

Unified Software Project Management Rational ProjectConsole Sample Metrics

**Paul R. Wyrick,
Bill Gilbert and Dave West**

Rational Software White Paper



Rational®
the e-development company™

Table of Contents

Introduction	1
Measuring the Inception Phase	2
Trend of Level 0 and 1 Requirements	3
Trend of Level 0 and 1 Requirements with Type	4
Trend of Level 0 and 1 Requirements with Status	5
Distribution of Level 0 and 1 Requirements by Priority	6
Distribution of Level 0 and 1 Requirements by Iteration	7
Measuring the Elaboration Phase	8
Distribution of affects-architecture of Use Cases	9
Distribution of affects-architecture Use Cases by Status	10
Distribution of Requirements by Level	11
Trend of Level 0 and 1 Requirements with Status	12
Distribution of Level 0 and 1 Requirements by Iteration	13
Measuring the Construction Phase	14
Trend of Open Defects in Construction Iterations	15
Trend of Lines Added, Modified, and Deleted	16
Trend of Use Case Status	17
Measuring the Transition Phase	18
Trend of Total Defects found in this Release	19
Trend of Open Defects in this Release	20
Trend of Defects Status in this Release	21
Trend of Lines Added, Modified, and Deleted	22
Summary	23
References	24

Introduction

This document explains how progress and quality on a Rational Suite project can be measured using Rational ProjectConsole. This document has been written for ProjectConsole end-users to help them start their metrics programs.

When first starting to use Rational ProjectConsole, we strongly recommend that you start your metrics program by using a simple set of metrics. In this paper we describe an initial set of metrics we think that each project will find valuable. These metrics are available in the sample metrics database that is shipping with the product, except where noted. Within the product, you will also find a tutorial that describes how you can deploy these metrics into your production environment.

The metrics shown in the paper are not meant to be comprehensive, but are meant to be a small sample of the types of metrics that can be generated using Rational ProjectConsole. As you become more comfortable with Rational ProjectConsole, you can expand upon this set of metrics. Our experience shows that you typically are much better off starting small and then expanding, rather than trying to adopt a large number of metrics from the beginning.

There are more possible objects and attributes to measure than can be covered in this paper. Furthermore, each organization has distinct and different needs with regard to data. This paper will follow the measurement of a few types of information through the phases of the Rational Unified Process (RUP) development lifecycle. Specifically, measurements of features, use cases, and defects will be shown. That information will be extracted from Rational RequisitePro and Rational ClearQuest. Lines of code added, modified, and deleted will also be shown using information from Rational ClearCase. The measurements are aimed at trying to help the Project Manager address this question: *how much work remains to be done in each of the four RUP development phases: Inception, Elaboration, Construction and Transition?*

In this paper we will limit the charts to those of interest to a “stakeholder”. Key Project Stakeholders include Upper Level Management, Department Managers, Project Managers, Product Managers, Development Managers, Technical Managers, and Project Team Members. All of these types of stakeholders want an easy and efficient way to gather and disseminate project team status and metrics information. Key Project Stakeholders are interested in going to a centralized location to view current project status metrics. Project Team Members want an efficient way to report current status to their managers and a centralized location where they can be kept up-to-date on current project status and where they can access current project artifacts.

Stakeholders want to see the rate of newly identified work decline as the end of the project approaches. Based on where a project stands at any given point in time, there are different data points to see the status of the project. This paper guides you through what to look at during each of the phases of a RUP project and how to interpret the various types of information.

All project metrics examples shown within this paper were generated using Rational ProjectConsole, which is part of the Rational Suite. Rational ProjectConsole can gather information from any Rational Suite tool, from text files, from Microsoft Project and other 3rd party products. Therefore, to achieve the goal of having necessary information visible in ProjectConsole, the information must come from one of those source locations.

We will now proceed by introducing a few sample metrics that can be used to measure the progress and quality of each of the four RUP phases.

Measuring the Inception Phase

The trend charts in this section show measurement data used during the inception phase.

The inception phase of a RUP project is focused on ensuring that all project stakeholders have a clear understanding of the scope of the project. Business and system requirements are of paramount importance in this phase.

The primary objectives of the inception phase are:

- Establish the project's software scope and boundary conditions, including an operational vision, acceptance criteria and what is intended to be in the product and what is not.
- Discriminate the critical use cases of the system, the primary scenarios of operation that will drive the major design trade-offs.
- Exhibit, and maybe demonstrate, at least one candidate architecture against some of the primary scenarios
- Estimate the overall cost and schedule for the entire project (and more detailed estimates for the elaboration phase that will immediately follow)
- Estimate potential risks (the sources of unpredictability)
- Prepare the supporting environment for the project.

Because the emphasis of the inception phase is understanding requirements and determining risk, primary artifacts include:

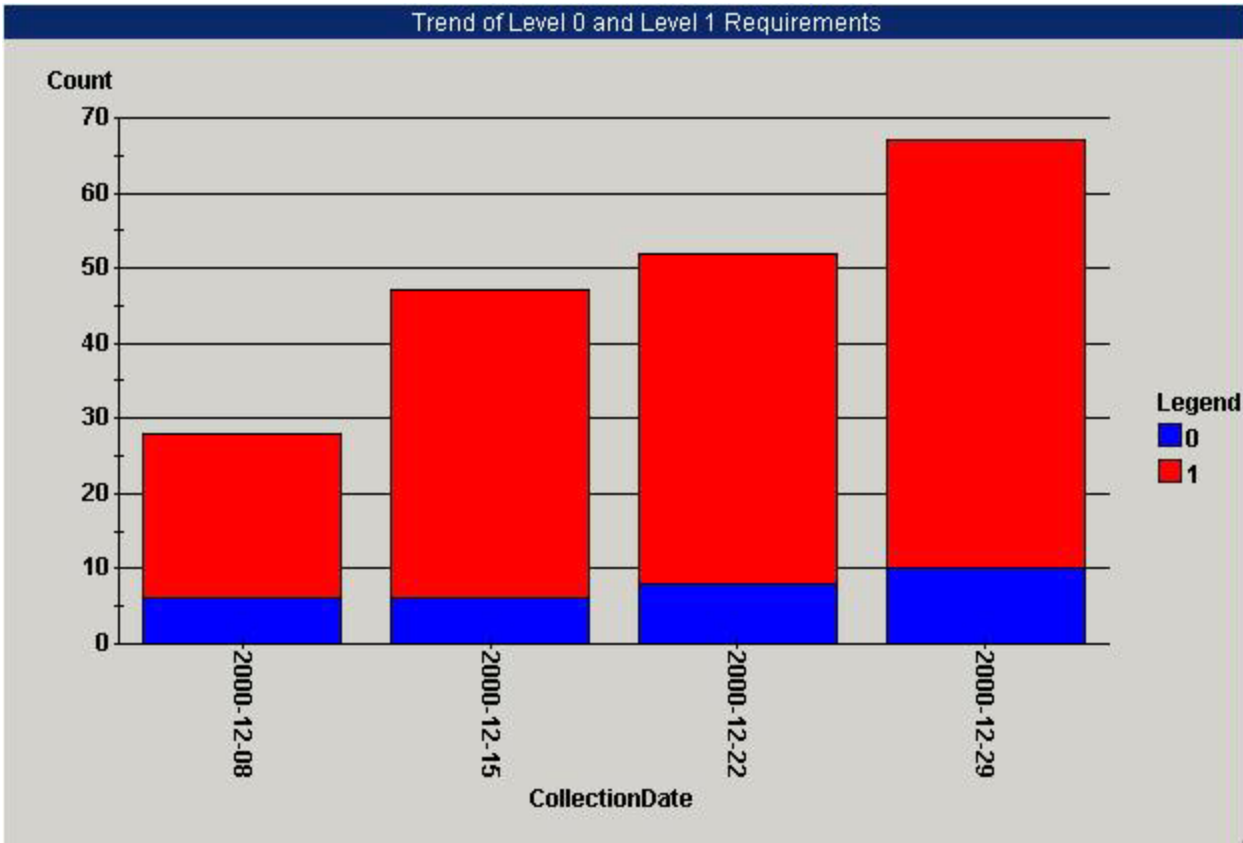
- Vision of the project
- Initial Software Development Plan
- Business Case
- Initial Use-Case Model
- Tailored Process Specification
- Risk list
- Risk Management Plan

Other optional artifacts may include:

- A Business Model
- A Domain Model
- An architectural proof of concept

Trend of Level 0 and 1 Requirements

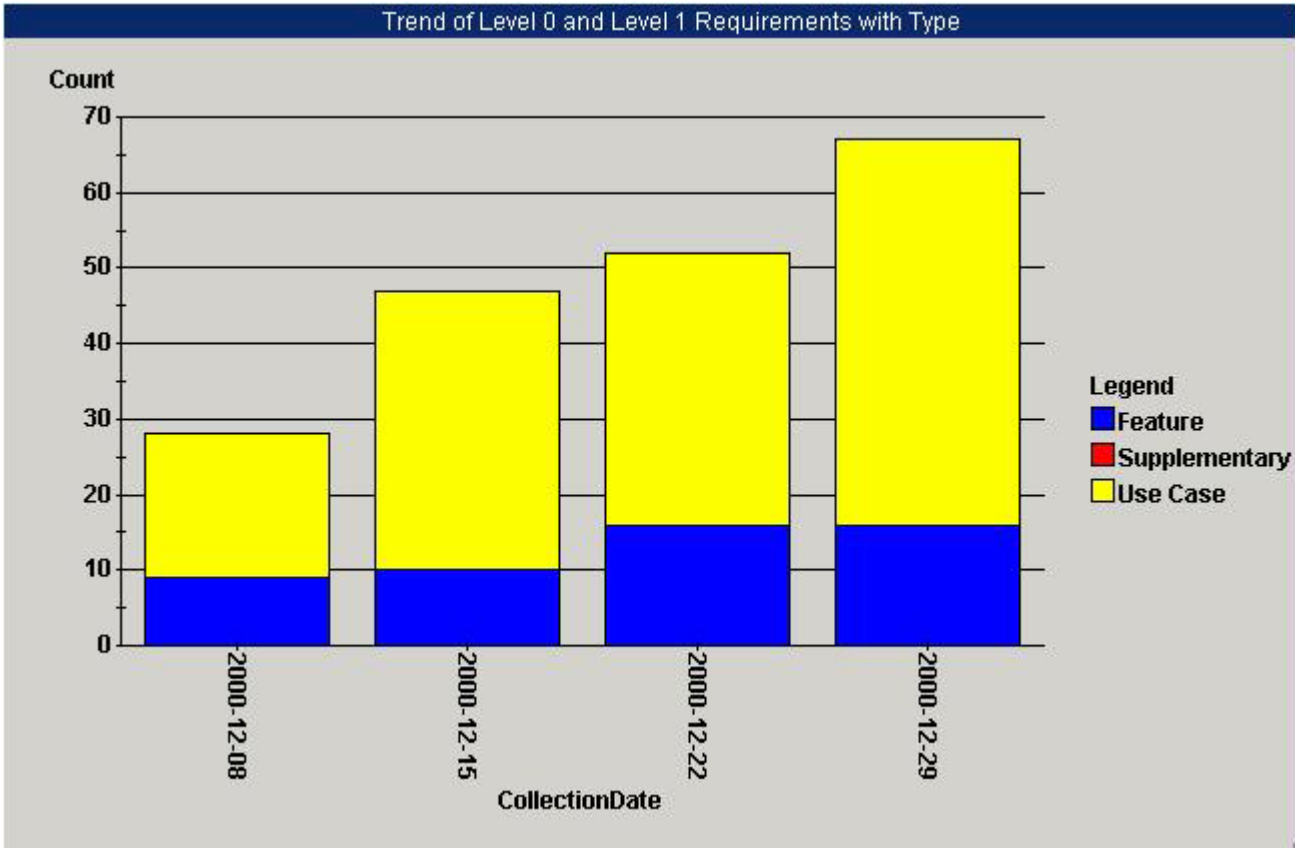
As the project progresses, more requirements will be defined and at a greater level of detail. Level 0 and Level 1 Requirements define the level of detail in which the Requirements have been defined with Level 0 requirements being at the top level of the hierarchy, level 1 being the next level of detail, etc. We should see the number of level 0 and level 1 requirements stabilizing towards the end of inception.



Symptom	Response
The chart keeps increasing	The scope hasn't yet stabilized. This may indicate that inception hasn't completed.
The number of level 1 requirements continues to increase as the number of level 0 requirements stabilized.	This indicates that the project is gaining a better understanding of the defined scope.

Trend of Level 0 and 1 Requirements with Type

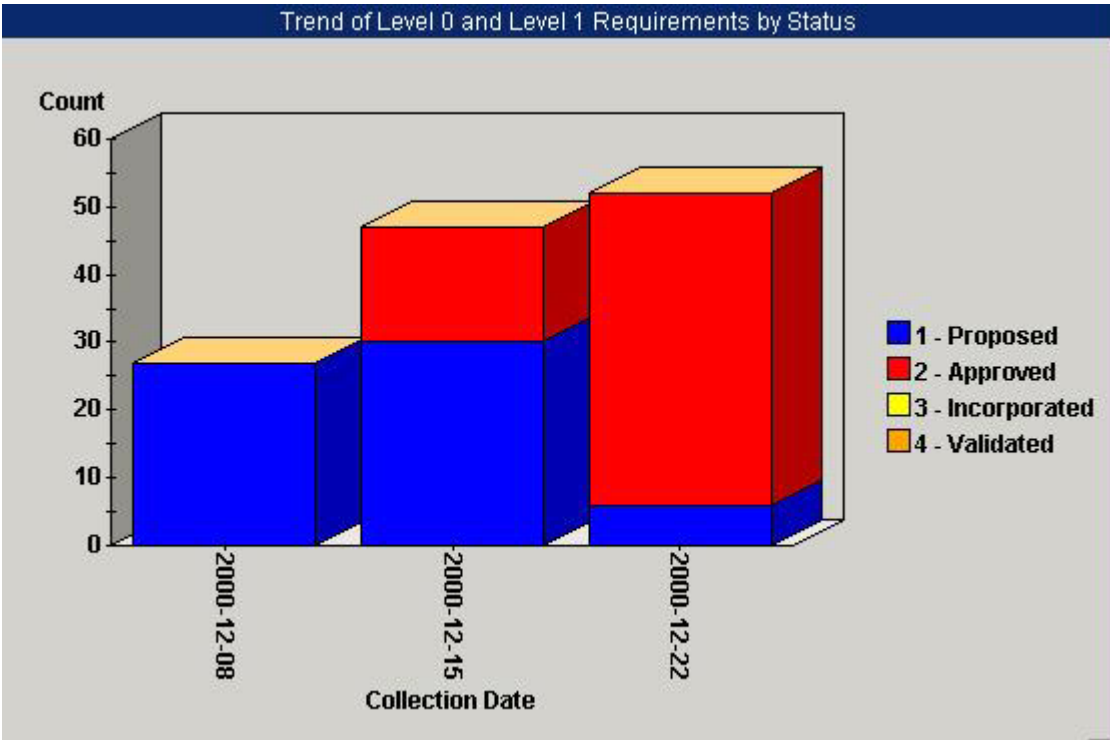
This chart shows the types of information, such as use cases, features and supplementary features recorded in RequisitePro and the growth of each type of information over time. Is this the type of information your company produces? Are these the growth curves your company expects? As the inception phase progresses, the number of new use cases, features, and supplementary features should increase throughout the phase as new requirements are identified.



Symptom	Response
The number of features stabilizes.	This indicates we've settled on the project scope.
The number of use case requirement types is increasing.	During inception the scope of the system is identified. To exit inception that scope must be stable. Use case requirements will continue to grow through inception and elaboration where as features should stabilize through inception.

Trend of Level 0 and 1 Requirements with Status

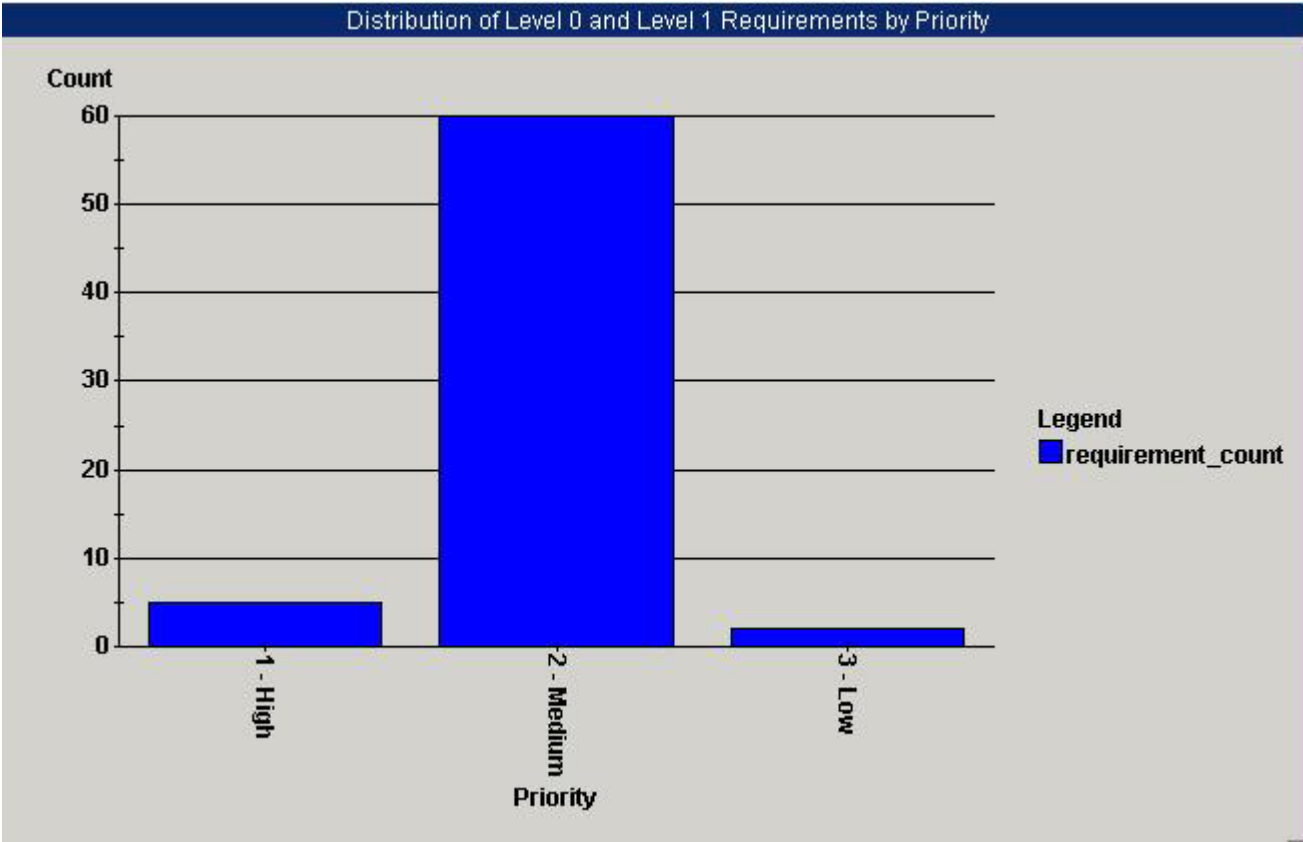
Requirements are transitioned from Proposed to Approved to Incorporated and to Validated. When requirements are first introduced, they are assigned a status of Proposed. Once Proposed, requirements will be Approved, Rejected, or remain in the Status of Proposed. Approved requirements will at some point be Incorporated as construction progresses. Incorporated requirements will then need to be Validated. At the beginning of inception, the number of requirements in the Proposed status is high and the number of requirements in the Approved status is low. By the end of inception, the number of requirements in the Approved status should be high and the number of requirements in the Proposed status should be low. The following chart only shows Proposed and Approved requirements representing time-periods from early to late inception. No requirements have yet been constructed.



Symptom	Response
The majority of requirements are being approved.	We're gaining consensus on the requirements for this project.
None of the identified Requirements have yet been incorporated.	We haven't done any prototyping during inception. This may be a good or bad thing depending on the nature of the project. For some projects scope cannot be identified without doing some sort of prototyping. Depending on scope and risk, projects may not require the usage of a technology prototype.

Distribution of Level 0 and 1 Requirements by Priority

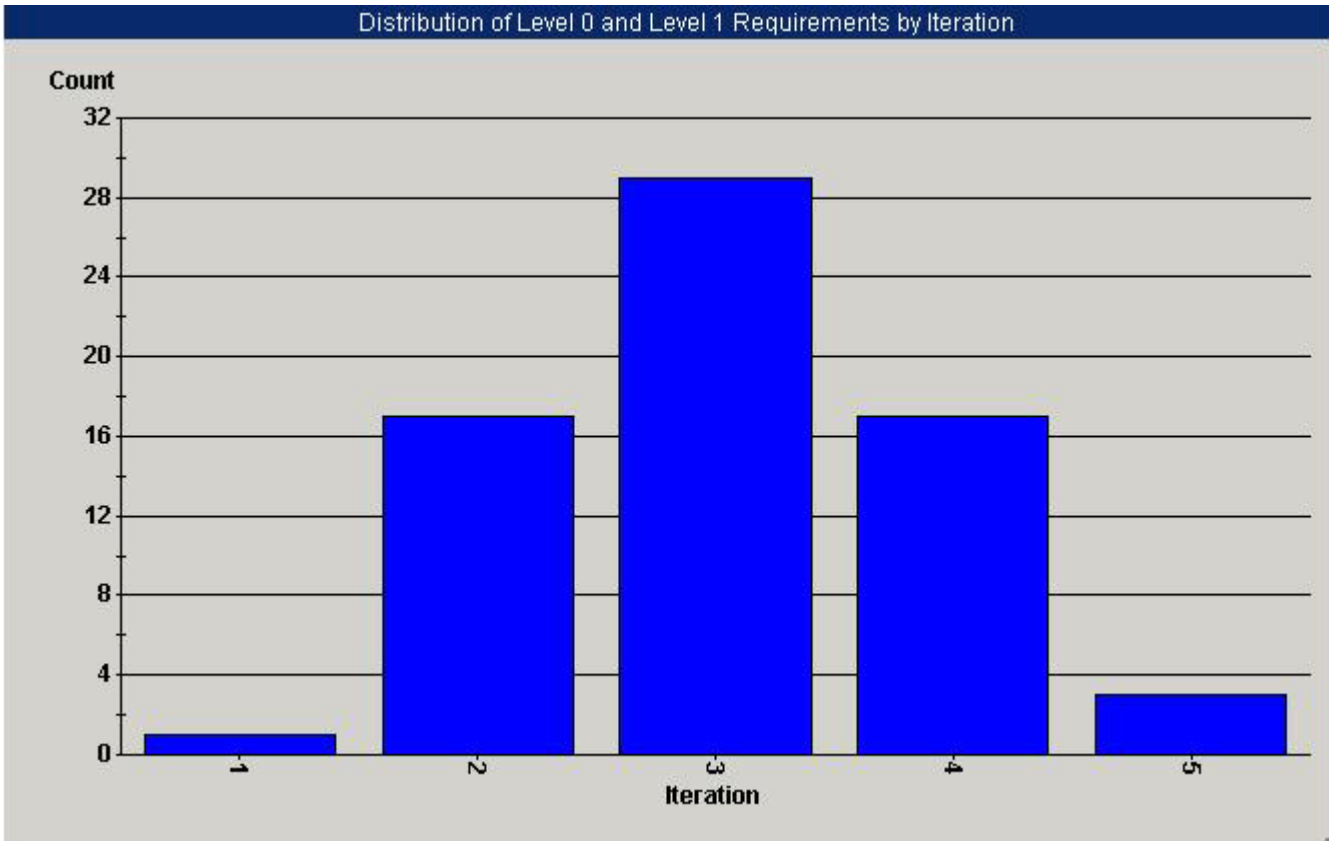
Towards the end of inception, the understanding of the requirements should become clearer. Information about the prioritization of requirements should become more complete.



Symptom	Response
The distribution is unbalanced.	This could be an indication that nobody has really made any tough decisions about the priority of the requirements.
No requirements have a null priority.	All identified Level 0 and 1 requirements have been assigned some type of priority. Currently, there are no non-prioritized requirements.

Distribution of Level 0 and 1 Requirements by Iteration

Towards the end of inception, the understanding of the requirements should become clearer. Information about ownership and planned iteration should become more complete.



Symptom	Response
The planned work isn't evenly distributed across the iterations.	The shape of this chart will depend on the amount of resources available in each iteration and how difficult the requirements are to implement. In this sample project the first construction iteration is #3, which is when most of the work is done. Be aware of a heavy load of requirements towards the end of elaboration. Is the plan realistic?
Roughly 20% of the requirements are being implemented in the Elaboration phase.	This indicates that the architecture is relatively stable.

Measuring the Elaboration Phase

The trend charts in this section show data from the beginning of the project through the elaboration phase.

The elaboration phase of a RUP project is focused on understanding the architecture's scope, major functionality, and nonfunctional requirements such as performance requirements.

The primary objectives of the elaboration phase are:

- To establish and demonstrate a sound architectural foundation
- To analyze the problem domain
- To design the solution
- To address the highest risk elements of the project
- To develop a comprehensive plan for the construction and transition of the project
- To refine previous course-grained plans

Products of the elaboration phase are:

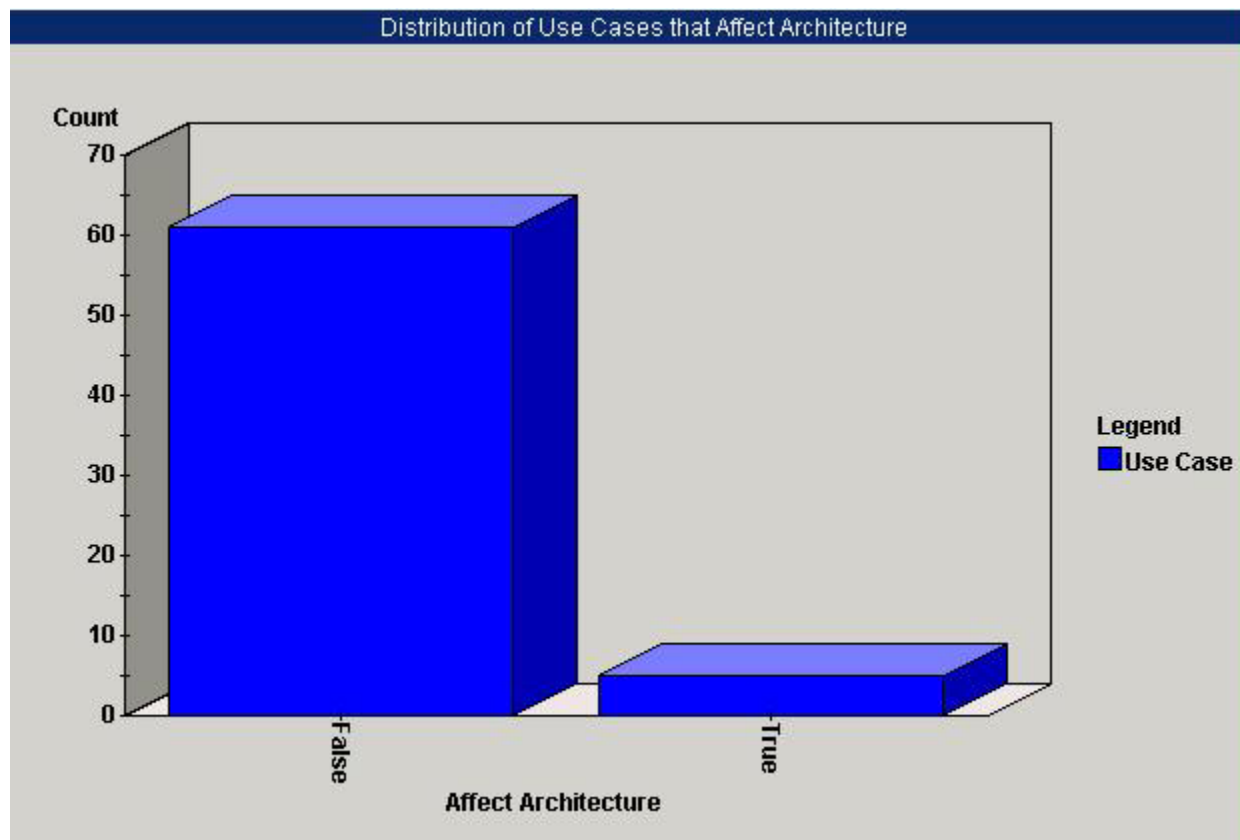
- Use case model and other requirements
- Software Architecture Document
- Executable Baselines Architecture
- Revised Software Development Plan

The elaboration phase is evaluated based on:

- Stability of product vision
- Stability of architecture
- Resolution of risks
- Sufficiency and credibility of plan for construction
- Stakeholder buy-in
- Actual Expenditures versus planned

Distribution of affects-architecture of Use Cases

By the end of elaboration, all use cases that affect the architecture should be verified. Affects-architecture use cases are those use cases that have direct impact on the architectural design.

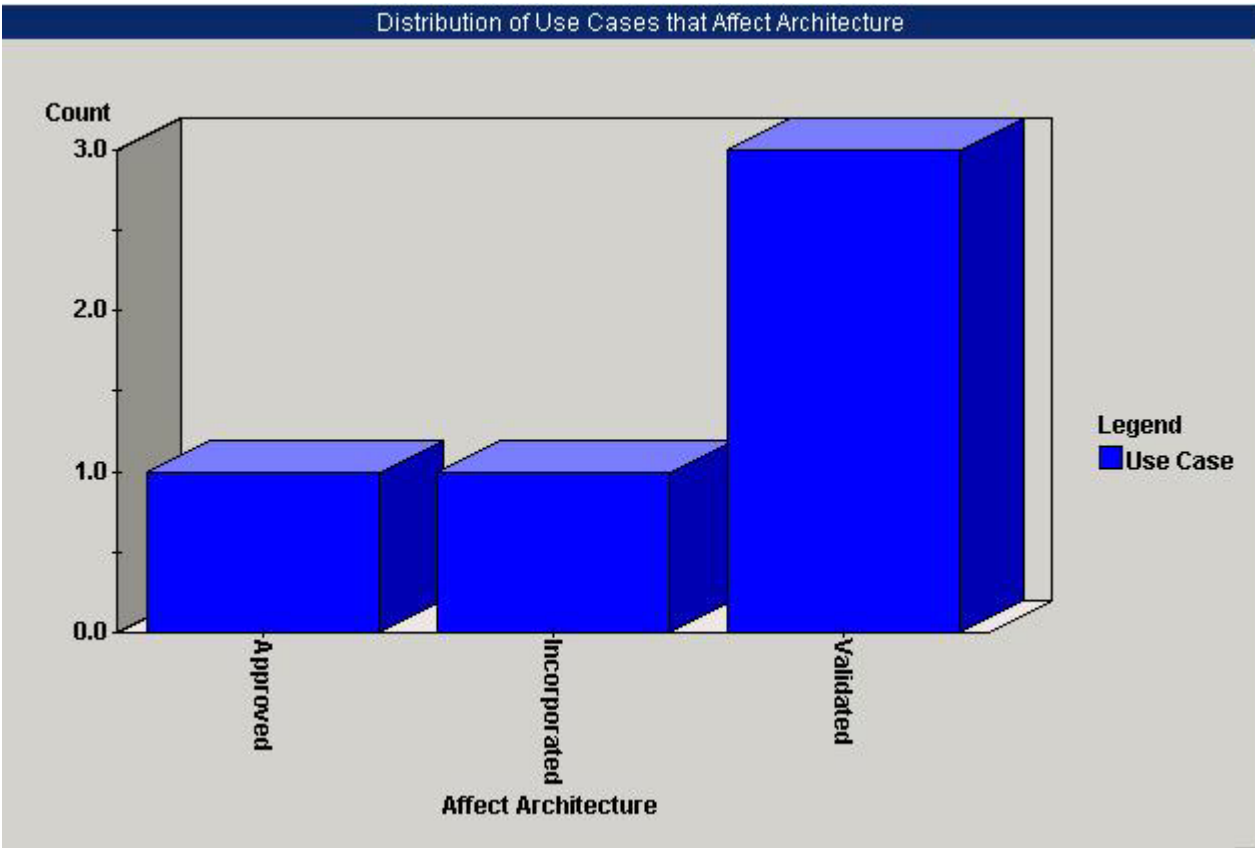


* Note: This chart not included in the ProjectConsole Sample Site Metrics, but easily generated.

Symptom	Response
There are a few affects-architecture use cases that must be verified before the elaboration phase can end.	During the elaboration phase the affects-architecture use cases should be validated. If no use cases have yet been developed, it will not be possible to complete the elaboration milestone and exit that phase.

Distribution of affects-architecture Use Cases by Status

Drilling-Down further into the Status of the Use Cases that Affect Architecture, we see the following chart:

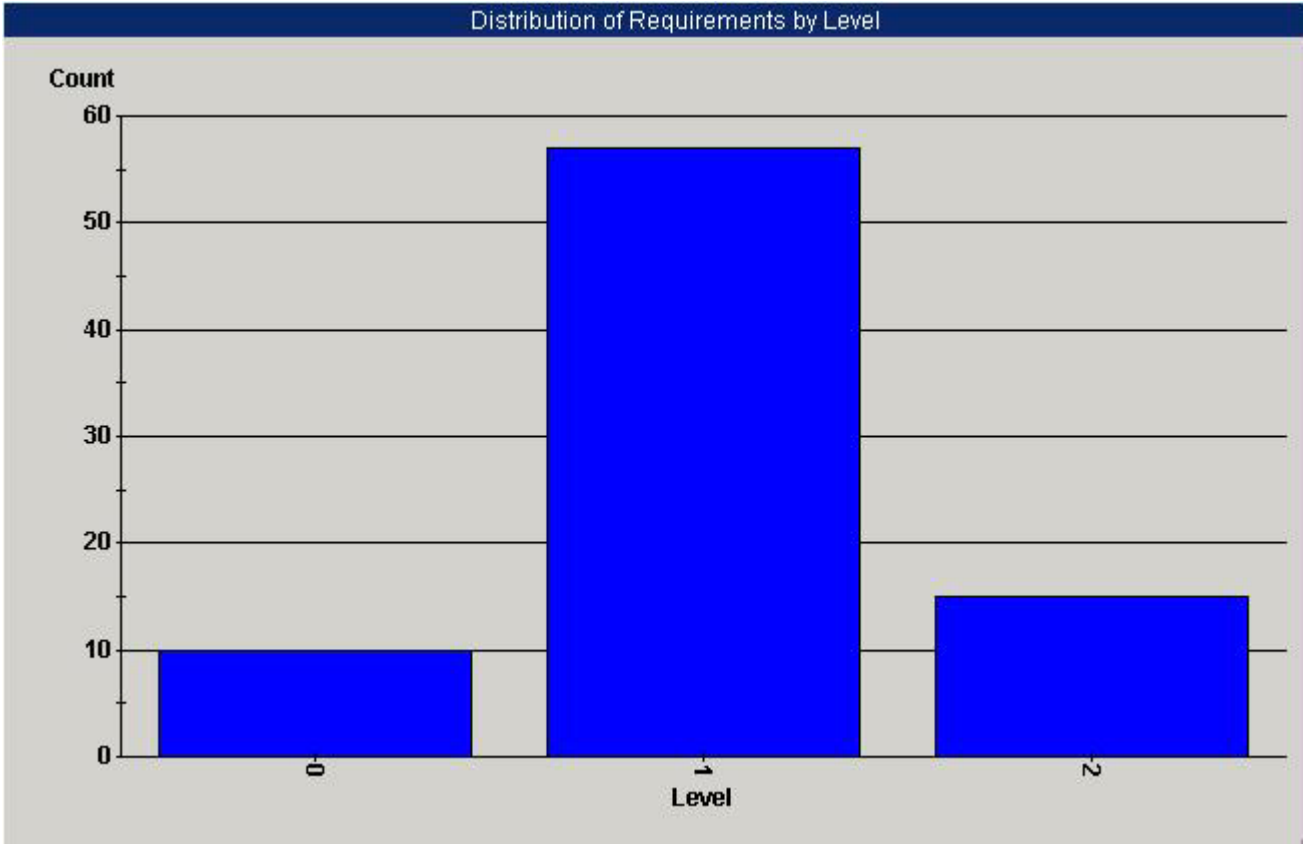


* Note: This chart not included in the ProjectConsole Sample Site Metrics, but easily generated.

Symptom	Response
There are a few affects-architecture use cases that must be verified before the elaboration phase can end.	By drilling-down into the data, we see that there are a total of 3 defined Use Cases that affect architecture and that all 3 Use Cases have been validated. Therefore, from an architectural design standpoint, elaboration is ready to conclude.

Distribution of Requirements by Level

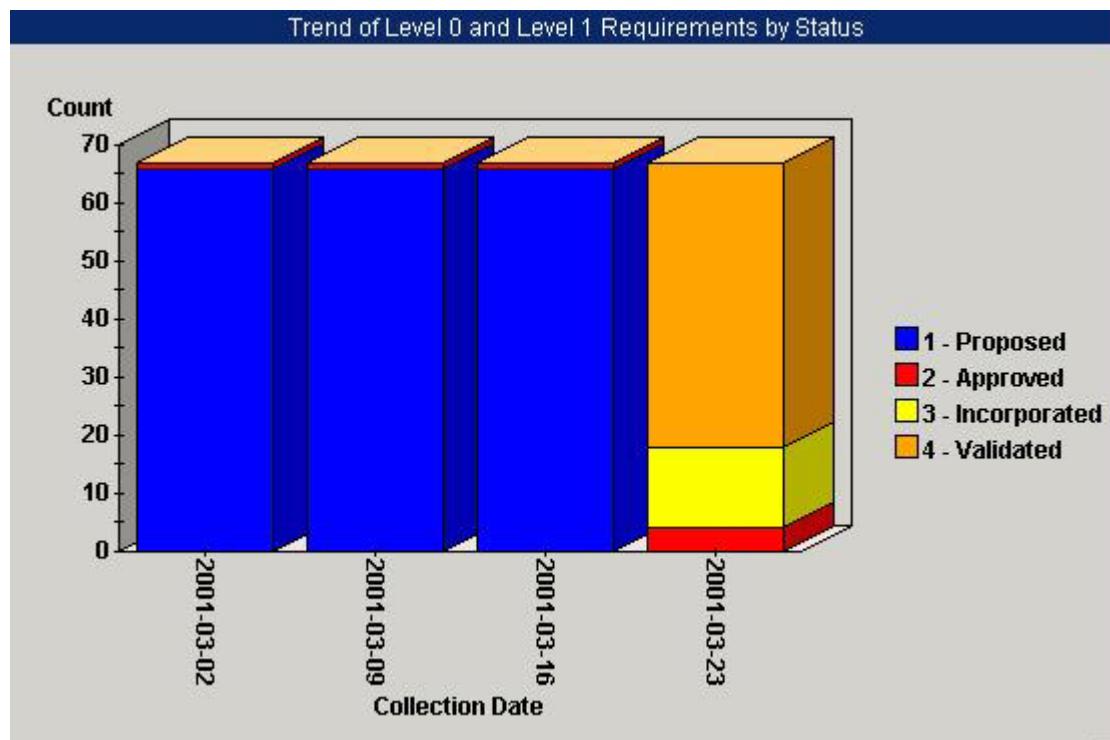
As elaboration progresses, more requirements are being defined at a greater level of detail.



Symptom	Response
Not enough detail in requirement definitions. Too many requirements still only defined at a Level 0 of detail (not detailed enough).	Requirements need to be detailed-out by the end of elaboration. More prototyping may be needed.

Trend of Level 0 and 1 Requirements with Status

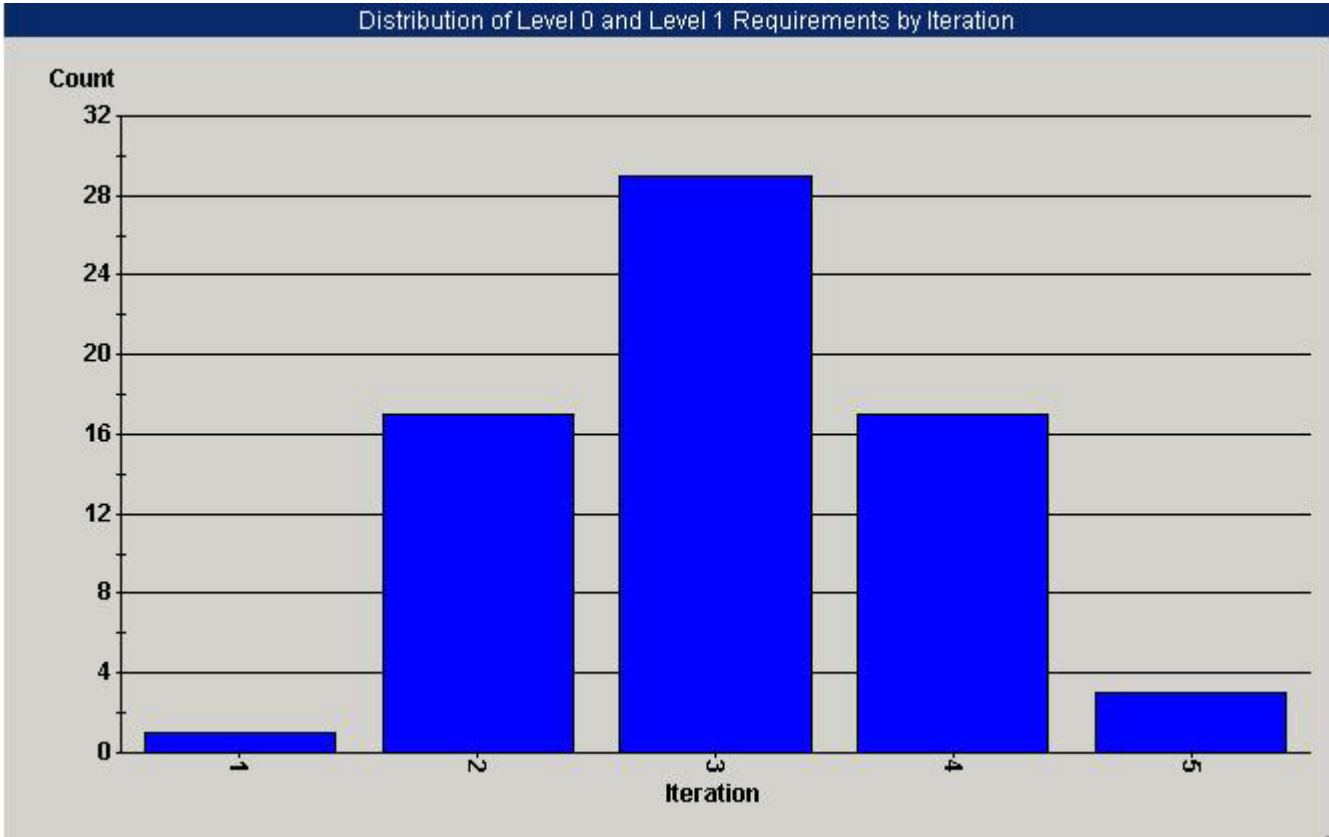
Towards the end of elaboration, requirements should begin to be incorporated and the number of Approved vs. Incorporated requirements should begin to stabilize. There should be a trend of more Incorporated requirements.



Symptom	Response
Towards the end of elaboration, the number of approved requirements should be stabilizing.	We're gaining consensus on the requirements for this project.
More requirements are being incorporated.	As prototyping takes place, more of the approved requirements are beginning to be incorporated.

Distribution of Level 0 and 1 Requirements by Iteration

By the end of elaboration, planning should be completed. The goal should be to have an even distribution of requirements that will be worked on in each iteration.



Symptom	Response
The planned work isn't evenly distributed across the iterations.	By the end of elaboration, we should have a very good and realistic plan.

Measuring the Construction Phase

The construction phase of a RUP project is focused on developing and integrating all remaining components and application features into the product, and testing all features thoroughly.

The primary objectives of the construction phase are:

- To minimize development costs by optimizing resources and avoiding unnecessary scrap and rework
- To achieve adequate quality as rapidly as practical
- To achieve useful alpha, beta, and other test release versions as rapidly as practical

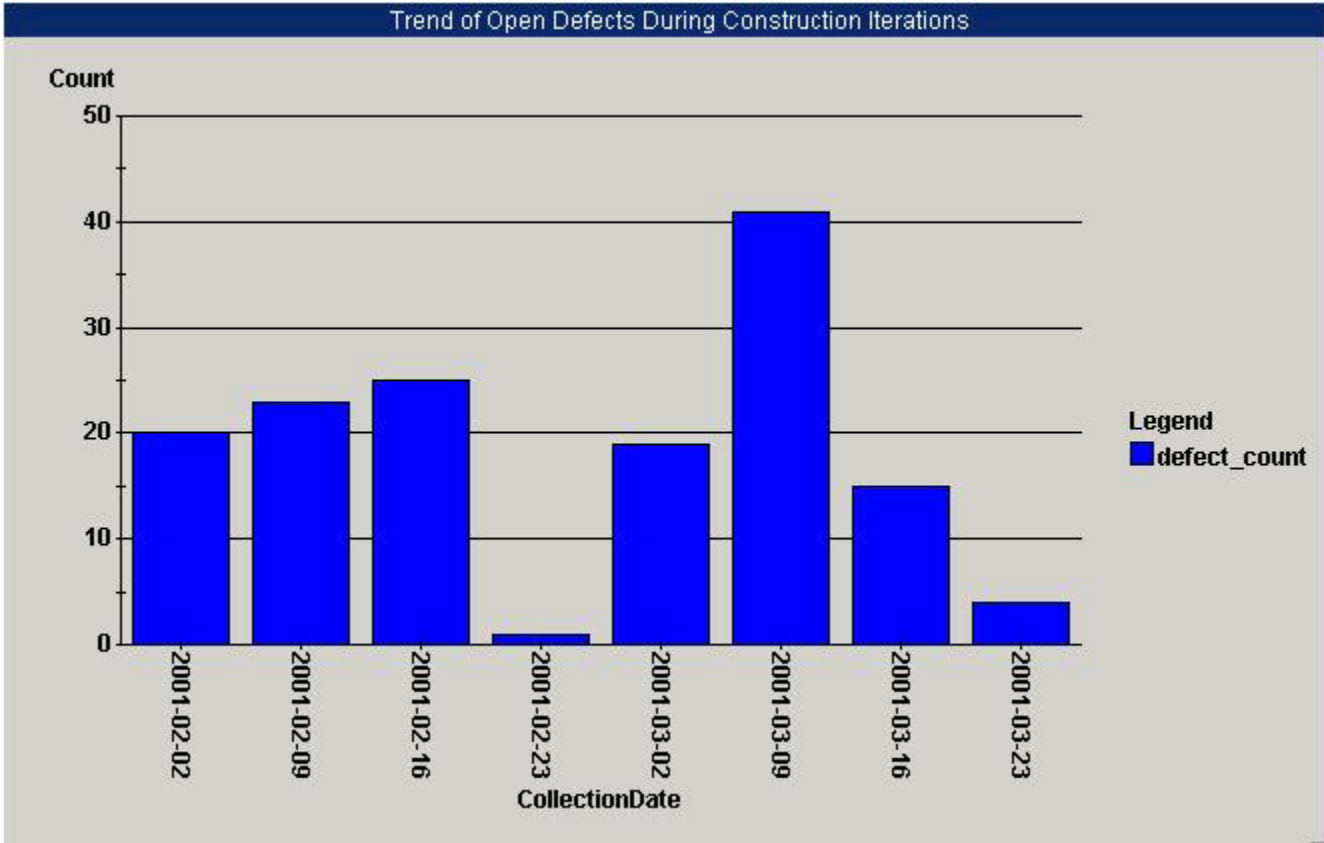
Products of the construction phase are:

- Executable releases of increasing functionality
- Models of the system's design and behavior
- User documentation
- Deployment documentation
- Evaluation criteria for each iteration
- Release descriptions, including quality assurance results
- Updated development plan

Evaluation of the Construction phase is based on:

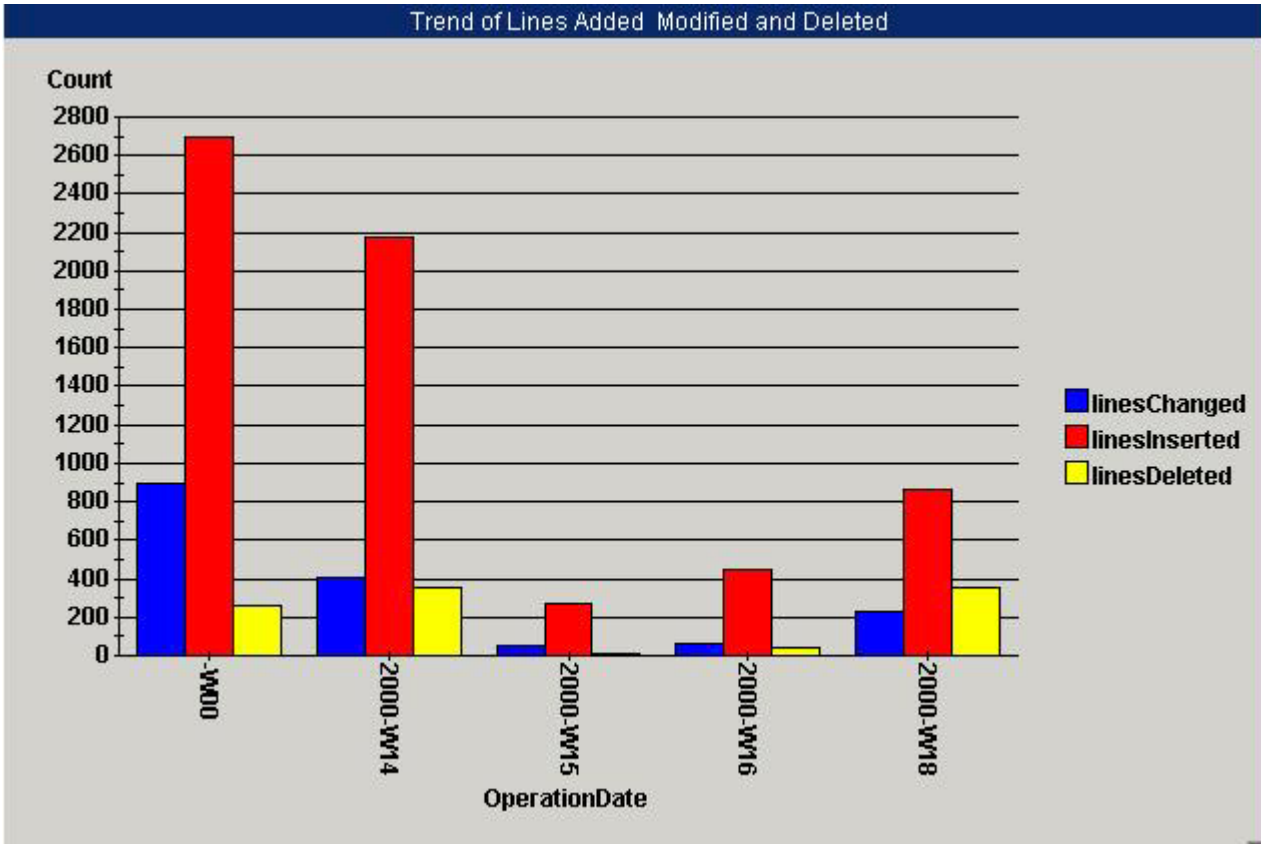
- Stability and maturity of product releases
- Readiness of stakeholders for transition to user community
- Actual expenditures versus planned

Trend of Open Defects in Construction Iterations



Symptom	Response
The trend of open defects is reaching near zero towards the end of each construction iteration.	Development is stabilizing the product by the end of each construction iteration. This is a good trend!
The number of open defects continues to increase.	Testing is continuing to find defects. If a project is not discovering defects often, that is an indication that the testing process is not working correctly. Finding defects early is a good project characteristic.

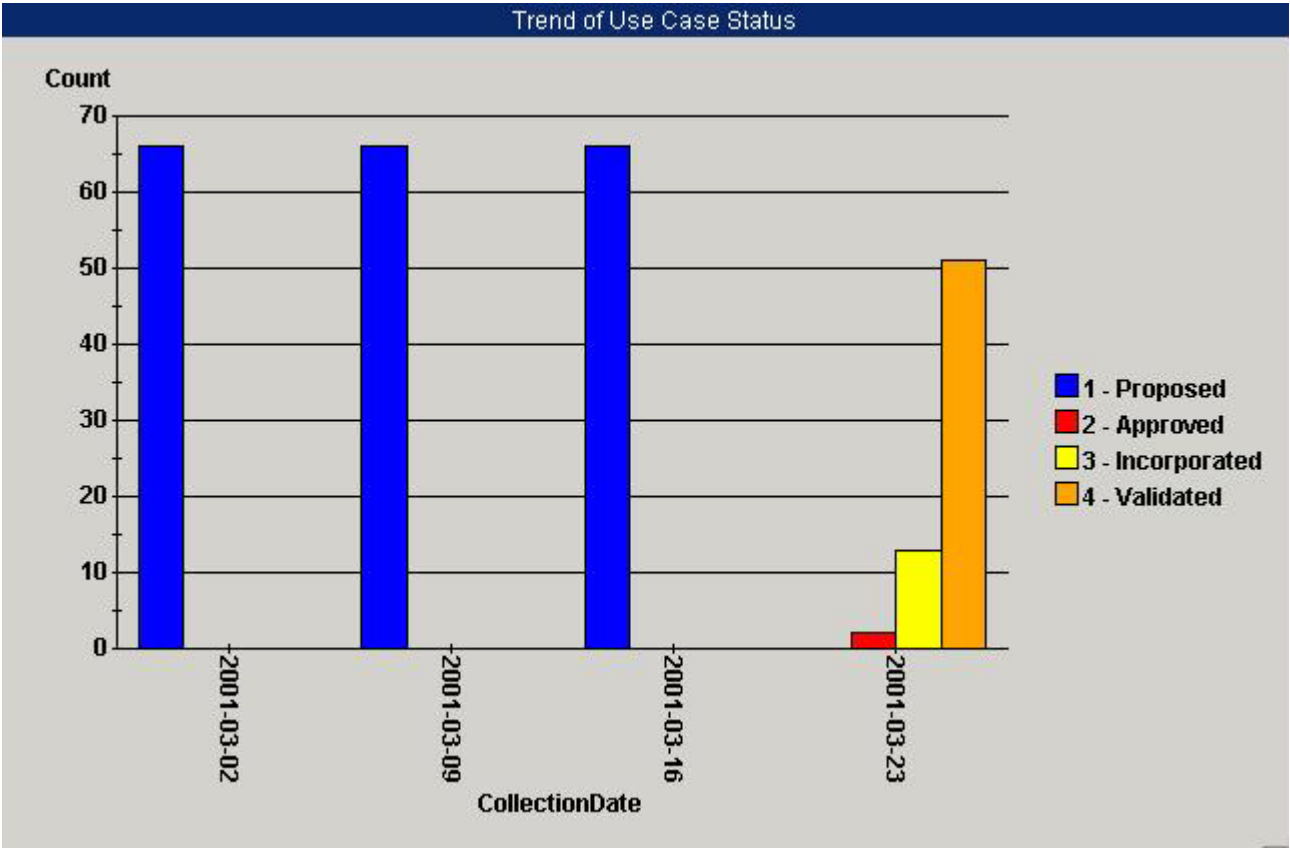
Trend of Lines of code Added, Modified, and Deleted



Symptom	Response
LOTS of code was added at the beginning compared to the end.	This is good! The trend is that change is reducing during construction. The code base is stabilizing with less code being added. The more code added, the more testing and bugs will be found.
There should not be a large amount of code churn at the end.	Make sure that developers are not waiting too long until they are checking in their code. Also, daily or weekly builds should be occurring.

Trend of Use Case Status

The following chart shows a trend of the number of Approved, Incorporated, and Validated, Use Cases project-to-date.



* Note: This chart not included in the ProjectConsole Sample Site Metrics, but easily generated.

Symptom	Response
Validation is not happening until the end.	This indicates a waterfall development process, not an iterative process, Or the testing process is reporting progress until the end. The project manager needs to make sure that enough test resources have been assigned that these resources are identified early enough.

Measuring the Transition Phase

The trend charts in this section show data from the beginning of the project through the transition phase. The transition phase of a RUP project is focused on moving the software product to the user community.

The primary objectives of the transition phase are:

- To achieve user self-supportability
- To achieve stakeholder concurrence that deployment baselines are complete and consistent with the evaluation criteria of the vision
- To achieve final product baseline as rapidly and cost effectively as practical

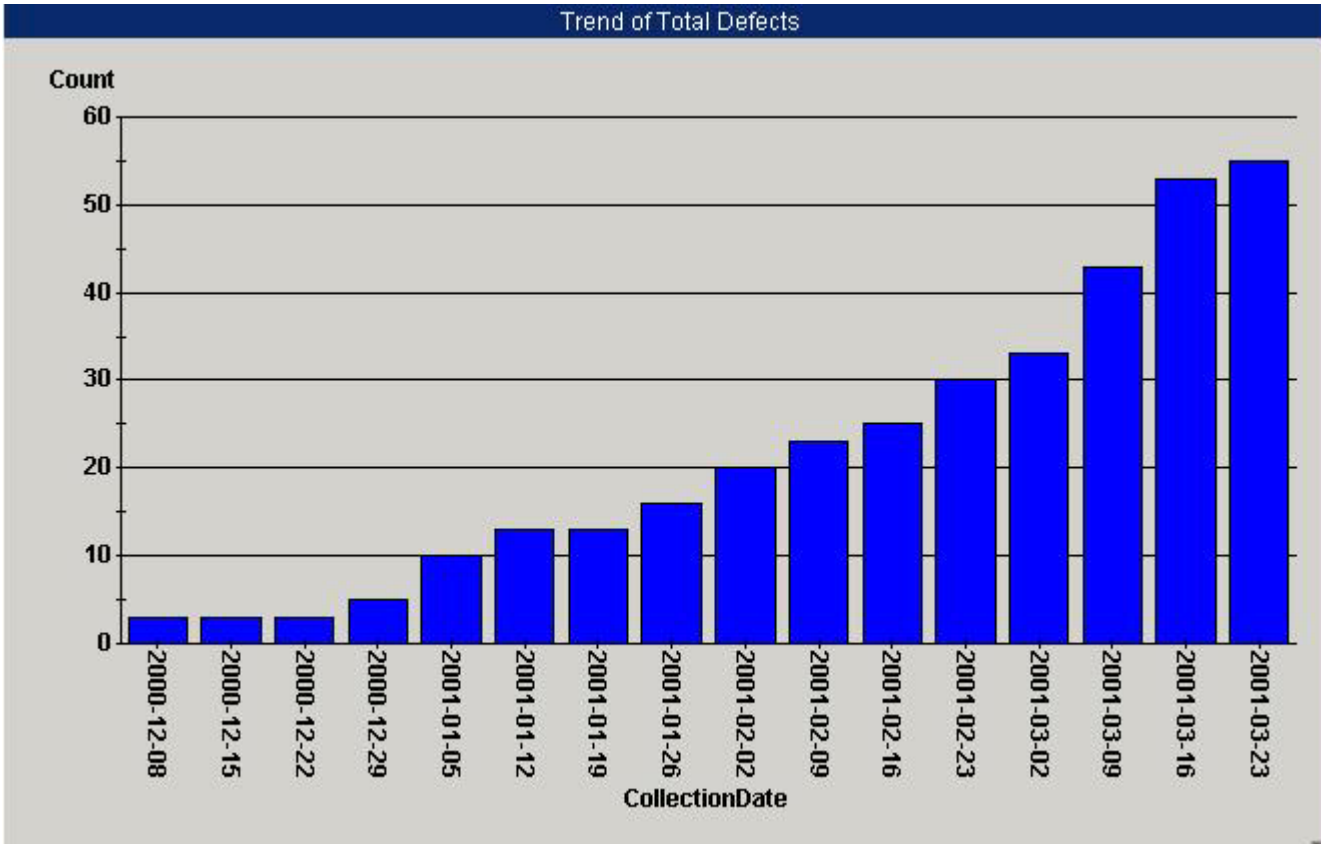
The products of the transition phase are:

- Executable releases
- Updated system models
- Release descriptions, including quality assurance results
- Updated user manuals
- Updated deployment documentation
- Training Materials
- Project Close-Out Plan portion of the Software Development Plan

Evaluation is based on:

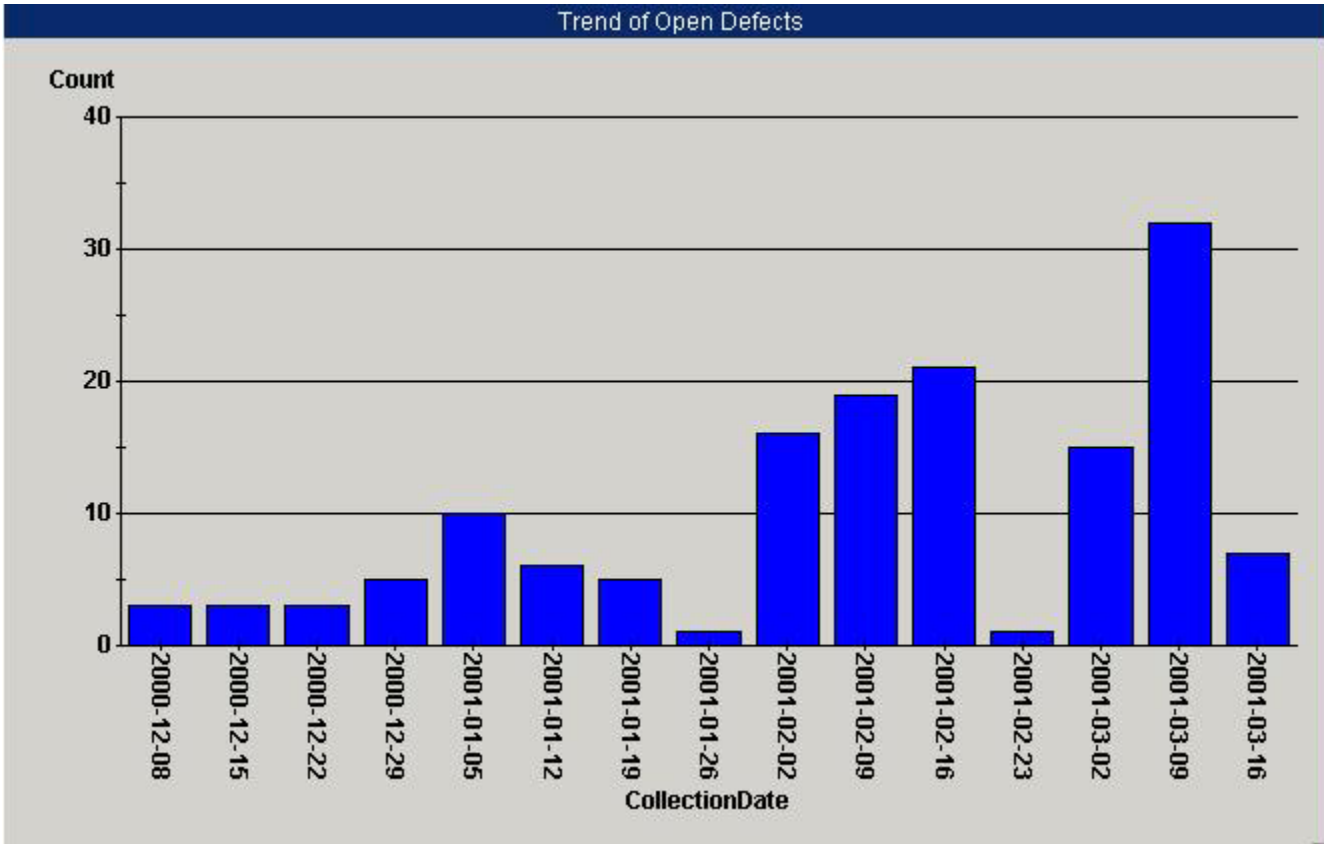
- Satisfaction of user
- Actual expenditures versus planned

Trend of Total Defects found in this Release



Symptom	Response
We will hope to see a leveling-off of total defects found during transition.	This will indicate that most defects were found during earlier iterations.

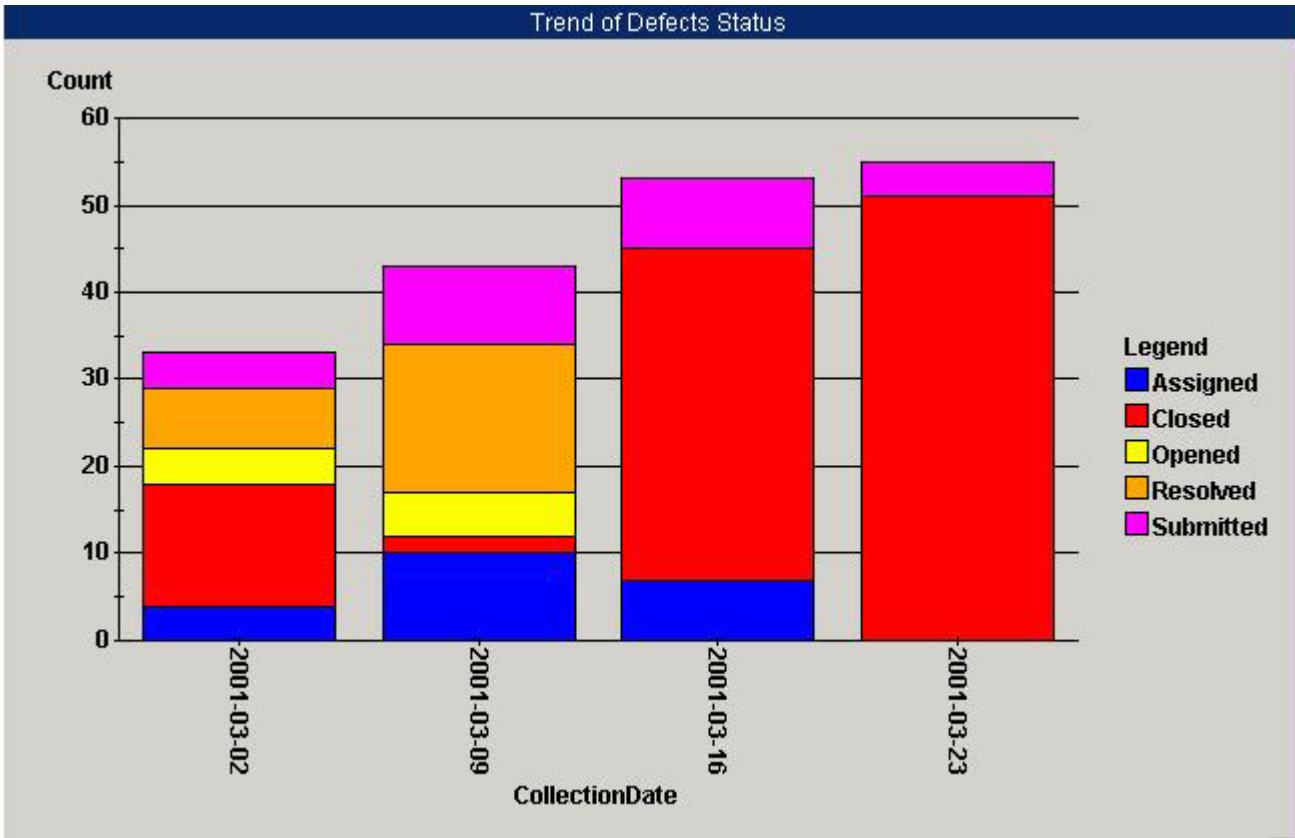
Trend of Open Defects in this Release



Symptom	Response
The trend of open defects should decrease during transition.	You are nearing release, so the quality of the product is reaching an acceptable level.

Trend of Defects Status in this Release

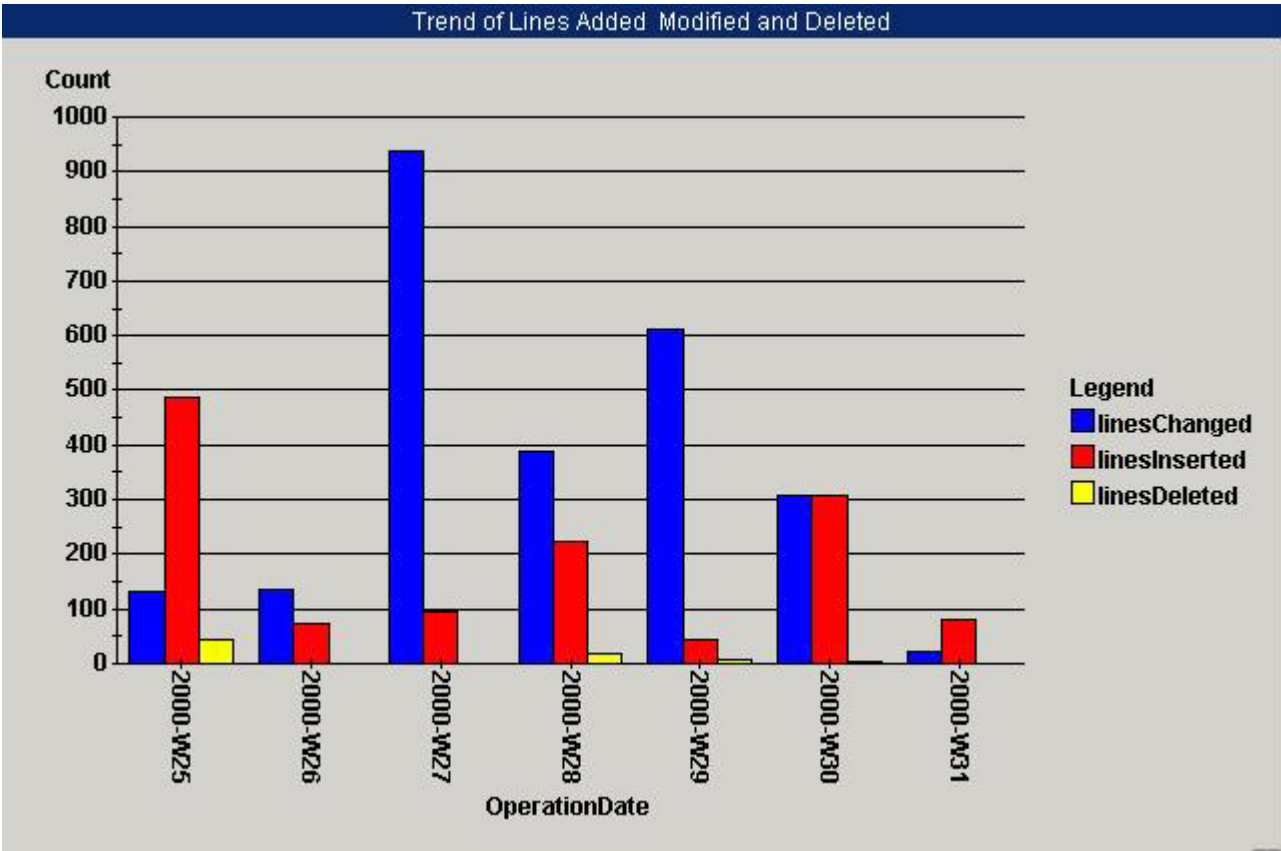
The following chart shows a trend of the number of the status of all defects.



* Note: This chart not included in the ProjectConsole Sample Site Metrics, but easily generated.

Symptom	Response
Transition is nearing completion.	Towards the end of transition the majority of identified defects should be closed.

Trend of Lines Added, Modified, and Deleted



Symptom	Response
There should not be a large amount of code churn at the end.	If there is still a large amount of code churn, there is a big risk that quality problems have been introduced. There may be an indication here that more regression testing needs to be done.

Summary

This paper introduces key metrics that can be used to measure the progress and level of quality during a Rational Suite project. We guided you through how metrics are used in the four phases of a Rational Unified Process project: Inception, Elaboration, Construction and Transition.

For the purposes of this paper, the various metrics charts represent a project that has a one month Inception Phase, a one month Elaboration Phase, 3 one month iterations during the Construction Phase, and a one month Transition Phase.

The sample metric charts are produced using Rational's ProjectConsole product. Rational ProjectConsole enables a software development team to automatically quantify the current project status and assess development trends of their project with up-to-date metrics. On a specified scheduled or on-demand basis, metrics data is collected from the Rational Suites' development environment, and from selected 3rd party tools, and then stored in the metrics warehouse. The resulting analysis is then visually presented in graphs, charts and gauges to indicate project status at a glance.

By viewing charts and indicators, team members can quickly understand the true status of the progress and quality of their project. ProjectConsole provides all members with the ability to analyze the individual discipline metrics, low-level details, planned-versus-actual metrics, historic data, trend charts or cross discipline metrics to get a better view across the entire project. These capabilities enable the software development team to take prompt corrective actions, realize the cause for late deliverables, set realistic project expectations, forecast future project milestones, and ultimately, put the entire team in a better position to objectively and accurately measure project progress and quality.

References

- [1] – Kruchten, P, *The Rational Unified Process An Introduction, 2nd Edition*, Addison-Wesley, 2000.
- [2] – Rational Unified Process version 2001A.



Corporate Headquarters
18880 Homestead Road
Cupertino, CA 95014
Toll-free: 800-728-1212
Tel: 408-863-9900
Fax: 408-863-4120
E-mail: info@rational.com
Web: www.rational.com

For International Offices: www.rational.com/worldwide

Rational, the Rational logo, Rational the e-development company and Rational Rose are registered trademarks of Rational Software Corporation in the United States and in other countries. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word, Microsoft Project, Visual C++ and Visual Basic are trademarks or registered trademarks of Microsoft Corporation. All other names used for identification purposes only and are trademarks or registered trademarks of their respective companies. ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2001 Rational Software Corporation.

TP-#### /00. Subject to change without notice.