Case study on rational unified process

Business

The Rational Unified Process (RUB) is an iterative software development process framework created by the Rational Software Corporation, a division of MM. RUB Is not a single concrete prescriptive process, but rather an adaptable process framework, which Is tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs.

The product Includes a hyperlinked knowledge base with sample artifacts and detailed descriptions for many different types of activities.

RUB is included in theIBMRational Method Composer (ARM) product which allows customization of the process. RUB is based on a set of building blocks, or content elements, describing what is to be produced, the necessary skills required and the step-by-step explanation describing how specific development goals are to be achieved. The main building blocks, or content elements, are: Roles (who)- A Role defines a set of related skills, competencies, and susceptibilities.

Work Products (what) – A Work Product represents something resulting from a task, Including all the documents and models produced while working through the process. Tasks (how) – A Task describes a unit of work assigned to a Role that provides a meaningful result.

Wealth each Iteration, the tasks are categorized Into nine disciplines: six “ engineering disciplines” (Business Modeling, Requirements, Analysis and Design, Implementation, Test, Deployment) and three supporting disciplines (Configuration and Change Management, Project Management, Environment).

The use of these tools is not standardized and is subject to interpretation. Some shops may prepare report mock-ups. I have frequently seen these done in Excel. These could be part of the appendix to the requirements documents.

The requirements should be a description in non-technical terms (“ English”) of the business rules being implemented. These are considered detailed functional requirements. There should also be a validation plan, which will help the testing team develop test cases and scenarios. DESIGN The Design document references what you are going to build to meet the acquirement, and not how (Reap, 2005).

This Is described In broad terms: It can include pseudo code but not necessarily actual code functionality.

Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudopodia, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

At this phase the test plans are developed. The level of review under the test plan depends on the level of risk to the project. The project gets system architect approval If it is going into a legacy system, to ensure that the changes are not going to “ break” software already In place.

To launch the coding phase, develop a shell program that Is then put under some form of version control, for example Source Control Management from ASS. This apneas Includes ten set up AT a development environment, Ana use AT an enhance editor for syntax checking.

It is at this phase that development testing or unit testing occurs. Each developer insures that their code runs without warnings or errors and produces the expected results. User Acceptance Testing (TAT) is a second part of the acceptance phase, which is ideally conducted by an independent test group.

This includes the development of an Independent Test Plan put together by an Independent Test Team. Ideally these would be programmers, but often they are not. This phase verifies input/output and reviews the expected results.

The Test Plan includes development of test data with test cases and scenarios which exercise all logical paths. The results are the validation of the code. This phase is also where regression testing and sign off occurs and the test team verifies that the development outputs still match the production outputs where expected.