

Reaching CMM Levels 2 and 3 with the Rational Unified Process

Rational Software White Paper



Rational[®]
the e-development company™

Table of Contents

Abstract	1
Introduction.....	1
Level 2, Repeatable	3
Requirements Management.....	3
Software Project Planning.....	4
Software Project Tracking and Oversight	5
Software Subcontract Management.....	6
Software Quality Assurance	6
Software Configuration Management	7
Level 3, Defined.....	8
Organization Process Focus	8
Organization Process Definition	9
Training Program.....	9
Integrated Software Management.....	10
Software Product Engineering	10
Intergroup Coordination.....	10
Peer Reviews.....	11
References.....	11

Abstract

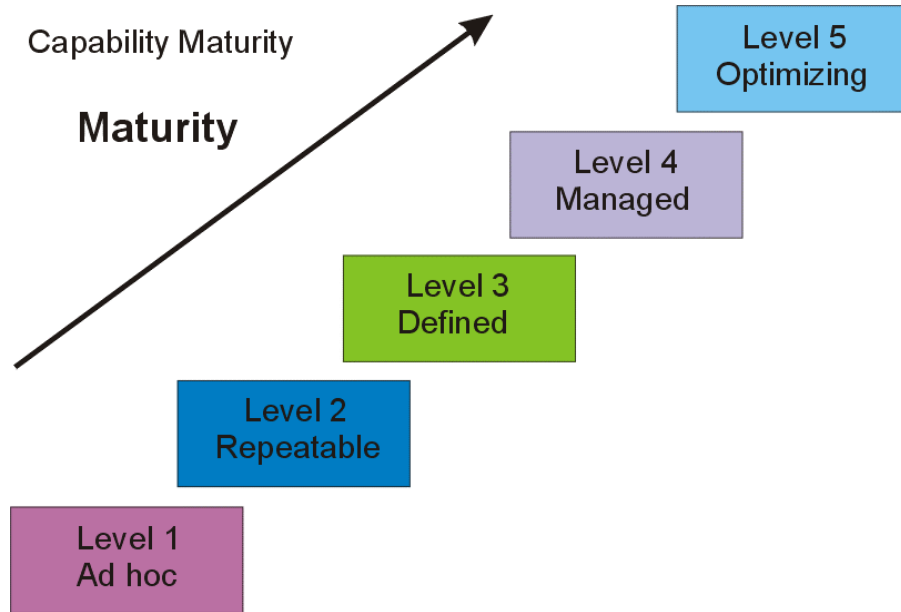
The Software Engineering Institute’s (SEI) Capability Maturity Model (CMM) provides a well-known benchmark of software process maturity. The CMM has become a popular vehicle for assessing the maturity of an organization’s software process in many domains. This white paper describes how the Rational Unified Process® can support an organization that is trying to achieve CMM Level-2, Repeatable, and Level-3, Defined, software process maturity levels.

Introduction

The Software Engineering Institute (SEI) Capability Maturity Model (CMM) is a framework that describes the elements of an effective software process; see the section titled *References*, [REF 1], at the end of this document. The CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process.

The CMM covers practices for planning, engineering and managing software development and maintenance. These key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality.

The CMM has five levels of maturity: Level-1 to Level-5. As illustrated in the following figure, each maturity level is composed of Key Process Areas (KPAs), and each KPA identifies a cluster of related activities. When performed collectively, these related activities achieve a set of goals considered important for establishing process capability at that maturity level.



Level-2, the Repeatable Level, is defined as follows:

At the Repeatable Level, policies for managing a software project and procedures to implement those policies are established. Planning and managing new projects is based on experience with similar projects. An objective in reaching Level-2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices developed in earlier projects, although the specific processes implemented by the projects may differ. An effective process can be characterized as practiced, documented, enforced, trained, measured, and able to improve.

Projects in Level-2 organizations have installed basic software management controls. Realistic project commitments are based on the results observed on previous projects and on the requirements of the current project. The software managers for a project track software costs, schedules, and functionality; problems in meeting commitments are identified when they arise. Software requirements and the work products developed to satisfy them are baselined, and their integrity is controlled.

Software project standards are defined, and the organization ensures that they are faithfully followed. The software project works with its subcontractors, if any, to establish a strong customer supplier relationship.

The software process capability of Level-2 organizations can be summarized as disciplined because planning and tracking of the software project is stable and earlier success can be repeated. The project's process is under the effective control of a project management system, following realistic plans based on the performance of previous projects.

Level-2 KPAs are:

- Requirements Management
- Software Project Planning
- Software Project Tracking and Oversight
- Software Subcontract Management
- Software Quality Assurance
- Software Configuration Management

Level-3, the Defined Level, is defined as follows:

At the Defined Level, the standard process for developing and maintaining software across the organization is documented, including both software engineering and management processes, and these processes are integrated into a coherent whole. The standard process is referred to throughout the CMM as the organization's standard software process. Processes established at Level-3 are used (and changed, as appropriate) to help the software managers and technical staff perform more effectively. The organization exploits effective software engineering practices when standardizing its software processes. There is a group that is responsible for the organization's software process activities; for example, software engineering or SEPG. An organization-wide training program is implemented to ensure that the staff and managers have the knowledge and skills required to fulfill their assigned roles.

Projects tailor the organization's standard software process to develop their own defined software process, which accounts for the unique characteristics of the project. This tailored process is referred to in the CMM as the project's defined software process. A defined software process contains a coherent, integrated set of well-defined software engineering and management processes. A well-defined process can be characterized as including readiness criteria, inputs, standards, and procedures for performing the work; verification mechanisms, such as peer reviews; outputs; and completion criteria. Because the software process is well defined, management has good insight into technical progress on all projects.

The software process capability of Level-3 organizations can be summarized as standard and consistent because both software engineering and management activities are stable and repeatable. Within established product lines, cost, schedule, and functionality are under control, and software quality is tracked. This process capability is based on a common, organization-wide understanding of the activities, roles, and responsibilities in a defined software process.

Level-3 KPAs are:

- Organization Process Focus
- Organization Process Definition
- Training Program
- Integrated Software Management
- Software Product Engineering
- Intergroup Coordination
- Peer Reviews

Each section of this paper describes how Rational Unified Process features, methods, procedures, and artifacts meet KPA goals.

This paper is written for organizational personnel concerned with achieving organizational maturity Level-2 and Level-3 within the CMM framework.

Level 2, Repeatable

Requirements Management

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. This agreement with the customer is the basis for planning (as described in the Software Project Planning KPA) and managing (as described in Software Project Tracking and Oversight) the software project. Control of the relationship with the customer depends on following an effective change control process, as described in Software Configuration Management.

One of the key features of the Rational Unified Process is that it is *use-case driven*. Use cases represent a systematic approach to eliciting, organizing, and communicating user requirements. They provide the way to document functional requirements that serve as a basis for project development, testing, and iteration planning. In the Rational Unified Process, use cases are maintained in a use-case model and referenced consistently throughout the project lifecycle, from analysis through testing to maintenance.

The Rational Unified Process artifacts that capture requirements in the engineering context are:

- Use-Case Model consisting of Use Cases and Use-Case Packages
- Non-functional "Supplementary Specifications"
- Use-Case-Model Survey
- Use-Case Report
- Glossary

The Rational Unified Process artifacts that describe use cases and scenarios (requirements) that are to be developed, as used in the management context, are:

- Iteration Plan
- Integration Build Plan
- Project Plan
- Software Development Plan

All of these artifacts are baselined and subjected to a change management discipline.

Goal-1: System requirements that are allocated to software are controlled to establish a baseline for software engineering and management use.

The Rational Unified Process advocates on-going configuration control of all evolving artifacts, however, the 'formal' baselines correspond to the following milestones.

- Lifecycle Objectives Milestone (Inception Phase)
- Lifecycle Architecture Milestone (Elaboration Phase)
- Initial Operational Capability Milestone (Construction Phase)
- Product Release Milestone (Transition Phase)

As such, the Rational Unified Process is compliant with the CMM for agreement on requirements, their management, tracing, and baselining.

Goal-2: Software plans, products, and activities are kept consistent with the system requirements allocated to software.

The emphasis of this CMM goal is to ensure that delivered systems meet user requirements. The Rational Unified Process helps organizations reach this goal in two ways:

- The *Use-Case Approach* ensures that user requirements are understood and captured. Once captured, the requirements are flowed down to the various "visual" Rational Unified Process models (Use Case, Design, Implementation, and Test) to ensure consistency and adherence.
- The *Controlled Iterative Development* approach is a risk mitigation approach whereby project risks are understood and explored early, and then revisited often. Each progressive iteration, through continuous integration of added functionality, uncovers risks early. Using traditional waterfall methodologies, these risks may have been left undiscovered until late in the development lifecycle. Early risk identification is of direct benefit to managing the project through requirements re-scoping or by prompting other tactical changes.

Rational Unified Process management documents are:

- Business Case
- Software Development Plan
- Measurement Plan
- Risk List
- Project Plan
- Iteration Plan(s)
- Iteration Assessment(s) and Status Assessment(s)

Effective Change Control and Management is another Rational Unified Process feature that ensures that software is developed to specified, traced and allocated requirements.

The Rational Unified Process advocates that each project establishes a Change Control Board (CCB) that can arbitrate on the scope and impact (budgetary, technical or schedule) of proposed changes or defects uncovered during the course of development. To assist the operation of the CCB, the Rational Unified Process recommends the use of a strong configuration management and version control tool/environment.

Software Project Planning

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project. These plans are necessary for managing the software project, as described in the section titled *Software Project Tracking and Oversight*. Without realistic plans, effective project management cannot be implemented.

Goal-1: Software estimates are documented for use in planning and tracking the software project.

One of the goals of the Rational Unified Process is to ensure that the expectations of all parties are synchronized and consistent. This is ensured through periodic assessments throughout the project lifecycle, and is documented in the *Status Assessment Report*. The report calls for tracking data on resources (staffing and financial), top ten risks, technical progress as gauged through metrics, and major milestone results.

The Rational Unified Process makes use of the following classes of metrics:

- Progress (lines of code, number of classes, function points per iteration, rework)
- Stability (type of rework, requirements or implementation volatility)
- Adaptability (rework cost)
- Modularity (extent of rework impact)
- Quality (defect discovery rate, density, depth of inheritance, rework indicator)
- Maturity (test hours per failure)
- Expenditure profiles on resources (planned versus actuals)

Goal-2: Software project activities and commitments are planned and documented.

Rational Unified Process documents that capture project plans and commitments are:

- Business Case
- Software Development Plan
- Measurement Plan
- Risk List
- Project Plan
- Iteration Plan(s)
- Iteration Assessment(s)
- Status Assessment(s)

Goal-3: Affected groups and individuals agree to their commitments related to the software project.

In the Rational Unified Process, the Software Development Plan defines the overall plan for the project; the Iteration Plan defines in detail what is to be accomplished in an iteration. The Iteration Plan Review, required by the Rational Unified Process, exposes the Iteration Plan to all stakeholders, allowing for a consensus to be developed before the iteration begins. From the agreed Iteration Plan, the Project Manager produces a series of work orders, which communicate the intent of the Iteration Plan in detail to the affected project teams and individuals. The Project Manager gets agreement on these work orders with the affected staff so the iteration may proceed.

Software Project Tracking and Oversight

The purpose of Software Project Tracking and Oversight is to establish adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.

Goal-1: Actual results and performances are tracked against the software plans.

As described in the *Software Project Planning* section, the Rational Unified Process has several levels of project plans and a *Status Assessment Report* that is generated to assess planned versus actual performance. This report, generated for specific milestones, is the Project Manager's responsibility.

Major Rational Unified Process milestones correspond to the end of a phase (Inception, Elaboration, Construction or Transition) and have well specified completion criteria. Review opportunities exist at minor milestones at the end of each iteration within a phase, and serve as decision points and lessons learned for future direction.

For example: The goals of the elaboration phase are to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the highest risk elements of the project. Architectural decisions must be made with an understanding of the whole system. This implies that most of the use cases would be described, taking into account some of the constraints: supplementary requirements. To verify the architecture, a system is implemented that demonstrates the architectural choices and executes significant use cases.

At the end of the elaboration phase, the detailed system objectives and scope are examined, as well as the choice of an architecture and the resolution of major risks. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.

The *Risk List* is a Rational Unified Process artifact that provides an overview of all known risks on the project, and serves as input to planning and project assessments. Each risk is described in terms of its impact and the contingency plan that is to be invoked to mitigate the risk in question. The Risk List is developed along with the Business Case to form the basis for a "go" or "no go" decision on the project. The Risk List is maintained throughout the project lifecycle.

Goal-2: Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.

The Rational Unified Process requires the tracking of performance at two levels: as part of the routine monitoring and control of the project, the Project Manager will use collected metrics to produce a Status Assessment. Problems identified in the

Status Assessment are dealt with according to the Problem Resolution Plan for the project, either directly by the Project Manager (through Work Orders), or, for problems of wider scope, through Change Requests. In addition, the results of each iteration are subjected to an Iteration Assessment and Review. This allows the experience of previous iterations to guide the planning for subsequent iterations, managing corrective actions through Change Requests.

Goal-3: Changes to software commitments are agreed to by the affected group and individuals.

The controlled iterative development process, as described in the Rational Unified Process, ensures that stakeholders get regular visibility into project progress and any changes that may be necessary to keep the project on track. Proposed changes are reviewed by a Change Control Board (CCB) to ensure they are realistic and can be accommodated into the overall project schedule.

Software Subcontract Management

The purpose of Subcontract Management is to select qualified software subcontractors and manage them effectively. It combines the concerns of Requirements Management, Software Project Planning, and Software Project Tracking and Oversight for basic management control, along with necessary coordination of Software Quality Assurance and Software Configuration Management, and applies this control to the subcontractor as appropriate.

Goal-1: The prime contractor selects qualified software subcontractors.

Goal-2: The prime contractor and the software subcontractor agree to their commitments to each other.

Goal-3: The prime contractor and the software subcontractor maintain ongoing communications.

Goal-4: The prime contractor tracks the software subcontractor's actual results and performance against its commitments.

These goals fall beyond the current scope of the Rational Unified Process and are dependent on the organization.

While subcontracting is not specifically addressed in the Rational Unified Process, it is assumed that its tools, techniques, and mechanisms will be flowed down to subcontractors so the process remains homogenous.

All subcontracting decisions should be documented in the Business Case. Subcontractors following the same development plan as the prime contractor would also participate in the same technical interchanges, major milestones, and status assessments.

Software Quality Assurance

The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and the products being built. Software Quality Assurance is an integral part of most software engineering and management processes.

The Rational Unified Process considers quality to be the collective responsibility of all project staff and not embodied in any given organization per se.

Goal-1: Software quality assurance activities are planned.

Planning software quality assurance tasks is an organizational responsibility. However, the Rational Unified Process has a number of attributes that lend themselves to shaping an effective project quality assurance program.

Each Rational Unified Process milestone has specific completion criteria that can serve as a basis for auditing. Each activity within the Rational Unified Process has a separate review activity. Associated with each review is a set of checkpoints that represent "gates" that need to be "passed" before the following activity can be entered.

The Rational Unified Process provides guidance on who should review given artifacts. For example, the results of "Object Analysis", as performed by a Designer, need to be reviewed by an independent Architect, Designer, Use-Case Designer, and Design Reviewer. Given the defined Rational Unified Process and artifact review criteria, an objective body concerned with product quality should easily be able to assess adherence to process, and conformance to development standards and guidelines.

Goal-2: Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.

This goal would be met by adopting the organization's quality personnel. However, the Rational Unified Process provides the necessary review checklists and document templates that can be applied as project standards.

Goal-3: Affected groups and individuals are informed of software quality assurance activities and results.

As described in *Software Project Planning*, one of the Rational Unified Process goals is to ensure that the expectations of all parties are synchronized and consistent. Apart from any input from quality audit results, the Rational Unified Process calls on reporting on resources (staffing and financial), top ten risks, technical progress as gauged through metrics, and major milestone results. The Rational Unified Process metrics program provides guidelines on the collection of the following metrics:

- Progress (Lines of Code, classes, function points per iteration)
- Stability (type of rework, volatility)
- Adaptability (rework cost)
- Modularity (extent of rework impact)
- Quality (defect discovery rate, density, depth of inheritance)
- Maturity (test hours per failure)
- Expenditure profiles (planned versus actuals)

Goal-4: Noncompliance issues that cannot be resolved within the software project are addressed by senior management.

This falls beyond the scope of the Rational Unified Process and is an organizational responsibility. However, the Change Control Process described in the Rational Unified Process would enable a mechanism whereby noncompliances could be documented and escalated for resolution.

Software Configuration Management

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software lifecycle. Software Configuration Management is an integral part of most software engineering and management processes.

Goal-1: Software configuration management activities are planned.

As described in the Rational Unified Process, strong configuration management is an essential element in the controlled iterative development method. Since software is evolved in stages, it is vital that software versions from the preceding development effort be available for subsequent development. Planning how demonstrably working software is to be produced at each stage is at the core of the Rational Unified Process.

The Rational Unified Process has two major instruments for defining how a project's software development assets are to be maintained, and how they are to be integrated:

- the Configuration Management Plan
- the Integration Build Plan

The Configuration Management Plan, initiated in the Inception Phase, describes the following:

- managing software versioning and handling
- saving given Rational Unified Process models and dividing them into configuration items
- managing changes and releases using the change control method

The Integration Build Plan provides the details on configuration items that are to be built and the order in which they are to be integrated in a given iteration.

Goal-2: Selected software work products are identified, controlled, and available.

The Rational Unified Process Configuration Management Plan calls for a description of the configuration control and management process to ensure that work products are indeed identified, controlled, and available.

Goal-3: Changes to identified software work products are controlled.

The Rational Unified Process advocates that a project maintains a Change Control Board (CCB) and has a Change Management System to adequately manage, cost, track, and implement change requests.

Goal-4: Affected groups and individuals are informed of the status and content of software baselines.

The Rational Unified Process advocates that requirements, design and implementation baselines, and traceability between them, are maintained in electronic format. Various levels of project control arbitrate any changes to the baselines. For instance, the Change Control Board (CCB) considers the impact of requirement-level changes. Design and implementation changes that are smaller in scope are reviewed at the appropriate level of technical authority. Approval, control levels, and the way they are communicated are described in the *Configuration Management Plan* and in the *Software Development Plan*.

Level 3, Defined

Organization Process Focus

The purpose of Organization Process Focus is to establish the organizational responsibility for software process activities that improve the organization's overall software process capability. The primary result of the Organization Process Focus activities is a set of software process assets, which are described in Organization Process Definition. These assets are used by the software projects, as is described in Integrated Software Management.

Goal-1: Software process development and improvement activities are coordinated across the organization.

The Rational Unified Process is an iterative process that relies on the re-enactment of the same defined process over a number of iterations. This repetitive nature of process enactment, and the assessment of status metrics and lessons learned at each phase and iteration, provides an opportunity for fine-tuning the process for each successive iteration.

Goal-2: The strengths and weaknesses of the software process are identified relative to a process standard.

The Rational Unified Process represents an overall software development process that can be tailored for effective use on any given type of project. Guidance on how to tailor the Rational Unified Process is provided in the *Environment Workflow*. Other than technical and management complexity, some of the process discriminants that will determine the "shape" of the process to be used on a project are:

- business context (contract, speculative or internal)
- size of the software development effort
- degree of innovation
- type of application

Goal-3: Organizational-level process development and improvement activities are planned.

This Level-3 goal depends entirely on the adopting organization.

Organization Process Definition

The purpose of Organization Process Definition is to develop and maintain a useable set of software process assets that improve process performance across the projects and provide the basis for cumulative, long-term benefits to the organization. These assets provide a stable foundation that can be institutionalized via mechanisms such as training, which is described in the Training Program.

Goal-1: A standard software process for the organization is developed and maintained.

The Rational Unified Process can provide a head-start and serve as an organization's baseline software development process, which can be evolved, tailored, and maintained.

Goal-2: Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.

This goal would need to be supported by the organization that is adopting the Rational Unified Process.

Training Program

The purpose of the Training Program is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently. Training is an organizational responsibility, but the software projects should identify their needed skills and provide the necessary training when the project's needs are unique.

Goal-1: Training activities are planned.

This goal can only be met by the organization adopting the Rational Unified Process. However, the Rational Unified Process is an "industry best practices" knowledgebase providing guidelines, concepts, and detailed step-by-step descriptions on how to perform various software development activities. As such, the Rational Unified Process is itself a good source for training material.

Rational Unified Process-related support courses include:

- The Rational Unified Process Overview with modules on Requirements, Analysis & Design, Implementation, Test, Architecture, Process Configuration, Management, Tools, and an Introduction to Object Orientation
- Requirements Management with Use Cases (RMUC)
- Object-Oriented Project Management (OOPM)
- Object-Oriented Design and Analysis (OOAD)
- Automating Software Quality
- Configuration Management
- Software Architecture and Iterative Process

Goal-2: Training is provided for developing the skills and knowledge needed to perform software management and technical roles.

Goal-3: Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.

These training program goals would need to be met by the organization that is adopting the Rational Unified Process. However, the Rational Unified Process provides a range of courses, as described in the preceding section.

Integrated Software Management

The purpose of Integrated Software Management is to integrate the software engineering and management activities into a coherent, defined software process that is tailored from the organization's standard software process and related process assets, which are described in the section titled *Organization Process Definition*. This tailoring is based on the business environment and technical needs of the project, as described in Software Product Engineering. Integrated Software Management evolves from Software Project Planning, and Software Project Tracking and Oversight at Level-2.

Goal-1: The project's defined software process is a tailored version of the organization's standard software process.

In accordance with the Rational Unified Process *Environment Workflow*, the standard delivery of the Rational Unified Process is configurable and can be rescoped for use on various types of projects.

Goal-2: The project is planned and managed according to the project's defined software process.

This goal would need to be addressed by the organization adopting the Rational Unified Process.

Software Product Engineering

The purpose of Software Product Engineering is to consistently perform a well-defined engineering process that integrates all software engineering activities to produce correct, consistent software products effectively and efficiently. Software Product Engineering describes the technical activities of the project; for example, requirements analysis, design, code, and test.

Goal-1: The software engineering tasks are defined, integrated, and consistently performed to produce the software.

The Rational Unified Process activities and definitions of what is required by each role, against a backdrop of required project-planning artifacts, ensure that tasks are defined, allocated, and completed. The iterative development process inherent in the Rational Unified Process serves to quickly prove the effectiveness of the software development team and provides an assessment of the end product.

Goal-2: Software products are kept consistent with each other.

Traceability is maintained among the engineering models, such as use case models, design models, source code, and executable components, by the environment.

Intergroup Coordination

The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with other engineering groups so the project is better able to satisfy the customer's needs efficiently and effectively. Intergroup Coordination is the interdisciplinary aspect of Integrated Software Management that extends beyond software engineering; not only should the software process be integrated, but the software engineering group's interactions with other groups must be coordinated and controlled.

Goal-1: The customer's requirements are agreed to by all affected groups.

One substantial benefit of using use cases as a basis for requirements capture and description over other "formal" requirements specification methods is that use cases are readily understandable by involved stakeholders. As such, the Rational Unified Process method of capturing use-case requirements means that all stakeholders can agree on what needs to be done. This is further carried through the process, and is reflected in the models and reviews that are used as a basis for software development.

Goal-2: The commitments between the engineering groups are agreed to by the affected groups.

This goal would need to be addressed by the organization that is adopting the Rational Unified Process. However, Rational Unified Process visual models facilitate the understanding of what is required at each stage of product development—from requirements capture to deployment. The Rational Unified Process Change and Configuration Management process ensures that proposed changes are appropriately assessed and communicated to all stakeholders.

The engineering groups identify, track, and resolve intergroup issues. The Rational Unified Process iterative development process facilitates early detection of software problems through on-going integration of all developed software. Integration problems with software that is developed by a number of teams can serve as a "common space" for raising and resolving

cross-team issues. This notion is supported by the Rational Unified Process defect and change request process, which provides a formal mechanism for capturing, tracking, and resolving project development issues.

Peer Reviews

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently. An important corollary effect is to develop a better understanding of the software work products and of the defects that can be prevented. The peer review is an important and effective engineering method that is called out in Software Product Engineering.

Goal-1: Peer review activities are planned.

As described in the Quality Assurance goals for Level-2, each activity within the Rational Unified Process has a separate review activity.

Since early problem detection lowers overall costs, the Rational Unified Process advocates “early and often” peer reviews of all artifacts, particularly the critical ones. The Rational Unified Process provides checklists of salient features to review at each stage and within each model.

Goal-2: Defects in the software work products are identified and removed.

The Rational Unified Process artifact reviewers must determine whether the artifact is ready for the next stage of development. If the artifact fails to meet the review “pass” criteria then, in accordance with the Rational Unified Process metrics program, details need to be captured on:

- Stability (type of rework, volatility)
- Adaptability (rework cost)
- Modularity (extent of rework impact)
- Quality (defect discovery rate, density, depth of inheritance)
- Maturity (test hours per failure)
- Expenditure profiles (planned versus actuals)

References

- [REF1] Mark C. Paulk et al, "Key Practices of the Capability Maturity Model - Version 1.1", Software Engineering Institute—Carnegie Mellon University.



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 2000 Rational Software Corporation.

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