

## Distributed Rational Unified Process in an On-Demand Geographically Distributed Development Environment

---

IBM's Rational software development tools were built to be deployed on desktop LAN-based environments, they are weakly-integrated with each other, and are not integrated with modern collaboration tools nor do they have a common web interface that provides a complete end-to-end visibility or tracking of the true state of the project. As a result, Rational tools just don't scale to support geographically dispersed software development teams over the internet.

However, the Rational Unified Process, or RUP, is tool independent. As a result, when your design challenges scale to include distributed software development, you can protect your people, process, and training investment in your RUP methodology by applying it within CollabNet's development environment which integrates a set of software application lifecycle development capabilities designed specifically for the web.

This white paper describes the structure of CollabNet's Distributed RUP, the template that has been codified directly into CollabNet Application Lifecycle Manager and the value that CollabNet Distributed RUP provides within an iterative and distributed software process.

---

## Table of Contents

Overview .....	3
What is the IBM Rational Unified Process? .....	3
What is Distributed RUP? .....	4
Why consider CollabNet Distributed RUP?.....	4
Distributed RUP Lifecycle Phases .....	5
Distributed RUP Implementation.....	10
Conclusion .....	14
For More Information.....	15

## Overview

Do more with less! Reduce Software Development Release cycles and improve quality of software (speed and quality - the ultimate paradox!). Provide an easier way for business users and subject matter experts to add new functionality and follow it through the development life cycle tracing it back to ensure that the deployed assets map to the requirements and use cases that were originally submitted.

As software is built around the clock in various parts of the world, the need for distributed software development teams to collaborate, securely integrate their internal teams and external partners, have a repeatable process and reuse existing code and software assets is greater than before. To realize the full benefits of distributed development while addressing the related challenges and risks, project teams should re-assess the way they work today. Every development team should evaluate, both from a technical and economical point of view, whether the processes and tools that they use today are adequate to support a globally distributed development model.

IBM's Rational software development tools were built to be deployed on desktop LAN based environments, and are not integrated with modern collaboration tools. As a result, Rational tools just don't scale to support geographically dispersed software development teams that need to collaborate over the internet. However, the Rational Unified Process, or RUP, is tool independent. As you expand your development resources to include remote teams, you can protect your investment in your RUP methodology by applying it within the CollabNet development environment – a set of software application lifecycle development capabilities designed specifically for the web.

This paper discusses how CollabNet addresses the challenges of iterative and distributed software development. It describes the Distributed Rational Unified Process (Distributed RUP), which is a distributed iterative software development process based on the Rational Unified Process (RUP) using CollabNet Enterprise Edition. This paper also explains why Distributed RUP ought to be considered and the value it provides when implementing an iterative software process. It further details the Distributed RUP lifecycle stages as mapped inside CollabNet Enterprise Edition and CollabNet Application Lifecycle Manager and how activities and artifacts can be easily tracked from the inception phase through the transition phase.

### What is the IBM Rational Unified Process?

RUP is a collection of software disciplines which provide a framework to support an iterative, use case driven, architecture centric approach as illustrated in figure 1.0.

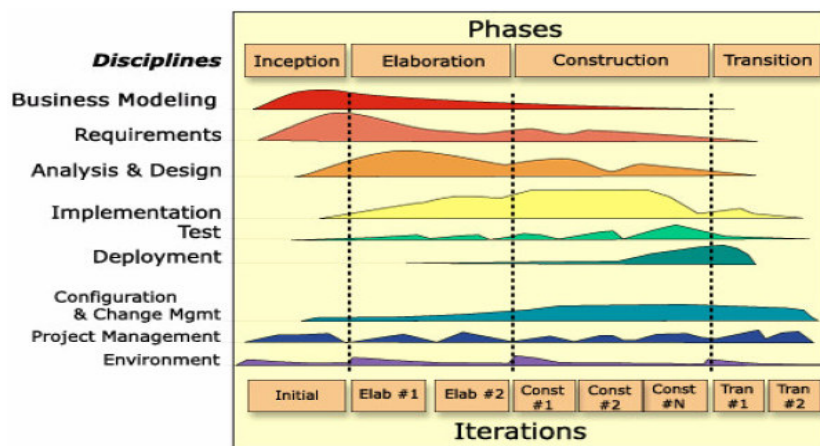


Figure 1: RUP Phases

When using RUP, software product lifecycles are broken into individual development cycles. These cycles are further broken down into phases. In RUP, these phases are well-defined:

- Inception Phase
- Elaboration Phase
- Construction Phase
- Transition Phase

Each phase is defined by a list of activities and objectives and can be further broken down by iterations. Iterations are time boxes that have specific deadlines.

## What is Distributed RUP?

Distributed RUP is a template mapped inside CollabNet Enterprise Edition that can be used to support and codify development best practices. This unique template provides a software development lifecycle process based on a subset of the IBM Rational Unified Process (RUP) that can be shared across geographically dispersed software development teams. It allows for real-time visibility and traceability of software artifacts without the need for replication. Distributed RUP also manages iterations and enforces process rules. Distributed RUP along with CollabNet Enterprise Edition provides collaboration tools to track (who, why and what) project plans and provides visibility into the project status throughout the development lifecycle. Distributed RUP also extends Application Lifecycle Management (ALM) concepts with a library of reusable software assets and templates.

## Why consider CollabNet Distributed RUP?

CollabNet Enterprise Edition is an integrated web environment architected to support distributed development. Figure 1.1 illustrates this solutions architecture

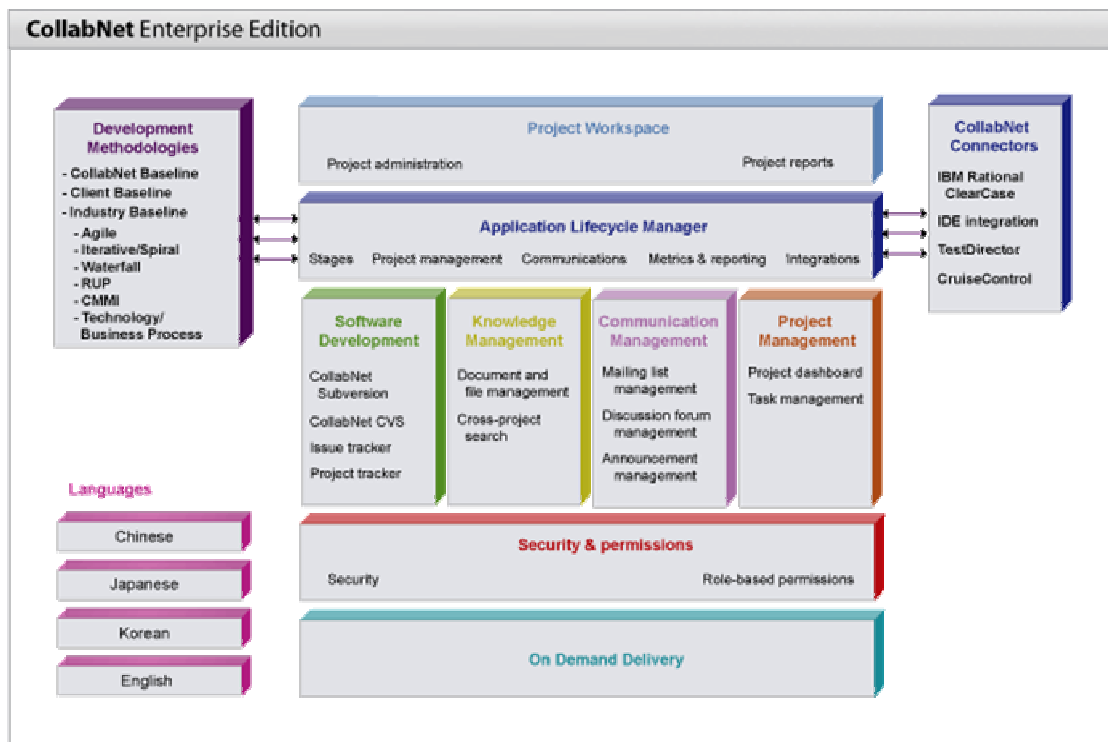


Figure 1.1 CollabNet Architecture

As illustrated above in the CollabNet architecture, CollabNet's support for Distributed RUP provides the essential components needed to implement most popular methodologies and process models in a "distributed" way. CollabNet provides the following capabilities:

- Software application development tools that are 100% web-based and integrated with modern collaboration tools,
- A process independent toolset which can be codified to adapt your RUP process to distributed development,
- A secure, permission-based portal that provides configurable, internet-based access to anyone -- no matter where (or in what time zone) they reside,
- A central repository for storing and collaborating around the generation of deliverables -- be they source code, documents, planning assets, or other artifacts,
- A collaborative environment to communicate vital information and working details to extended project participants,
- Tasking and tracking of individual work items, without the need to constantly pester people over the phone and/or through email,
- Content customization necessary to communicate and implement consistent methodologies/processes across the enterprise.

### **In Contrast**

- The Rational architecture won't scale over the web to support distributed development - it was designed for desktop environments.
- The Rational family of products relies on disparate databases that do not share a common web interface and therefore cannot provide end-to-end visibility and proper tracking of projects.
- Using Rational tools for distributed development is leading to costly slippages in software development cycles as well as very expensive deployment and operational investments.

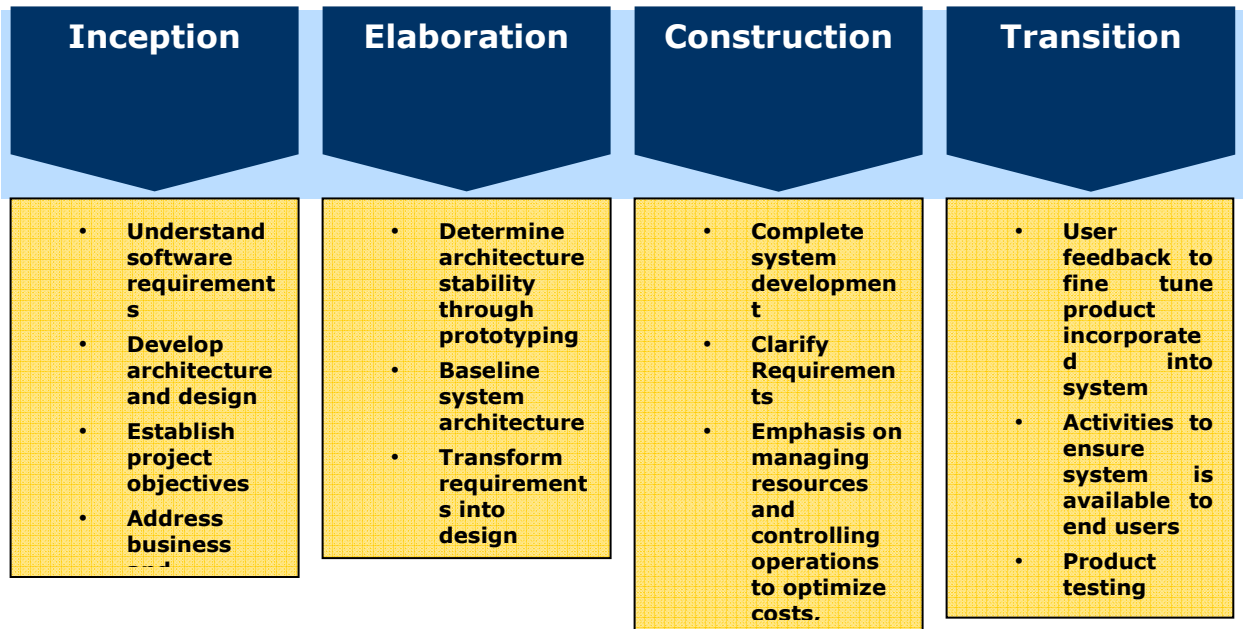
Organizations that use RUP often experience difficulties extending their processes into a distributed development environment. CollabNet's support for Distributed RUP enables these organizations to map portions of their RUP methodology into a web-based software collaboration environment in order to efficiently collaborate with developers and external partners regardless of their locations. With Distributed RUP, project managers and software teams can:

- Provide process visibility and real-time awareness to all stakeholders regardless of their geographical location,
- Reduce software development cycles by facilitating parallel and iterative development,
- Identify defects earlier in the development cycle,
- Make it easy for business users to submit requirements and follow them from design to deployment,
- Maintain traceability between requirements, defects, code and test plans,
- Provide a central location for everyone (Project Managers, Developers, Testers) to access project information,
- Provide roles with permissions so only users with specific set of permissions can view and or modify Intellectual Property (IP),
- Assign tasks, and action items and track them,
- Centralize project management activities and project reporting.

### ***Distributed RUP Lifecycle Phases***

Figure 2 illustrates four phases of the Distributed RUP lifecycle phases, highlighting tasks and artifacts created within each phase.

**Phases**



**Artifacts**

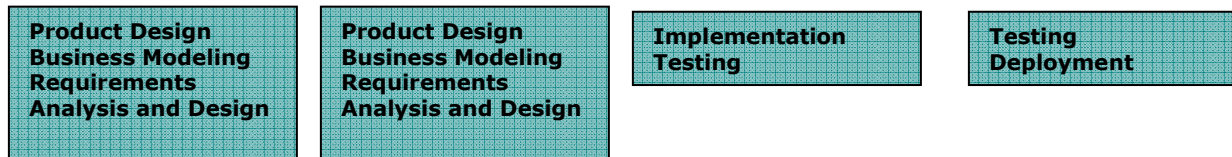


Figure 2.0: Distributed RUP Phases

**Inception Phase**

The main goal of the inception phase is to achieve consensus among all stakeholders on the objectives of the project. The inception phase is critical for new development efforts in which significant business and requirements risks need to be addressed before the project can proceed. For project focused on enhancing existing systems, the inception phase is brief but is still focused on ensuring the feasibility of the project.

The goals of the inception phase are:

- To describe the initial requirements,
- To develop and justify the business case for the project,
- To determine the scope of the project,
- To identify resources, organizations, and external systems that will interact with the project,
- To develop an initial risk assessment, schedule, and estimate for the project,
- To develop an initial tailoring of the process to meet exact needs.

Figure 3 illustrates the workflow within the inception phase. The entry criteria for this phase are the approval of the project concept. Work is done in the inception phase to create the initial iterations of the business models, requirements, and design documents. These artifacts are tracked from inception, and once these artifacts are approved based on established goals, the inception phase is considered complete. The approved artifacts can now move on to the elaboration phase.

### Inception Phase: Process Workflow

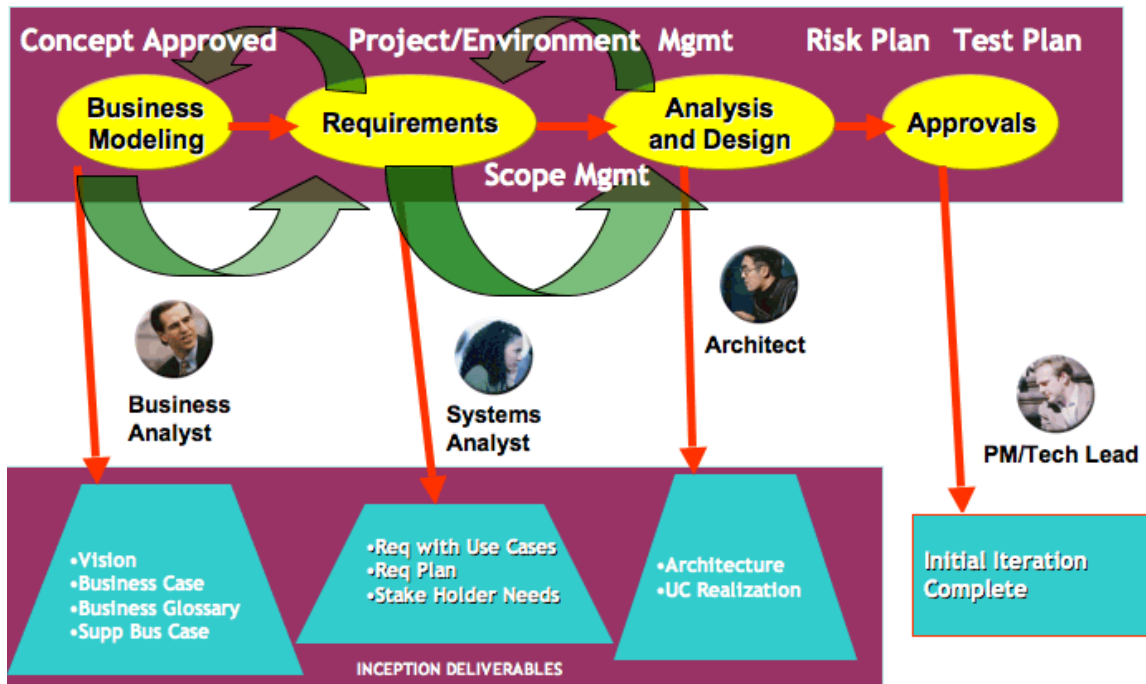


Figure 3.0: Inception Phase

### Elaboration Phase

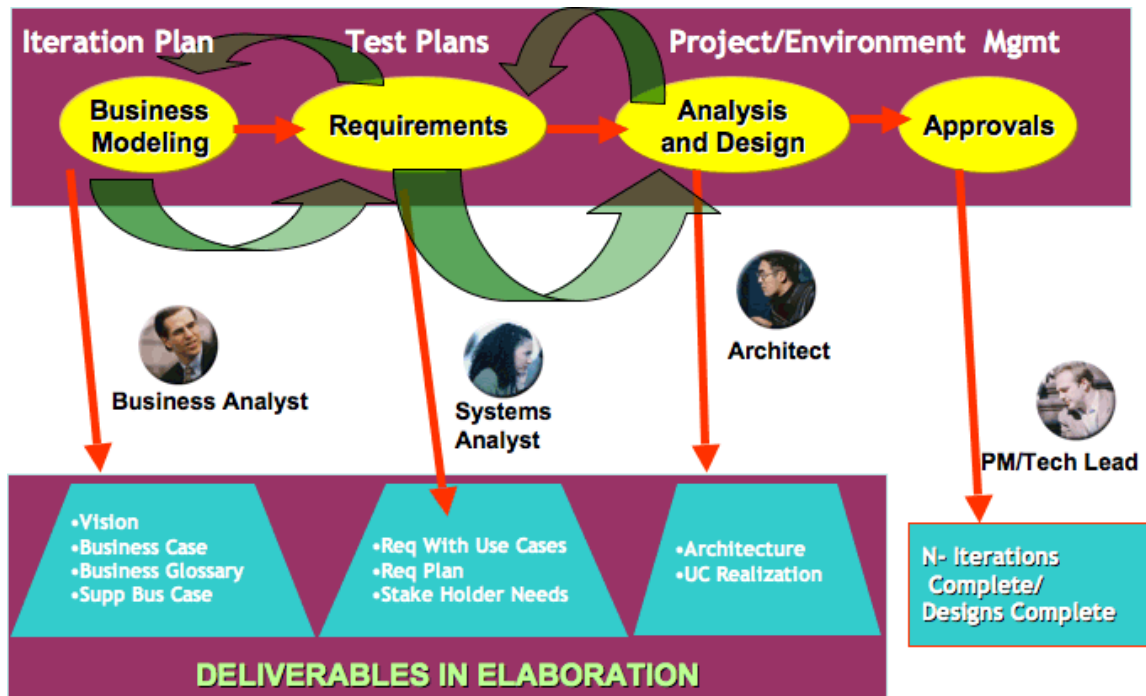
The goal of the Elaboration phase is to baseline the architecture of the project to provide a basis for the bulk of the design and implementation efforts in the construction phase. The architecture evolves out of a consideration of the most significant requirements (those that have a great impact on the architecture) and an assessment of risk. The stability of the architecture is evaluated through one or more architectural prototypes.

The goals of the elaboration phase are:

- To produce a proven, architectural baseline for your system,
- To evolve your requirements model to the "80% completion point",
- To develop a coarse-grained project plan for the entire Construction phase,
- To ensure that the critical tools, processes, standards, and guidelines have been put in place for the Construction phase,
- To understand and eliminate high-priority risks for your project.

Figure 4 illustrates the workflow within the elaboration phase. The entry criteria for this phase are the approval of the initial iterations of the business models, requirements, and design. Work is done in the elaboration phase to create additional iterations of these artifacts in order to evolve and fine tune the architecture of the system. These artifacts are tracked based on established goals and once approved, the phase can be considered complete. The approved artifacts can now move on to the construction phase.



**Elaboration Phase: Process (Detail) Workflow***Figure 4.0: Elaboration Phase***Construction Phase**

The goal of the construction phase is to clarify the remaining requirements and complete the development of the project based upon the baseline architecture. The construction phase is in some sense a manufacturing process, where emphasis is placed on managing resources and controlling operations to optimize costs, schedules, and quality. In this sense the management mindset undergoes a transition from the development of intellectual property during inception and elaboration, to the development of deployable products during construction and transition.

The goals of the construction phase are:

- To describe the remaining requirements,
- To flesh out the design of your project,
- To ensure that your project meets the needs of its users and fits into your organization's overall system portfolio,
- To complete component development and testing, including both the software product and its documentation,
- To minimize development costs by optimizing resources,
- To achieve adequate quality as rapidly as possible,
- To develop useful versions of your system.

Figure 5 illustrates the workflow within the construction phase. The fundamental goals of these workflows is to ensure that the system is being developed and tested according to the design documents developed and approved within the inception and elaboration phases. Once development and testing are completed, the project can move to the transition phase.



**Construction Phase: Process Workflow**

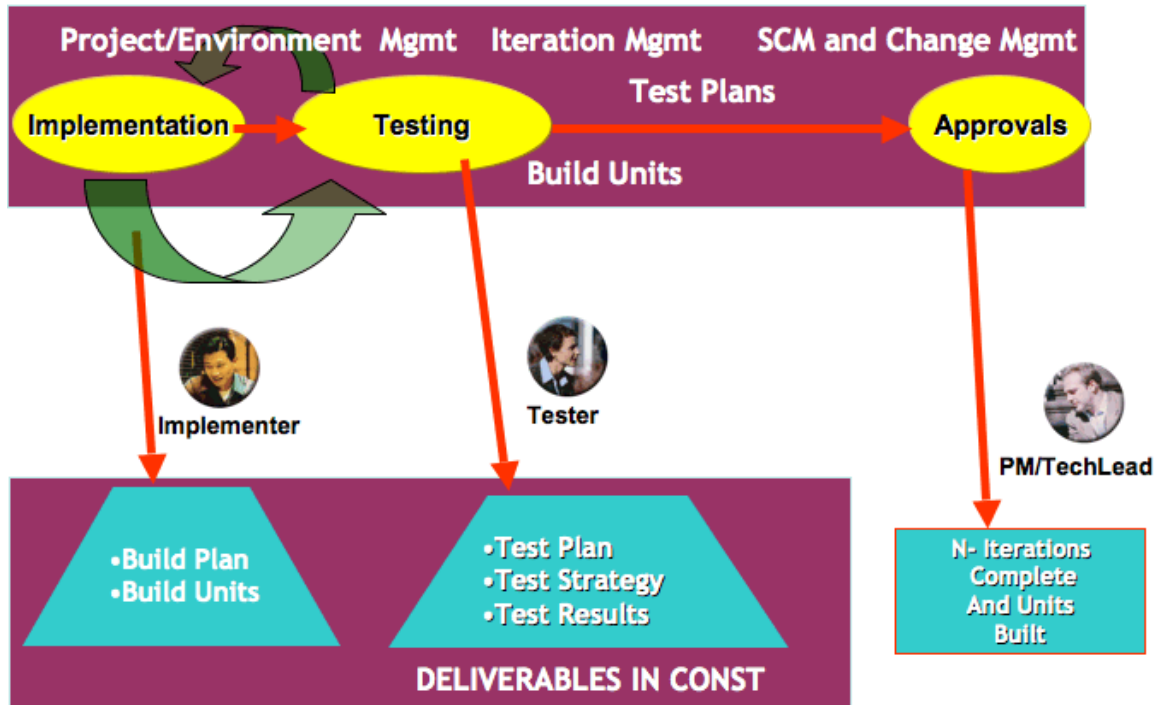


Figure 5.0: Construction Phase

**Transition Phase**

The Transition phase is the fourth phase that a system experiences throughout its complete lifecycle. Figure 6.0 illustrates the workflow in this phase. During the Transition phase the project team will focus on testing and validating the complete system. As these activities are performed, several artifacts will be created and evolved:

- Final product baseline (also known as a production baseline) of the system along with deployment units
- Bill of Materials, including user manuals, and other documentation such as release notes, deployment plans.

The phase is concluded with the Product Release milestone. To pass this milestone you must show that your users are satisfied with the system and that the actual expenditures versus the planned expenditures are still acceptable.

Figure 6 illustrates workflow details within this phase. The goal of these workflows is to ensure that the system is completely tested and deployed based on the requirements that have been approved.

**Transition Phase**

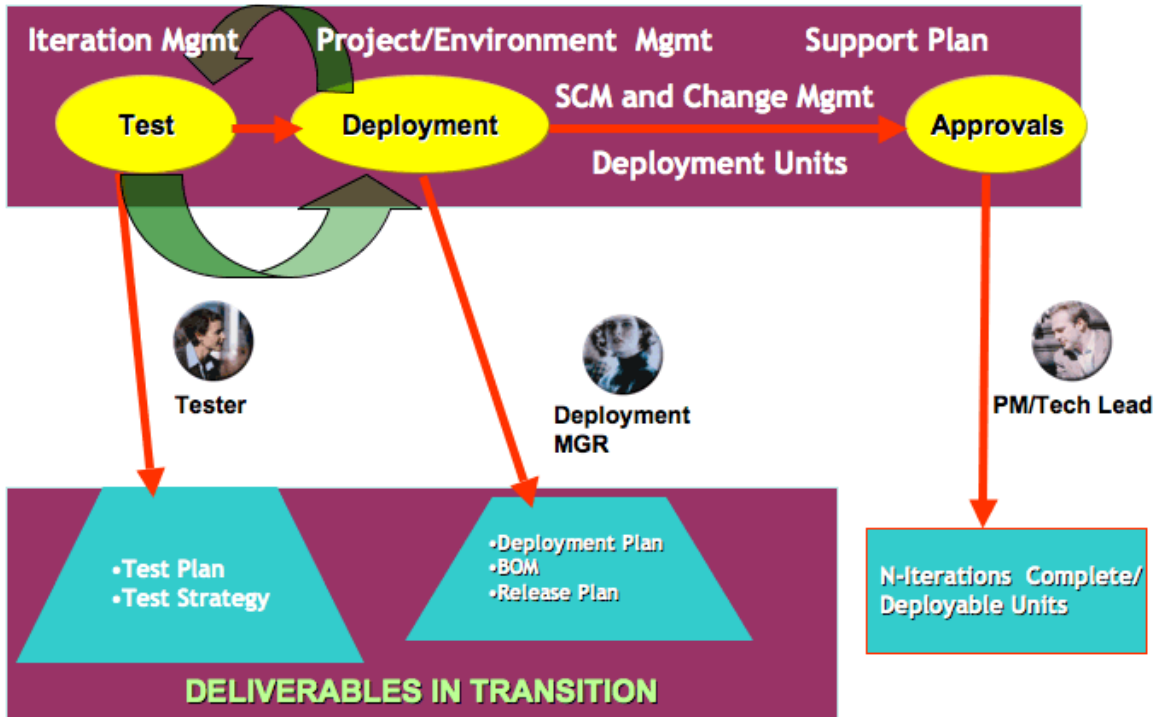


Figure 6.0: Transition Phase

**Distributed RUP Implementation**

The first screenshot below shows CollabNet Distributed RUP as a codified ALM template within CollabNet Enterprise Edition to enable RUP-oriented developers and development teams to successfully pursue distributed development.

Additional screenshots illustrate the following:

- Business Models in the Inception phase
- Overall portfolio status
- Backlog ECOs and Enhancements during the Inception phase

The screenshot shows a web browser window displaying the 'drupswdevel' project page. The browser's address bar shows 'https://drupswdevel.alm.collab.net/'. The page header includes 'COLLABNET Enterprise Edition' and a user login 'nick'. The main content area features a 'Project home' section with a summary: 'Summary DRUP SW Devel', 'Categories None', 'Owner(s) nick', and 'Your role(s) Project Owner, Developer'. Below this is a 'Stages' section with icons for 'Inception', 'Elaboration', 'Construction', and 'Transition'. A large diagram titled 'RUP DISCIPLINES WITH ITERATIONS' is displayed, showing various disciplines across four phases and multiple iterations.

**RUP DISCIPLINES WITH ITERATIONS:**

Disciplines	Phases			
	Inception	Elaboration	Construction	Transition
Business Modeling	Active	Active	Active	Active
Requirements	Active	Active	Active	Active
Analysis & Design	Active	Active	Active	Active
Implementation	Active	Active	Active	Active
Test	Active	Active	Active	Active
Deployment	Active	Active	Active	Active
Configuration & Change Mgmt	Active	Active	Active	Active
Project Management	Active	Active	Active	Active
Environment	Active	Active	Active	Active

Using RUP, software product lifecycles are broken into individual development cycles. These cycles are further broken into their main components, called phases. In RUP, these phases are termed as:

- Inception Phase**
- Elaboration Phase**
- Construction Phase**
- Transition Phase**

Phases are composed of iterations. Iterations are timeboxes; iterations have deadlines while phases have objectives.

*Distributed RUP as a codified Application Life Cycle template*

drupswdevel  
Project Tracker: Query results

Your changes were saved.

Save this query Refine query Export results to: Process results:

Query results (4 artifact(s) found)

2006-04-23 13:16:13 GMT

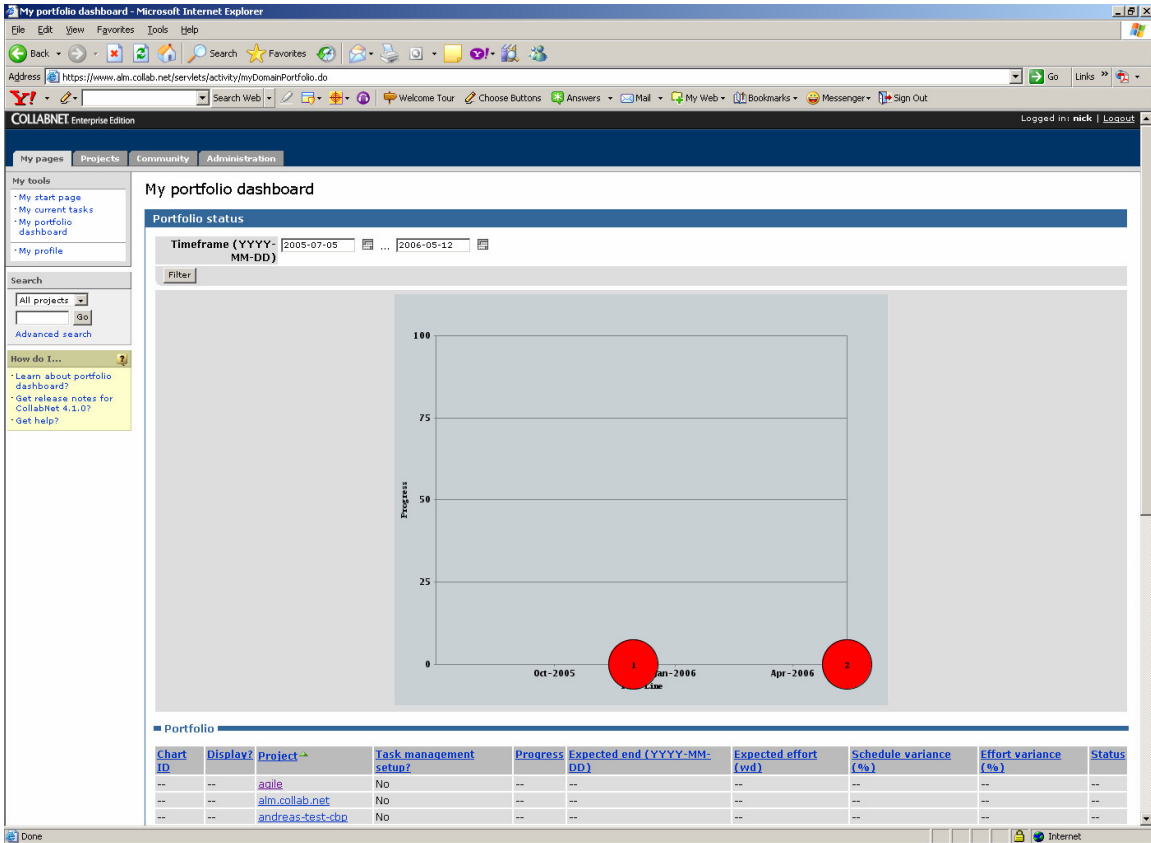
[Add/remove columns](#)

Select	Artifact ID	Last date/time modified	Age (days)	Date committed	A-D Iteration	A-D Milestone	Business Model Name	DRUP Phase
<input type="checkbox"/>	<a href="#">dr1</a>	2006-04-01 20:27:24 GMT	21	2006-04-01 20:14:37 GMT	-----	-----	RFID Model	Inception
<input type="checkbox"/>	<a href="#">dr2</a>	2006-04-23 13:11:41 GMT	0	2006-04-23 13:11:41 GMT	-----	-----	ATM Transaction	Inception
<input type="checkbox"/>	<a href="#">dr3</a>	2006-04-23 13:12:30 GMT	0	2006-04-23 13:12:30 GMT	-----	-----	Check Account Balance	Inception
<input type="checkbox"/>	<a href="#">dr4</a>	2006-04-23 13:15:02 GMT	0	2006-04-23 13:15:02 GMT	-----	-----	Transfer funds from checking to savings	Inception

Max items per page  25  50  75  100  All Refresh

Save this query Refine query Export results to: Process results:

***Business Models in Inception***



Overall Portfolio Status

**ECO Tracking Issues**

Feature	Status	Tracking Issue
CVS Upgrade	Approved	<a href="#">43707</a>
Admin of SSH Tunnel Keys	Approved	<a href="#">43730</a>
PT Multiple Assignments per Attribute	Approved	<a href="#">43985</a>
Simplified queries for PDQs, templated PDQs, DDQ edits result in project-scoped PDQs with the same XML name as the DDQ. Links to <a href="#">[PowerPoint]</a> and <a href="#">[MS-Project]</a> files used for ECO review.	Approved	<a href="#">44220</a>
Additional WS APIs to support Connectors - <a href="#">44442</a> , <a href="#">44376</a> and <a href="#">44375</a>	Approved	<a href="#">44949</a>

**Committed Backlog Issues (Defects and Enhancements)**

Defect	Component	Sponsor	Description
<a href="#">37887</a>	IZ	Support	'export this issue' button failing to go to this issue
<a href="#">14413</a>	Framework	Support	"email change" script needs to be servletized (was: Too easy to enter wrong hp system into HP scripts)
<a href="#">31767</a>	Framework	Support	Migration script fallback masks real errors behind fake errors
<a href="#">31531</a>	Picayne	PIC	Typo error in 'Delete Announcement' page
<a href="#">33084</a>	PT	Support	"Add/remove columns" is not working when different users have set different preferred columns for a project scoped query
<a href="#">34934</a>	PT	Support	Problem with sequence numbers of attribute that is used in "Issue Tracker artifact" (immutable)
<a href="#">37283</a>	IZ	Support	Issue tracker notification not generated for additional comment

Enhancement	Component	Sponsor	Description
<a href="#">43309</a>	Framework	UEX	Change admin function of "reporting" to "statistics"
<a href="#">36236</a>	Framework	Ops	set-permissions.py should run component-specific permission hook scripts
<a href="#">43263</a>	Project Tracker	Reporting	Cross Artifact Meta Data API
<a href="#">43147</a>	Profile (Ops)	PSR	Upgrade MySQL to at 4.0.26
<a href="#">32342</a>	Deployment	PIC	Remove the old migration system and temporary migration scripts

*Committed Backlog ECOs*

## Conclusion

CollabNet Distributed RUP provides a fast and accurate answer to "what has changed & why". With CollabNet and Distributed RUP, project stakeholders can be confident that there will be no surprises when moving the project from one phase of development to the next. With better defined and better manager artifacts, new projects can pass audits faster and more predictably.

The business trends of outsourcing, emerging markets, access to technology, mergers and acquisitions, increased reliance on partnerships and more rapid rates of innovation are all central pieces of the business environment in which we operate. Having a development process that is communicated and visible to all project stakeholders increases the quality of the end product and contributes to project being delivered on-time and on-budget.

CollabNet Distributed RUP overcomes some of the common criticisms of RUP which lacks tracking of activities with milestones, and the lack of role-based enforcements. It provides access and visibility to a process framework that uses software development best practices enabling:

- Project participants to obtain real time visibility to their tasks:
  - Developers know what to develop,
  - Testers know what to test,
  - Product Managers know what to define.
- Traceability between requirements, defects, code, and test plans,
- The handoff of artifacts - requirements, designs, code or test plans - amongst project stakeholders,
- Projects to maintain a central repository for everyone (project managers, developers and testers),
- Participants to address the system from a role and task based perspective,

- Centralization of project management,
  - Ties all work to milestones and schedules,
  - A milestone (point in time) can be tied to requirements due to be implemented and defects that need to be resolved,
  - Microsoft Project tasks that need to be completed.

Benefits that projects derive from these best practices are:

- Improved the overall quality of the product delivered,
- Increased resource utilization,
- Accelerated project startup and delivery.

## For More Information

To learn more about CollabNet solutions for distributed, collaborative software development, please call 1-888-778-9793, e-mail [info@collab.net](mailto:info@collab.net), or visit [www.collab.net](http://www.collab.net).

For the latest information about CollabNet, CollabNet White Papers, and Web Seminars, visit <http://www.collab.net/news/archives/>.