

PM WORLD TODAY – PM ADVISORY – OCTOBER 2009

Build your Project using Rational Unified Process

By Pavan Kumar Gorakavi

1. What is Rational Unified Process?

Rational unified process is a disciplined software development methodology which targets on producing high quality software deliverable. Rational unified process encourages systematic development while classifying tasks, assigning tasks and responsibilities within a development organization. Rational unified process was developed by Philippe Kruchten, Ivar Jacobsen, and others at Rational Corporation.

The setting for development of Rational Unified Process began in early 90's at Rational Software Corporation, founded by Paul Levy and Mike Devlin. Earlier Rational Software Corporation targets to develop large defense software development projects in Ada. In 80's, to withstand competition from different open-system platforms Rational Software Corporation decided in porting their system to open-system platforms. To with-stand suitable market place, Rational Software Corporation acquired Pure-Atria which provide clearCase configuration management tools, Purify line of testing tools, Requisite for Requisite Pro, SQA for testing tools suites. Rational worked with three Amigos [Grady Booch, James Rumbaugh, Ivar Jackobson] to develop a single unified language UML and drafts best software development practices which yields Rational Unified Process.

Rational Unified Process is a successor to the Rational Objectory Process. After merging of Rational Software Corporation and Objectory AB in 1995, Rational Objectory Process was the results of integrations of the Rational Approach and Objectory Process. Rational Unified Process inherits its process structure and use cases structure from Objectory Process. Rational Unified Process inherits its iterative development and architecture from Rational Approach.

There are lots of controversies about whether Rational Unified Process is agile or not. If Rational Unified Process is tailored, we can affirm RUP as an agile practice. Following salient features are to tailor made to make RUP as an agile process

- Choose only the most vital artifacts needed by the customer and the development team.
- Strive to keep iterations as short as possible.
- Business owners are part of the team
- Incremental short releases
- Collaborative and Communicative flow.
- Modularity on development model
- Adaptive with emergent risks

2. Principles of Rational Unified Process

The Rational Unified Process provides six guidelines to implement successful projects. These six best practices were developed from Rational's experience while developing large and complex systems. They were also designed to help drive the use of tools offered in Rational's product line. The designers of the RUP continue to evolve the process as methods and practices mature through their application [1]. The six basic principles are outlined below in figure 1.

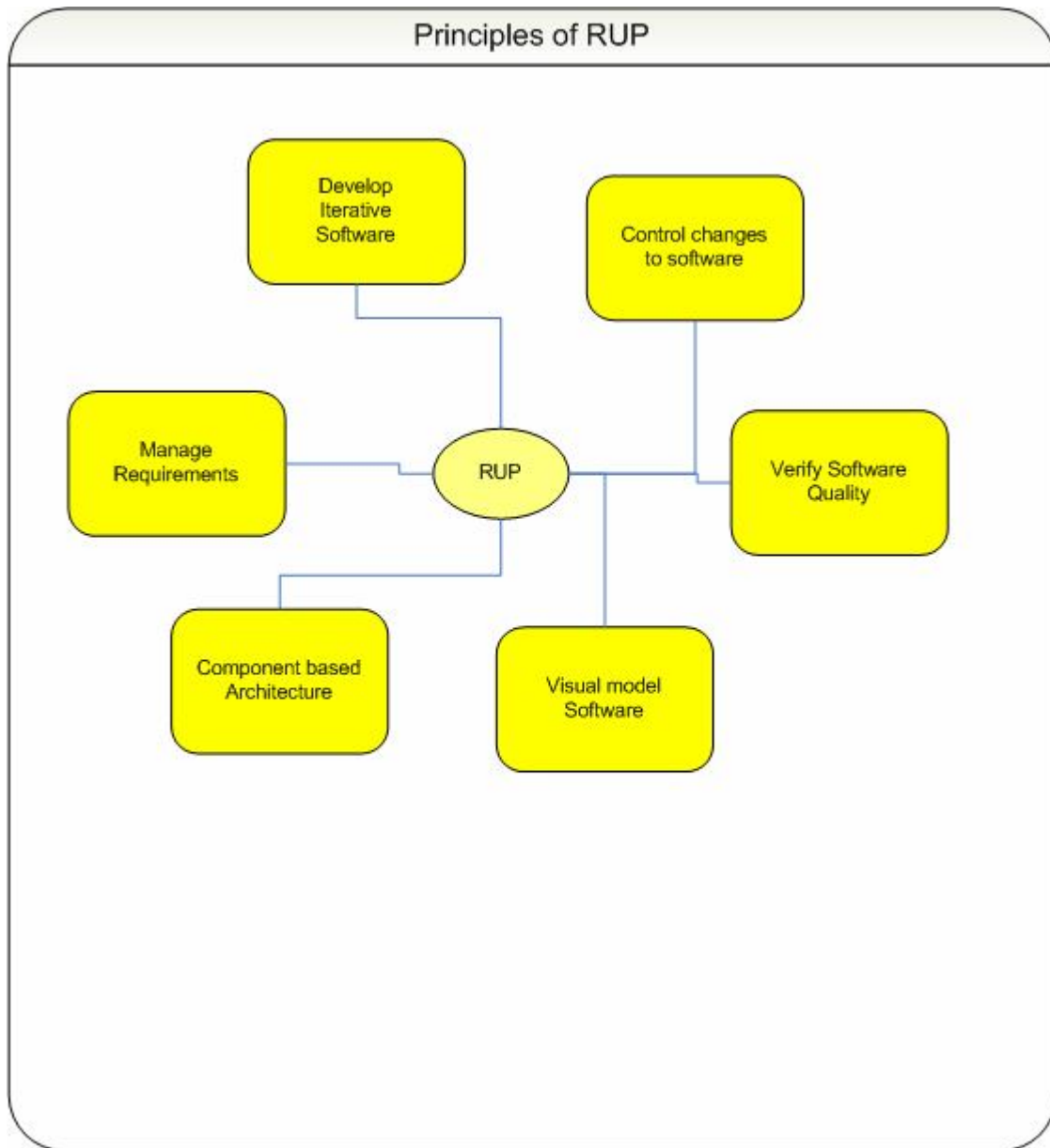


Figure 1: Principles of Rational Unified Process

The Rational Unified Process supports an iterative approach to development that helps in identifying risk proactively, reducing re-factor cost, and building models with easy exit strategy. Rational Unified Process recommends using use-cases and scenarios to capture functionality requirements. The Rational Unified Process supports component-based software development. Components are non-trivial modules, subsystems that fulfill a clear function. The Rational Unified Process provides a systematic approach to defining an architecture using new and existing components [2]. RUP encourages visual software models to depict architectures and component. Frequent verification of quality should be reviewed with respect to the requirements based on reliability, functionality, application performance and system performance. Control changes to software are recommended by the process. The process describes how to control, track and monitor changes to enable iterative development.

Per Kroll and Walker Royce updated the six best practices in IBM Rational e-zine , October 2005, as follows: [3]

1. Adapt the process.
2. Balance competing stakeholder priorities.
3. Collaborate across teams.
4. Demonstrate value iteratively.
5. Elevate the level of abstraction.
6. Focus continually on quality.

Every project is different in its appeal. Complexity, size, dependent variables, distributed team are some of the factors which impacts the project. Identifying dependent factors helps in adjusting the process to fit it. Project size must be optimal and must be decided based on the requirement. Process must be introduced slowly in initial part of the project and it must be implemented intensely at later parts of the project. Periodic reviews in RUP helps to identify risk proactively and continuously improve the process. Appropriation of process strength is dependent on project size, distributed teams, complexity, stakeholders and other dependent factors. More formal control is required when a project meets following criteria.

- Project members are distributed in different place
- User community is large and distributed.
- Large projects
- Many stakeholders

Defining and understanding business needs is an important aspect while implementing RUP. Key player's needs to identify the business needs and try to prioritize business and stakeholder needs. Per Kroll and Walker Royce argue strongly about having *center development activities around stakeholder needs*. Performing value analysis on leverage of assets, balancing user needs and reusing the assets are some of their suggested best practices. Collaboration across the team is well required to build a high efficient team. Per Kroll and Walker Royce mentions effective collaboration can be obtained by

- Motivating individual on the team.
- Encouraging cross functional collaboration.
- Integrated collaboration across business, software and operation team.
- Providing feasible environment for effective collaboration.

Iterative value based development is one of the best practice. Delivering incremental value helps in obtaining early and continuous feedback. This early feedback helps in reducing re-factoring cost. Per Kroll and Walker Royce argue to elevate the level of abstraction by reusing existing assets. Reusability can be impacted by system complexity, loose coupling, and cryptic architectures. Reusability can be obtained by having modular design, and simple architecture.

3. Life Cycle of Rational Unified Process

According to Kruchten, RUP consists of four sequential processes during design and development of business solution. The four sequential processes includes: Inception, Elaboration, Construction and Transition. These phases are iterative in nature and yields products in each field. Iterations are between two weeks to six months. RUP lifecycle is illustrated in figure 2.

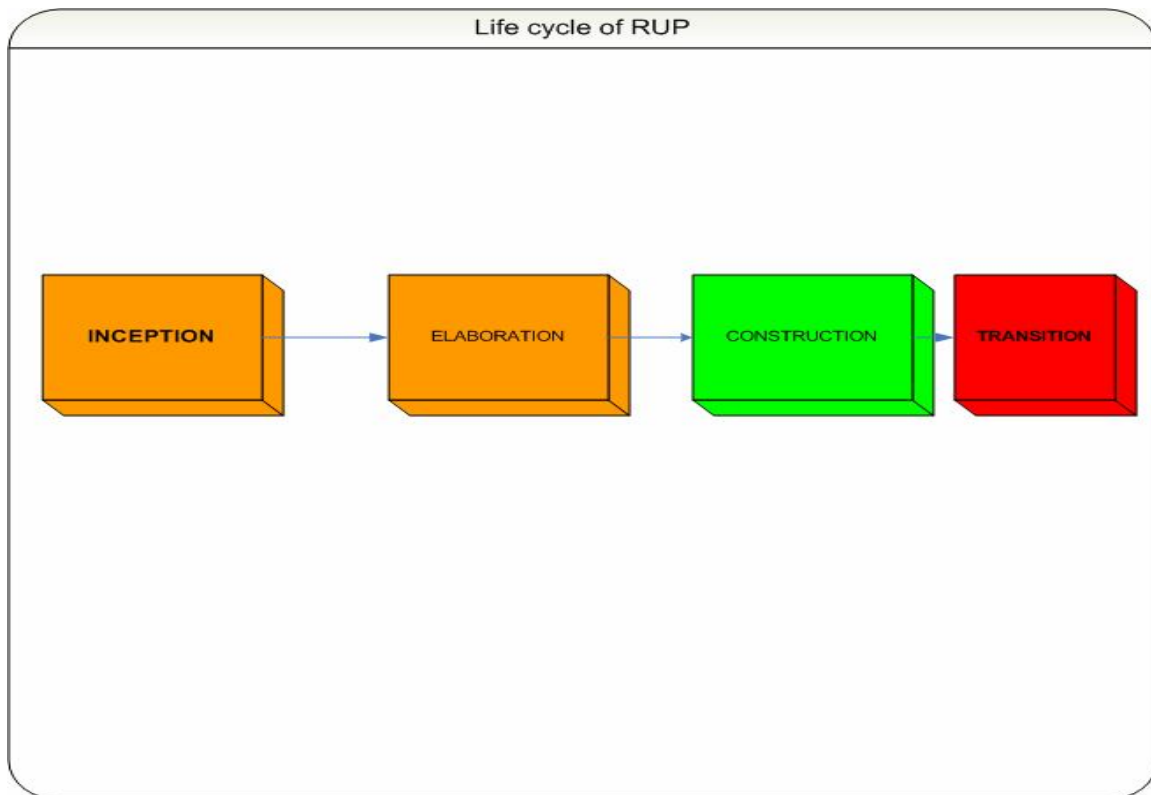


Figure 2: Life Cycle of RUP

Phase 1: Inception

In this phase overall domain descriptions are prepared. Document requirements are organized with proper use-cases and technical specifications. During this phase business case for the system is established and delimited the project scope. All key entries are identified and coupled communication is identified at higher level. Identify use cases is an important step in this phase. Business case includes success criteria, risk analysis, resource requirements, and game plan to be implemented.

Inception phase yields use-cases, vision documents, high level business case and glossary, higher level risk analysis, project plan, and prototype. Evaluation is performed iteratively in each phase. Stakeholders conducts regular meeting to discuss about scope definition and cost estimates. Conflict of interests and initial risk assessment are resolved while stakeholders are in process of having concurrence among team. Cost estimates, Risk factors, task estimations, resource estimations and schedule estimates are evaluated in these meetings. Prototypes developed in phase-1 are evaluated by its depth and breadth. Relative analysis of actual expenditure vs. planned expenditure is performed as part of evaluation criteria. Project can be rethought or cancelled or proceed in this phase.

Phase 2: Elaboration Phase

This phase plays a vital role in laying foundation for project architecture. This phase primarily helps in analyzing problem domains, developing flexible project plan, developing flexible architectural foundation, identifying risk and eliminating them. Architectural foundation can be performed from results of phase-1, i.e scope definition, requirements, and major high level functionality. This phase is helpful in identifying risk analysis, and value analysis and decision can be taken whether or not to commit to further phase: construction and transition phase. Executable architecture is built in iterations depending on project size, scope, cost, and time lines.

After this phase , most use cases are defines, actors are identified, Software architecture is described, and a prototype is created. Evaluation is performed based on vision of the product stability, architecture stability, relative analysis of actual resource expenditure vs. planned expenditure, and risk factors identified. The project may be aborted or considered based on the result of milestones achieved in this project.

Phase 3: Construction Phase

In construction phase all remaining components and application features are developed and integrated in the product. All developed features are thoroughly tested. Construction phase is a manufacturing phase where focus should be on managing resources, controlling operations, optimizing cost, meeting timeline, and maintaining acceptable quality. Successful elaboration phase yields successful construction phase. A robust architecture developed in elaboration helps as skeleton for construction phase.

There are scenarios where parallel construction increments can be spawned.

Construction phase results in software products integrated on adequate platform, User manual, release notes, and other proprietary documentations. The results of the construction phase: either alpha, beta, are developed iteratively and rapidly which helps in developing high quality product with rapid pace. Construction phase is evaluated considering following salient features.

- Stability of the product released.
- Maturity of the product developed.
- Stakeholders acceptance
- Relative analysis between actual resource expenditure vs. planned expenditure.
- Can be deployed in user community?
- Does user documentation stands as per industry standards.
- Release notes has up-to-date information
- Legal liability of the finished products.

Phase 4: Transition Phase

This phase is to transition the software deliverables to the user community. This phase is attained when the product reach suitable maturity and stability. As the product development is iterative in nature, defects are resolved in subsequent releases. This phase primarily focus on activities required to release stable products, product tuning, product configuring and resolving issues. The primary objective of the transition phase includes achieving baseline targets, user confidence and self supportability, and concurrence with vision documents.

Transition phase is evaluated based on following salient features.

- User satisfaction
- Relative an analysis between actual resource expenditure vs. planned expenditure.
- Self supportability of the product
- Value analysis based on cost effectiveness, and product baselines
- Acceptance based on deployment baselines.

4. Different Actors of RUP

RUP process can be classified into three main components: Activities, Artifacts, and workers. An activity is a unit of work that an individual is asked to perform in that role.

It can also be described as a granular, modular functionality which a role expects. Activity can vary from few hours to few days. Finding actors, developing modular functions, writing use cases are some of the examples of Activities.

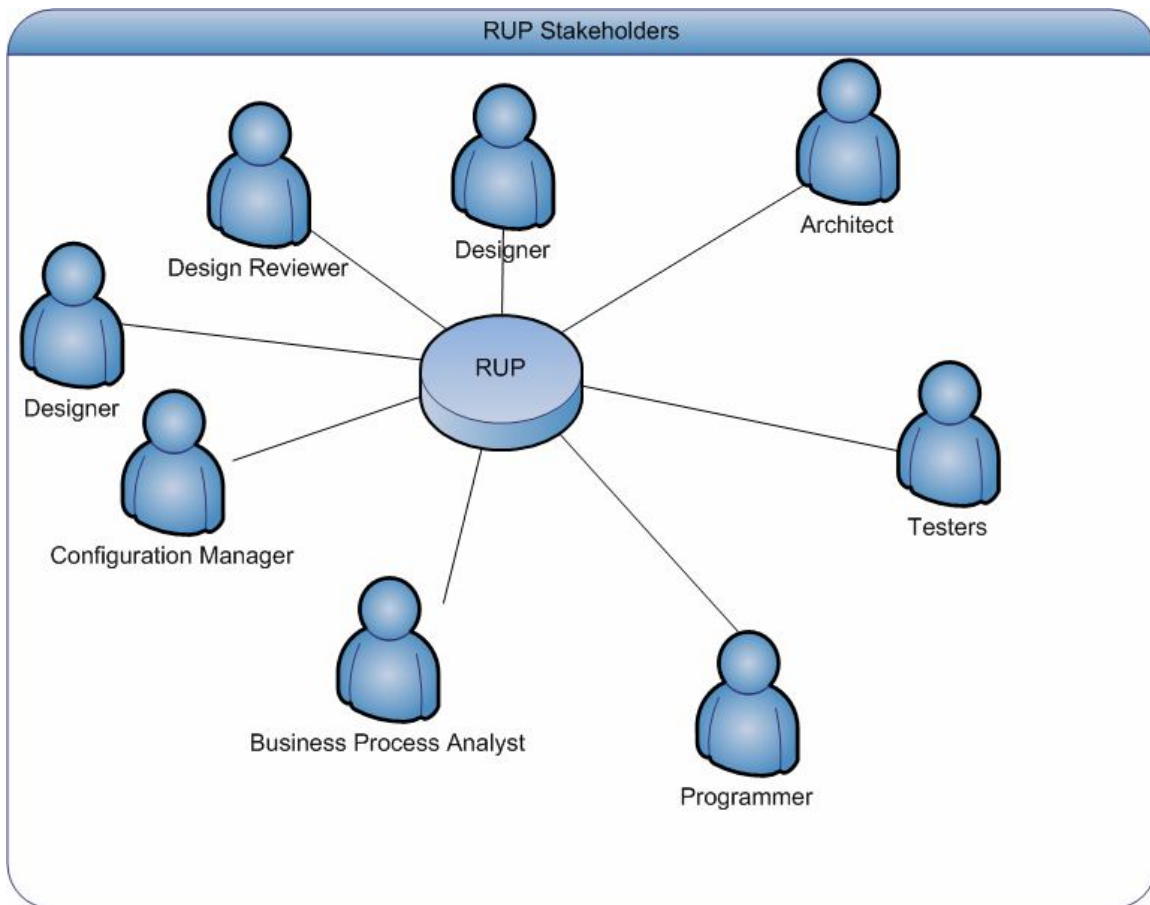


Figure 3: RUP stakeholders

An artifact is a information center which is developed, modified or used by a process. Activities are derived from artifacts. Artifact can be in various shapes. It can be in use case model or design model or business case or software architecture documents or executables or source code. A worker defines responsibilities of an individual. It designates a person role with a description or title. In RUP workers are role defining individuals. A workflow is a meaningful sequence of activities that produce some valuable result. Workflow is a sequence of activities that yields a value based result. Workflows are generally expressed as sequence diagrams, collaboration diagrams, or activity diagrams. A sample workflow is illustrated in figure 4.

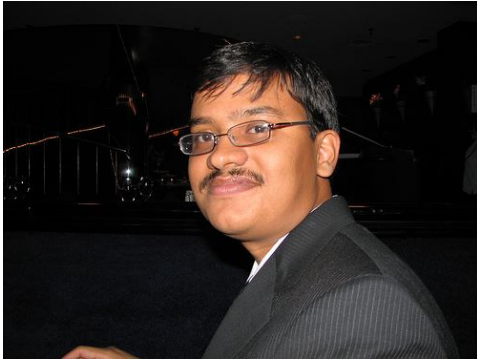
RUP is an activity driven development. Roles are basically defined based on the activities defined in the process. RUP has workers like developers, architects, designers, design reviewers, configuration managers, business process analyst, business reviewers, business designers, course developers, toolsmith , testers, project managers, stakeholders etc.

5. Conclusion

According to Kruchten (2000), rational unified process can be often adopted on a whole or in parts. RUP was successfully implemented across many organizations and it was well supported by rational tools. RUP can be tailor made to make that process agile. RUP process based approach helps in developing a robust system.

References:

- [1] Dennis Gibbs, R.. "Chapter 2 - Overview of the Rational Unified Process". Project Management with the IBM Rational Unified Process: Lessons From The Trenches. IBM Press.
- [2] Rational Unified Process Best Practices for Software Development Teams, http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- [3] <http://www.ibm.com/developerworks/rational/library/oct05/kroll/index.html>
- [4] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, Juhani Warsta. Agile software development methods, VTT Publications.
- [5] Péraire, Cécile. The IBM Rational Unified Process for System z. IBM Redbooks
- [6] Dennis Gibbs, R.. Project Management with the IBM Rational Unified Process: Lessons From The Trenches. IBM Press.
- [7] Kruchten, P. The Rational Unified Process: an Introduction. Addison-Wesley

About the Author:***Pavan Kumar Gorakavi****Author*

Pavan Kumar Gorakavi. M.S, M.B.A, G.M.C.P, C.A.P.M is currently working as

Sr. Software Developer in Dallas, TX. He is settled in Dallas with his small family [wife Swapna Gorakavi and son Anish Gorakavi]. He is VP - programs for asapm-young crew. Pavan did his bachelors in computer science from Jawaharlal Nehru Technological University in India and masters in computer science from Lamar University. He did his MBA from University of Texas at Dallas and GMCP from Southern Methodist School in Dallas. Pavan holds SUN, IBM and PMI certifications. Pavan Gorakavi is the author of books on '*Artificial Intelligence*' published by Rahul publications - India, and '*Digital Electronics*' published by Subhash publications, India. His research interests are Artificial Intelligence, Agile methodologies, and Software development in Ada, Prolog and Java. You can reach Pavan at gorakavi@gmail.com .