the software development company

In this section, we introduce the test-related roles in the Rational Unified Process.

## RUP Test Manager Role, Activities, and Artifacts



### 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

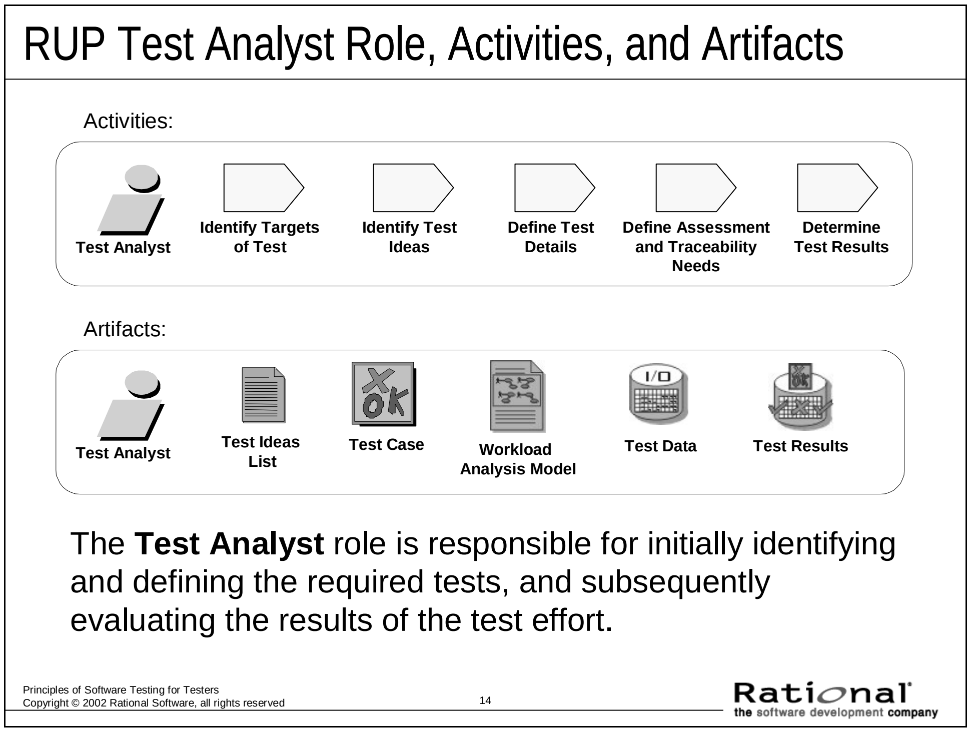
Artifacts:

Test Manager
Test Plan
Test Evaluation Summary

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

The **Test Manager** role is tasked with the overall responsibility for the test effort's success. The role involves quality and test advocacy, resource planning and management, and resolution of issues that impede the test effort.

## RUP Test Analyst Role, Activities, and Artifacts



### 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.

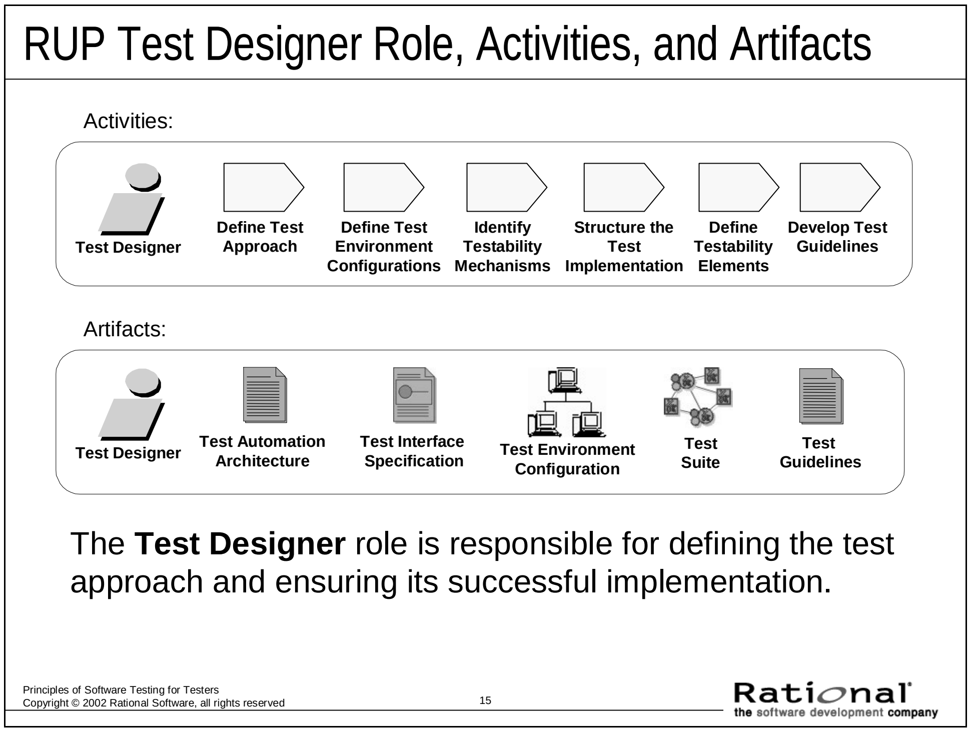
Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 14

The **Test Analyst** role is responsible for initially identifying and defining the required tests, and subsequently evaluating the results of the test effort. This involves monitoring the test coverage and evaluating the perceived software quality experienced during testing. This role also involves specifying required Test Data.

Sometimes this role may be referred to as the Test Designer, or considered part of the Tester role.

The Test Analyst role is the primary role that this course focuses on.

## RUP Test Designer Role, Activities, and Artifacts



### 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.

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 15

Rational® the software development company

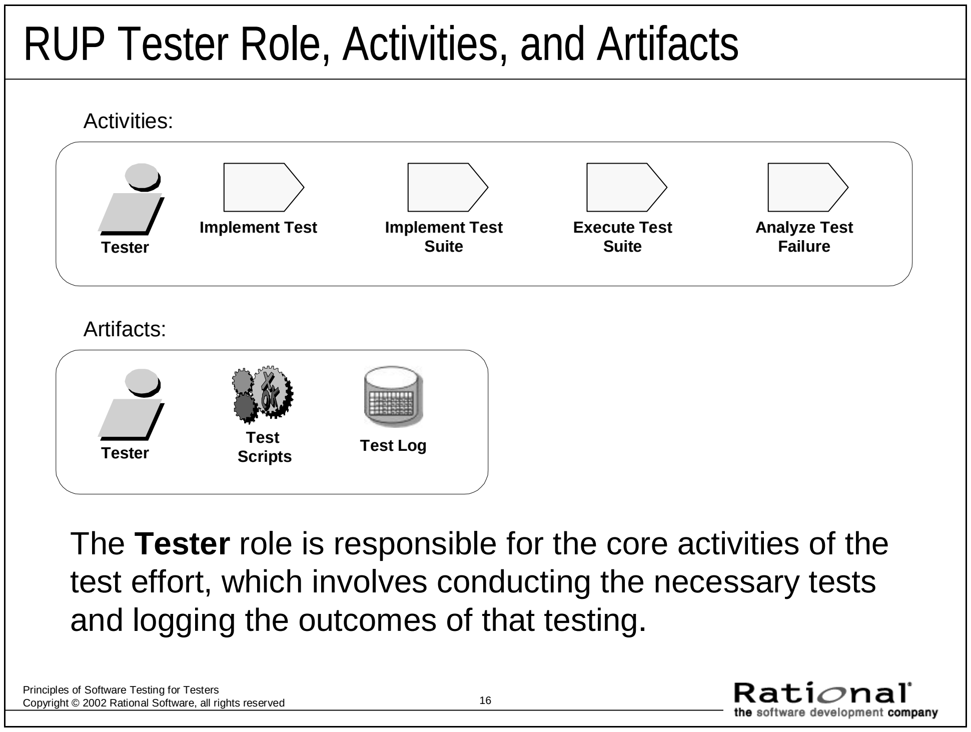
The **Test Designer** role is responsible for defining the test approach and ensuring its successful implementation. The role involves identifying the appropriate techniques, tools and guidelines to implement the required tests, and to give guidance on the corresponding resources requirements for the test effort.

Sometimes this role is referred to as the Test Architect, Test Automation Architect or Test Automation Specialist.

Where test automation is being conducted, the Test Designer role plays an important part in the work required to successfully achieve automation.

The Test Designer role is a secondary role that this course focuses on.

## RUP Tester Role, Activities, and Artifacts



### 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.

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 16

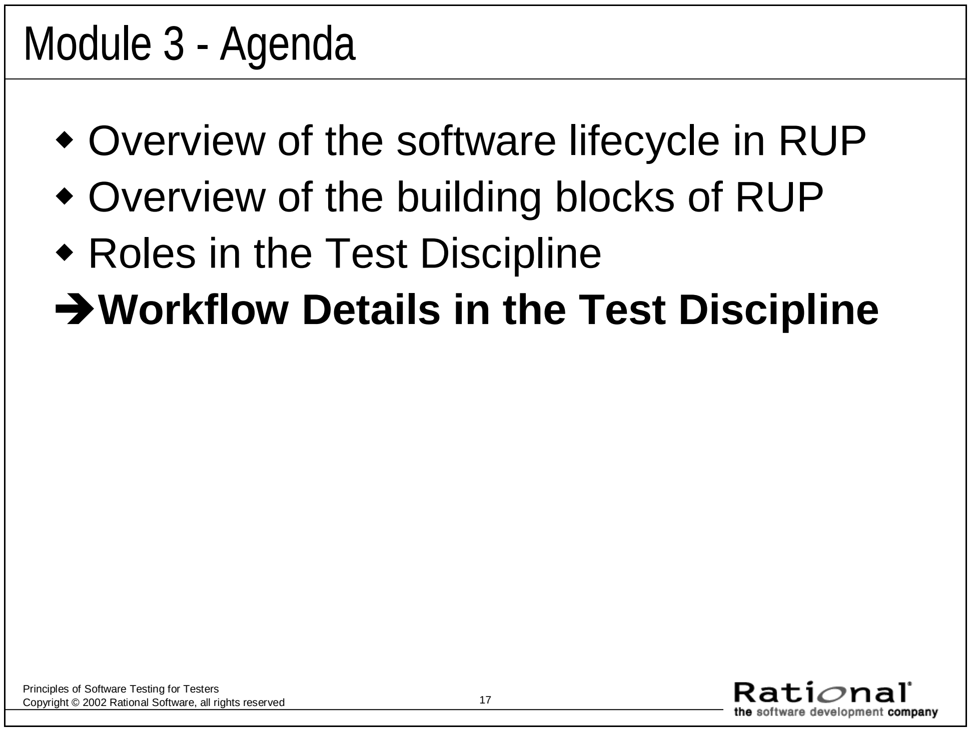
Rational the software development company

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.

Where test automation is being conducted, the Tester role plays a large part in the work required to successfully achieve automation.

The Tester role is a secondary role that this course focuses on.

## Workflow Details in the Test Discipline



### Module 3 - Agenda

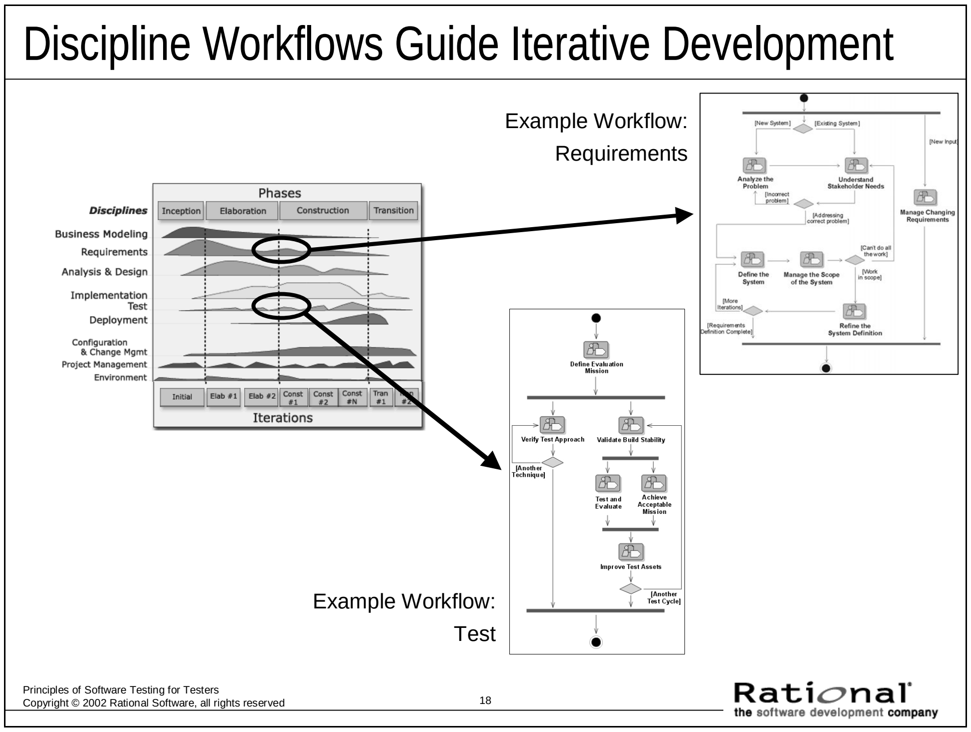
- Overview of the software lifecycle in RUP
- Overview of the building blocks of RUP
- Roles in the Test Discipline

➔**Workflow Details in the Test Discipline**

Principles of Software Testing for Testers
17

In this section, we introduce collections of Roles, Activities and Artifacts that collaborate to achieve meaningful goals.

## Discipline Workflows Guide Iterative Development



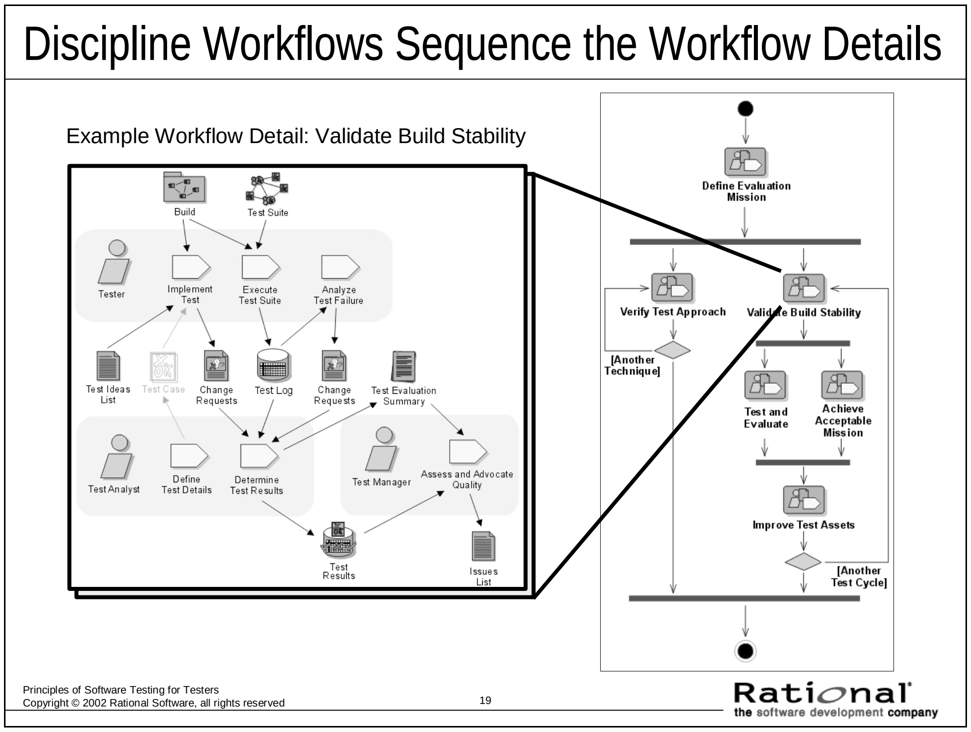
### Discipline Workflows Guide Iterative Development

18

Rational
the software development company

Within a discipline, workflows sequence groups of activities that are done together into workflow details. The Discipline workflows will be enacted to various levels of completeness, largely dependent on the lifecycle phase of the current iteration.

## Discipline Workflows Sequence the Workflow Details



Discipline Workflows Sequence the Workflow Details

Example Workflow Detail: Validate Build Stability

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 19

Rational® the software development company

This graphic illustrates the high-level view of the Test Discipline Workflow. Each workflow detail on the workflow flowchart contains finer-grained process guidance, as shown in the example workflow detail diagram.

The workflow detail diagram shows a set of activities that are often performed together to achieve a useful strategic objective. It also shows input and output artifacts (indicated with arrows), as well as the roles involved in conducting the work.

We'll briefly define the workflows in the next few slides, and will use them to structure the discussion in much of the rest of the class.

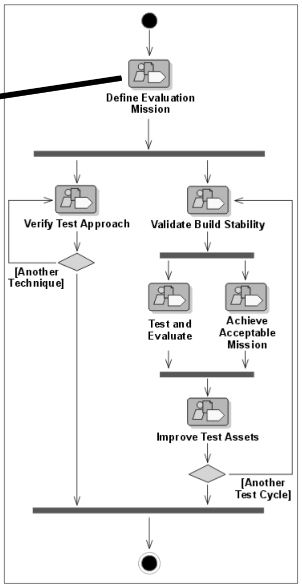
Note that within an iterative development lifecycle, each of these areas of activity will typically be addressed to some degree within each iteration.

Sometimes an area of activity will not require any work, in which case you will simply review it and verify that no work is required. In other situations, you may decide that entire areas of activity are not relevant in the context of the current iteration.

Note: The remaining slides in this module are designed to give you some context for the content covered in the subsequent modules of this course. We will look at the work involved in each of these workflow details in much more detail during each subsequent module. For now, we just want to expose you to the range of work being done in an iteration of testing.

## The RUP Test Discipline Workflow

### 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 purpose of this workflow detail is to:

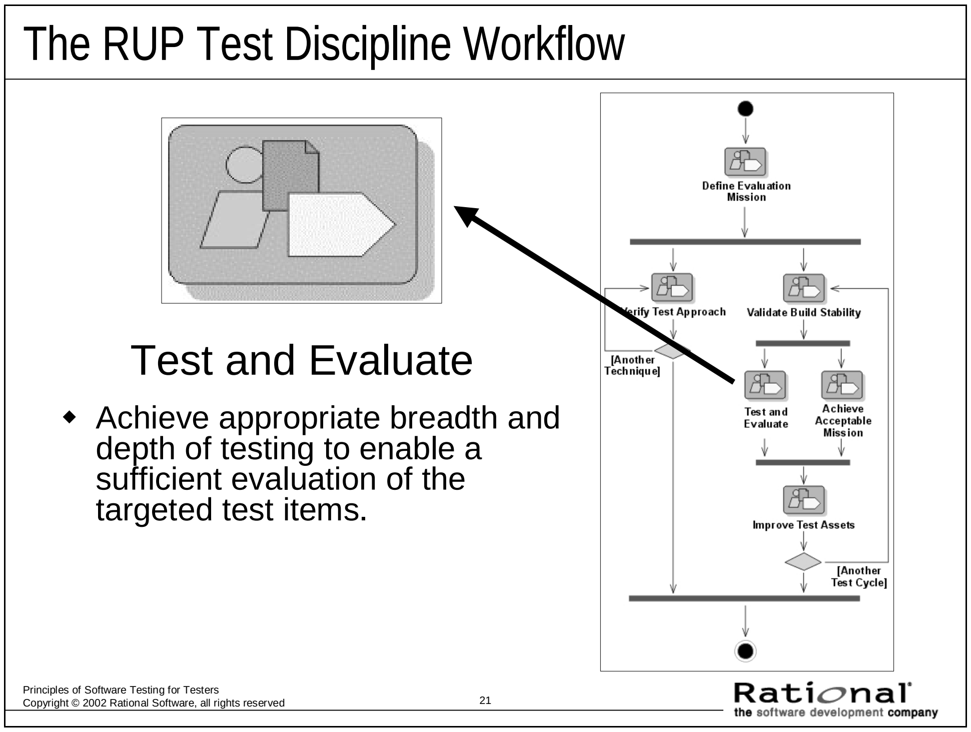
Identify the appropriate focus of the test effort for the iteration.

Gain agreement with stakeholders on the corresponding goals that will direct the test effort.

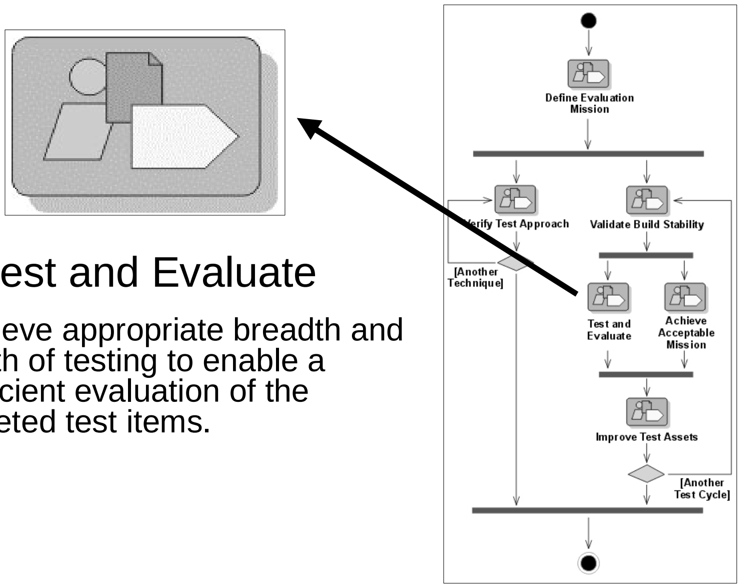
For each iteration, work is focused mainly on:

- Identifying the objectives for, and deliverables of, the testing effort
- Identifying a good resource utilization strategy
- Defining the appropriate scope and boundary for the test effort
- Outlining the approach that will be used
- Defining how progress will be monitored and assessed.

## The RUP Test Discipline Workflow



### 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.

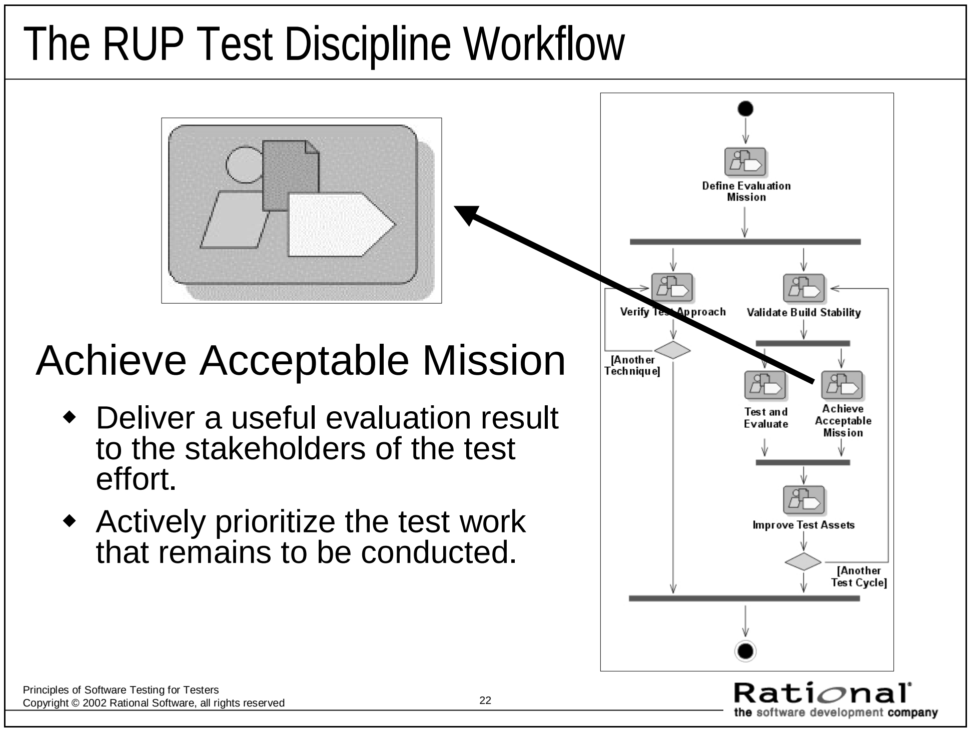
The purpose of this workflow detail is to:

Achieve appropriate breadth and depth of the test effort to enable a sufficient evaluation of the Target Test Items—where sufficient evaluation is governed by the Test Motivators and Evaluation Mission. Typically performed once per test cycle, this work involves performing the core tactical work of the test and evaluation effort: namely the implementation, execution and evaluation of specific tests and the corresponding reporting of incidents that are encountered.

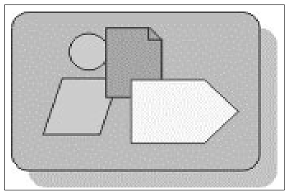
For each test cycle, this work is focused mainly on:

- Providing ongoing evaluation and assessment of the Target Test Items
- Recording the appropriate information necessary to diagnose and resolve any identified Issues
- Achieving suitable breadth and depth in the test and evaluation work
- Providing feedback on the most likely areas of potential quality risk

## The RUP Test Discipline Workflow

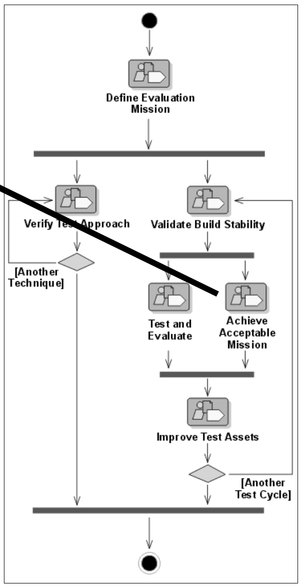


### The RUP Test Discipline Workflow



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.



Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved
22

Rational
the software development company

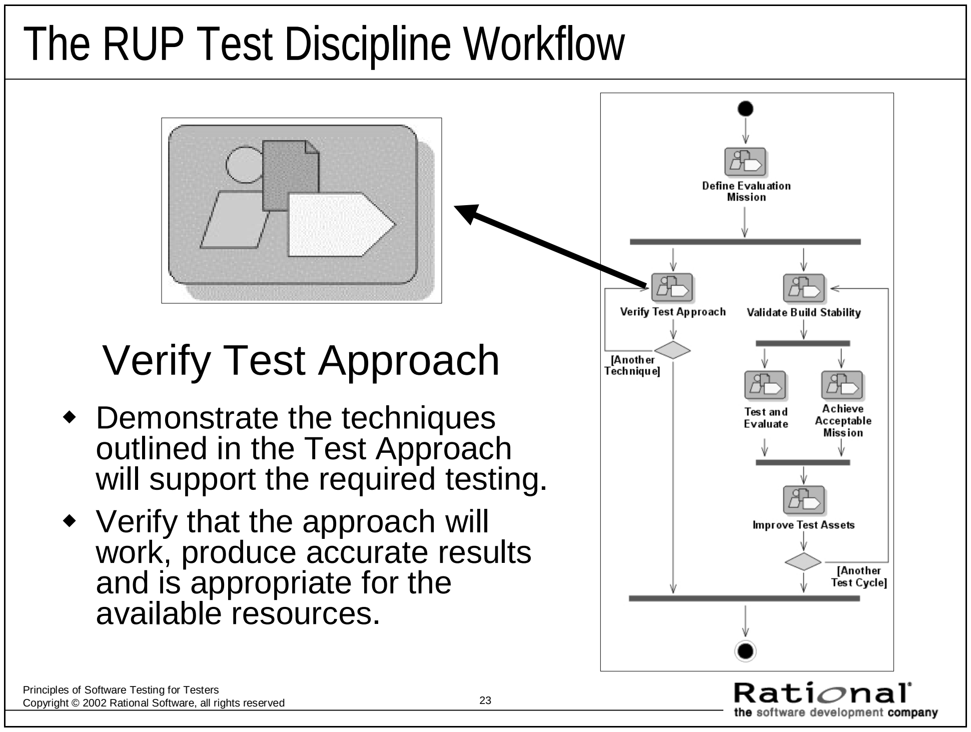
The purpose of this workflow detail is to:

Deliver a useful evaluation result to the stakeholders of the test effort—where useful evaluation result is assessed in terms of the Evaluation Mission. In most cases that will mean focusing your efforts on helping the project team achieve the Iteration Plan objectives that apply to the current test cycle.

For each test cycle, this work is focused mainly on:

- Actively prioritizing the minimal set of necessary tests that must be conducted to achieve the Evaluation Mission
- Advocating the resolution of important issues that have a significant negative impact on the Evaluation Mission
- Advocating appropriate quality
- Identifying regressions in quality introduced between test cycles
- Where appropriate, revising the Evaluation Mission in light of the evaluation findings so as to provide useful evaluation information to the project team

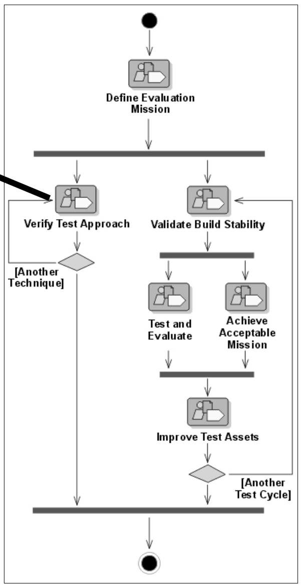
## The RUP Test Discipline Workflow



### 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.



Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved 23

Rational the software development company

The purpose of this workflow detail is to:

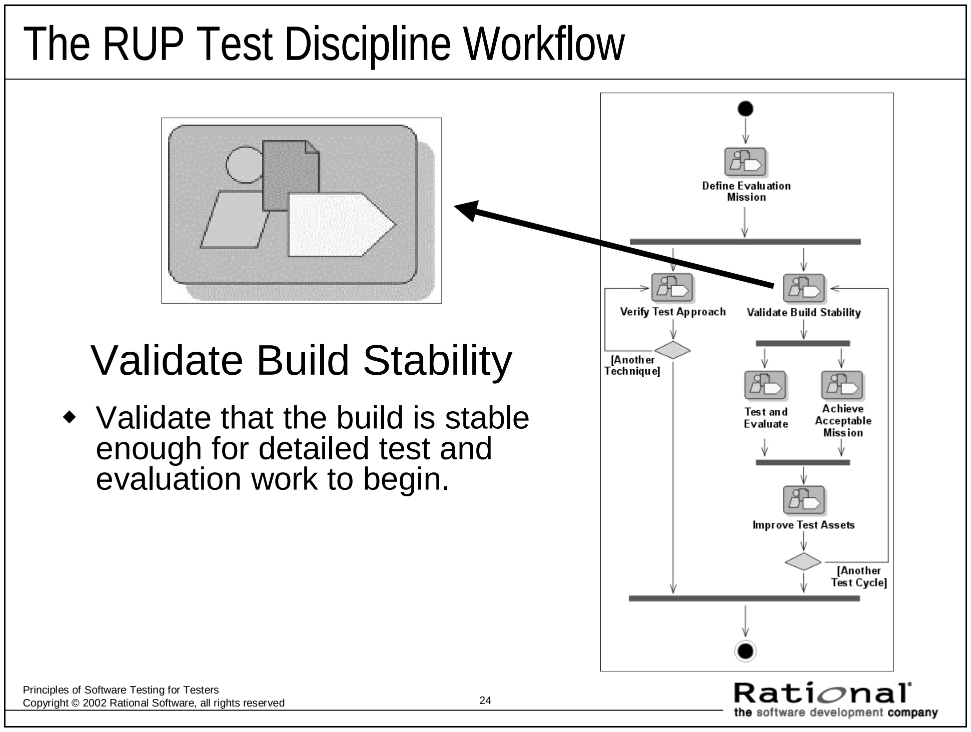
Show that the various techniques outlined in the Test Approach will facilitate the required testing. Verify by demonstration that the approach works, will produce accurate results and is appropriate for the available resources.

The objective is to gain an understanding of the constraints and limitations of each technique, and to either find an appropriate implementation solution for each technique or find alternative techniques that can be implemented. This helps to mitigate the risk of discovering too late in the project life-cycle that the test approach is unworkable.

For each iteration, this work is focused mainly on:

- Early verification that the intended Test Approach will work and produces results of value
- Establishing the basic infrastructure to enable and support the Test Approach
- Obtaining commitment from the development team to provide and support the required testability to achieve the Test Approach
- Identifying the scope, boundaries, limitations and constraints of each technique

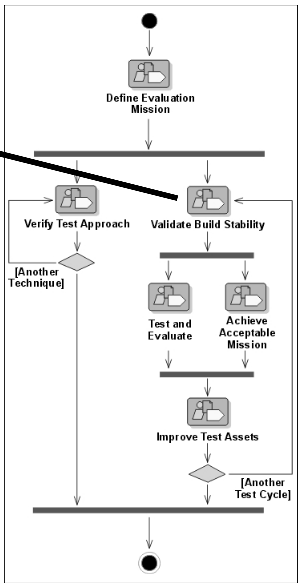
## The RUP Test Discipline Workflow



### The RUP Test Discipline Workflow

Validate Build Stability

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



Principles of Software Testing for Testers

24

Rational
the software development company

The purpose of this workflow detail is to:

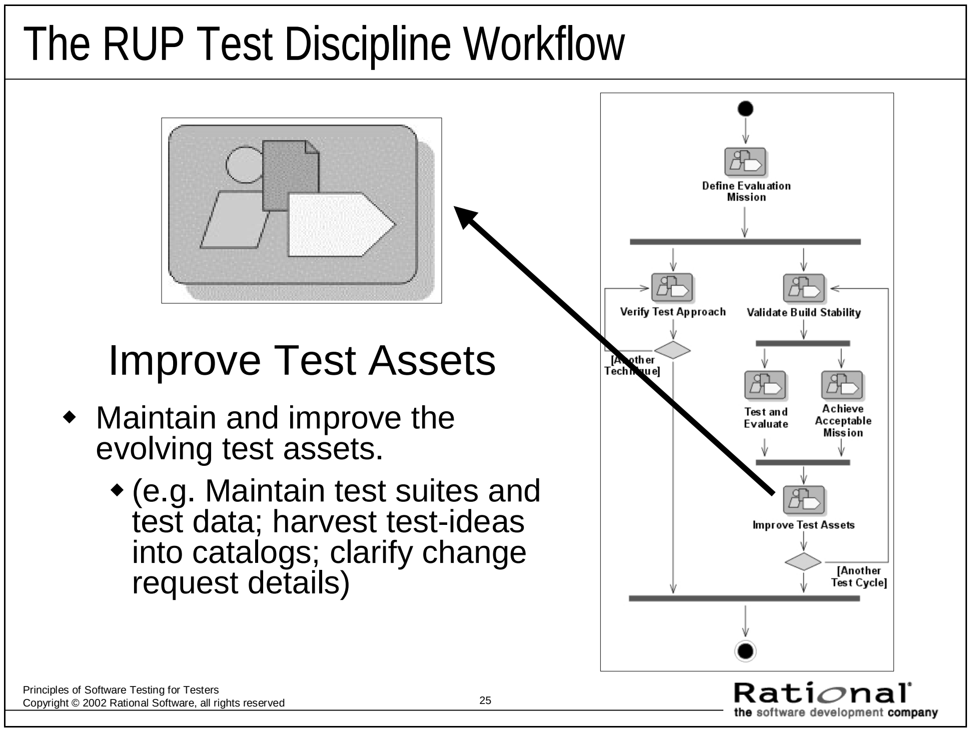
Validate that the build is stable enough for detailed test and evaluation effort to begin. This work helps to prevent the waste of test resources on a futile testing effort.

For each build to be tested, this work is focused on:

- Assessing the stability and testability of the build
- Gaining an initial understanding—or confirming the expectation—of the development work delivered in the build
- Deciding to accept the build as suitable for use—guided by the evaluation mission—in further testing, or to conduct further testing against a previous build.

This work is also referred to as a *smoke test, build verification test, build regression test, sanity check or acceptance into testing*.

## The RUP Test Discipline Workflow



### 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)

Define Evaluation Mission
Verify Test Approach
Validate Build Stability
[Another Technique]
Test and Evaluate
Achieve Acceptable Mission
Improve Test Assets
[Another Test Cycle]

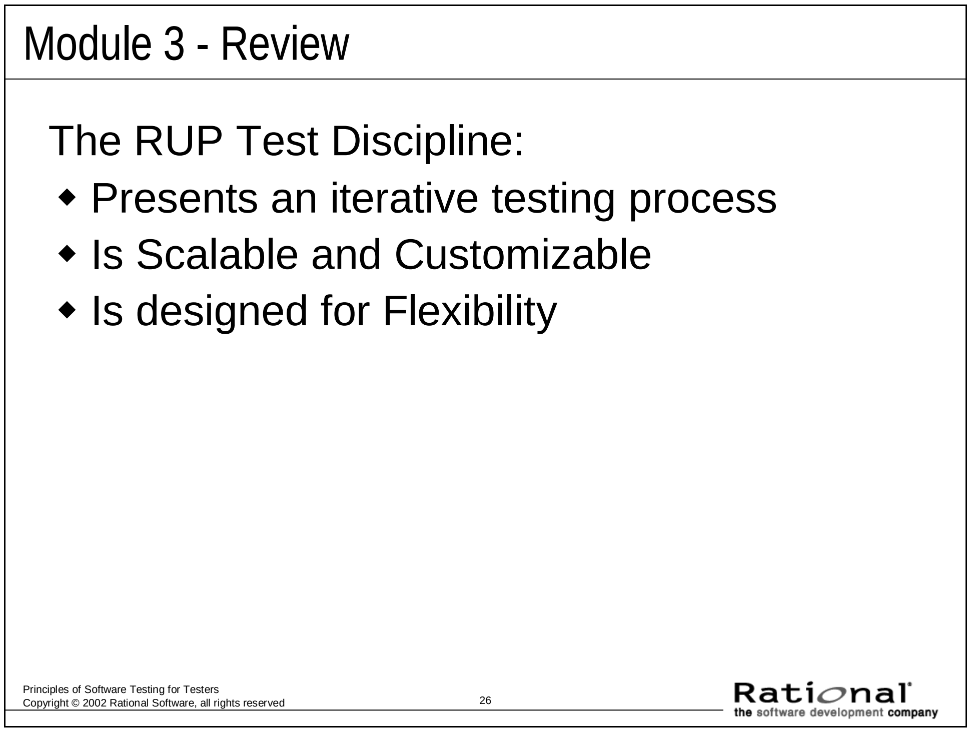
The purpose of this workflow detail is to:

Maintain and improve the test assets. This is important especially if the intention is to reuse the assets developed in the current test cycle in subsequent test cycles.

For each test cycle, this work is focused mainly on:

- Adding the minimal set of additional tests to validate the stability of subsequent Builds
- Assembling Test Scripts into additional appropriate Test Suites
- Removing test assets that no longer serve a useful purpose or have become uneconomic to maintain
- Maintaining Test Environment Configurations and Test Data sets
- Exploring opportunities for reuse and productivity improvements
- Conducting general maintenance of and making improvements to the maintainability of test automation assets
- Documenting lessons learned—both good and bad practices discovered during the test cycle.

## Module 3 - Review



Module 3 - Review

The RUP Test Discipline:

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

The RUP Test Discipline presents a process and set of recommended practices to guide your testing effort. Unlike traditional waterfall testing process, RUP advocates an iterative approach to testing. Iterative testing allows you to:

- Mitigate high risks earlier in the development process
- Focus your resources when and where they can have the most impact
- Maximize your effectiveness by adapting your approach, process, or assets as you go

The RUP Test Discipline is designed to be flexible and adaptable. Its process framework can support different sized organizations, and its recommended approach can be adapted to formal or informal testing styles within an organization.

A key focus of the Test Discipline is on maximizing the effectiveness of an organization’s testing efforts.