Tailoring RUP to Shataayu Project: A Case Study

Fatema Rangwala, Kirti Mathur, Taher Ali Rangwala, Ashwini Mathur
Devi Ahilya Viswavidhyalay, Vellore Institute Of Technology.

Abstract – For developing a project the framework which is widely used is Rational Unified Process (RUP). RUP is used widely due to its core principles i.e. architecture centric, uses use case model for user documentation and is an iterative and incremental process model. RUP defines many artifacts and roles which are very useful during software development. But RUP does not fit for all software’s to be developed as there are many different type of software with different requirements. Therefore RUP as defined cannot be used in some projects. Therefore it has to be tailored. In this case study we are representing a project in which how RUP is tailored is shown. Research method used in this case study is action research and for data collection postmortem analysis is done. In which all the team members evaluate the project being developed and project’s flip side is also discussed.

Keywords – RUP (Rational Unified Process), UML (Unified Modeling Language), action research, postmortem analysis.

I. INTRODUCTION –

RUP is a software engineering process widely used by software developers. It is an iterative software development process framework based on UML created by the Rational Software Corporation, a division of IBM since 2003[1]. UML (Unified Modeling Language) is a standardized general purpose modeling language in field of software engineering[2]. Rational Unified Process (RUP) is a software engineering approach to the assignment of tasks and responsibilities within a development organization. Its goal is to deliver the software on time with assured quality and within affordable cost. RUP is modified and customized in accordance with project but it does not deal with issues like how to provide security to a developing software. RUP deals with “HOW” and not with “WHAT” i.e. it shows how to do and not what to do. RUP is an iterative software engineering process, though it is not a well defined process model instead it is an adaptable process framework which can be tailored or customized by an organization according to different application. RUP is a disciplined, iterative approach used to assign task and responsibilities to software development team.

In 1995, the Standish group published the first version of the CHAOS report on software projects effectiveness. This report showed the main reasons for projects being in trouble: lack of user involvement, low quality of requirement specification, and changing requirements. RUP was proposed by its creators and advocates as a software development process that intends to address these sources of problems. However, it has also been argued that RUP has some strong
limitations. For example, it is argued that RUP lacks of a truly “architecture centric” approach to software development, it is said to be unnecessarily complex, inadequate to structure complex software processes, does not properly support project management and fails short in achieving adequate users feedback. Hence, the question of the advantages and disadvantages of RUP is still an open discussion in software engineering [3]. Security is also a big issue nowadays in every software but in RUP the security is limited to only initial phases of development during the identification and documentation of supplementary requirements. But only in the initial phase the threats cannot be identified completely. No security plans exists thereafter for finding malicious objects within a software. In RUP, software engineering processes are organized into phases and iterations[4]. It comprises of six best practices which are: develop iteratively, manage requirements, employ a component based architecture, model software visually, continuously verify quality and control changes. Four Phases of RUP are Inception Phase in this phase initial cost, budget and success factor is forecasted. Use case model, project plan and initial risk assessment is generated. Elaboration Phase this phase focuses on risk items and architecture of project. Construction Phase objective is to develop components and other features of the project results in deliverable output. Transition Phase in this the software transits from producers to end users. End user training, maintenance and testing is done. There are nine disciplines within each iteration which describes unit of work assigned to a role. Six engineering disciplines are - Business Modeling is used for enterprise process analysis, improving process efficiency and quality. Requirements deals with functional, quality and performance requirement which collectively defines what system should do. Analysis And Design Transforms requirement to system design using use cases and models defined by UML (Unified Modeling Language) defines object characterized by its class to develop comprehensive architecture adapting design of system for performance. Implementation Classes and objects are implemented in terms of executables, source files and others. Components are tested and integrate to produce executable system. Test and Deployment where test verify the implementation correctness. Any defects encountered is either fixed or elicited. Deployment refers to custom install and end user can access the system. Three supporting disciplines Generally are – Configuration and Change Management discipline deal with identifying and managing configuration item and control changes demanded and change version. Project Management discipline focuses on monitoring project by planning, staffing and risk management. Metrics and progress of iterative projects are also monitored. Environment discipline aims to configure the process and provides tools and Supporting processes to software development as shown in Fig. 3. [5]. How all this disciplines fit in our project is shown in next section.

II. RESEARCH METHOD –
As we were also involved in this project development, therefore the research method
used is action research. Action research is simply a form of self-reflective enquiry undertaken by participants in social situations in order to improve the rationality and justice of their own practices, their understanding of these practices, and the situations in which the practices are carried out (Carr and Kemmis 1986: 162). Action research is an cyclic process, the steps involved are –

![Figure 2. The cyclic process of action research [6]](image)

Before we describe the phases of action research, let us have a look at research context.

A. Research Context –

The company whose project has been undertaken in this case study name is not revealed due to some reason. At the time of this project the employee number of that company were 30. Company used to undertake both public sector project and private sector project. They have made many public sector projects. The projects were developed under guidance of two project leader. The projects developed were desktop application as well as web application projects. The technology used were Java, J2EE, C#, PHP. The technology used for project been discussed here is PHP, it is a web application, with strong back end logic and good user interface. The project was developed for a private doctor clinic to fix and cancel appointments and for doctor to keep track of the number of people in queue waiting at his/her clinic for checkup turn. RUP mentors were appointed for training developers to develop project according to RUP. But while using RUP the developers faced some problems. The solution of which was found after tailoring RUP according to our project requirements.

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Project Type</th>
<th>Team Size</th>
<th>Project Size</th>
<th>Process Model Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Mentioned</td>
<td>Desktop Application</td>
<td>10 team members</td>
<td>Small Project developed in 20 days.</td>
<td>RUP(Rational Unified Process)</td>
</tr>
</tbody>
</table>

Table 1. Representing company and project details.

B. Identify or diagnose the problem phase

In the diagnosing phase of the project, the training needs are determined for persons with different disabilities, abilities, learning styles and literacy levels. This research is conducted in partnership with training specialists, educators, educational, kinesiologists, and specialists in the disability domain[7]. In this case study team leader, project manager, project leader, project developer with different experience were interviewed.

a) The interview general experience –
This interview was conducted to find general area of problem and finding how well RUP artifact and roles fit while developing the project. RUP disciplines one by one were examined for that. Let's have a look on each discipline-

- **Business Modeling** – It is used in this project for understanding the need of the doctor for this project, the problems faced and the possible solution. Roles defined by RUP in business modeling are **business process analyst**, **business designer** and **business model viewer**. Of which business process analyst establishes business actors and use cases and also show how they interact. Business designer describe the workflow of use cases, decides team size and members and realize use cases. Activities mentioned in RUP are **assess target organization**, **set and adjust goals**, **maintain business rules**, **define business architecture**, **capture common business vocabulary**, **find business actors and use cases**, **structure business use case model**, **detail business worker**, **find business workers and entities**, **detail business entity**, **define automation requirement**, **review business use case model** and **review business object model**. Of which applied in our project were **assess target organization for understanding doctor need and environment of clinic**. Find business actor and use case to identify boundaries, customers for which it is developed and use case diagrams are developed in it and structure business use case model. Artifacts defined by RUP are **business glossary**, **business rules**, **business use case model**, **business object model**, **target organization assessment**, **business vision**, **business architecture document**, **supplementary business specification**, **business use case**, **business use case realization**, **business actors**, **business entity** and **business worker**. Of which we used were business use case and business use case realization.

- **Requirement Discipline** – It helps in eliciting requirement from stakeholder and developing software accordingly. The roles defined by RUP in it are **system analyst**, **software architect**, **requirements specifier**, **requirement reviewer** and **user - interface designer**. Roles required in our project were system analyst to outline project functionality, software architect coordinates technical activities and artifacts, user interface designer designs the user interface, requirement specifier details use case and doctor’s requirements. Activities specified in RUP are **develop requirements management plan**, **develop vision**, **elicit stakeholder requests**, **capture common vocabulary**, **find actors and use cases**, **manage dependencies**, **structure the use case model**, **prioritize use cases**, **detail use case**, **detail software requirements**, **model the user interface**, **prototype user interface**, **review requirement**, **used were develop management requirement plan for documenting requirements**, **develop vision is used for identifying the problems to be solved and describes feature and boundaries of the website**, **capture a common vocabulary describes the vocabulary used especially in use**
case, find actors and use cases is used for define functions of the website, use case model diagrams are created and defines website is being used by whom, model the user interface is used for building model of user interface and review requirements is used for verifying that requirement result conform user’s view. Artifacts are requirement management plan, stakeholder requests, glossary, vision, use case model, supplementary specification, use case, use case package, software requirement specification, user interface prototype and use case storyboard. Artifacts used are requirement management plan which is the output artifact of activity develop requirement management plan, glossary it is the output of activity capture a common vocabulary, Vision defines the stakeholders view of the product to be developed, use case model defines system’s intended functions, supplementary specification it captures specification other than use case like usability, reliability. Software requirement specification consists of a package containing use cases of the use-case model and applicable supplementary specifications.

- Analysis and Design – In this discipline requirements are converted to designs. Roles defined are software architect, capsule designer, designer, database designer, architecture reviewer, design reviewer. Used in our project were software architect, designer, database designer handles all back end logic. Activities are architectural analysis, identify design mechanisms, identify design elements, construct architectural proof of concept, incorporate existing design elements, describe run-time architecture, describe distribution, capsule design, use case analysis, use case design, sub system design, class design, design test classes and packages, database design, review the architecture and review the design. Activities used are architectural analysis used for planning and defining reuse strategy, identify design mechanism to refine analysis mechanisms into design mechanisms, incorporate existing design elements analyze interactions of analysis classes to find interfaces, design classes and design subsystems, use case analysis and use case design describes use case related things, database design ensures database entries to be stored consistently, review the architecture is done to identify any architectural flaws, review the design to ensure that the design guidelines fulfills the requirements. Artifacts are deployment model, software architecture document, analysis model, design model, reference architecture, interface, signal, event, protocol, use case realization, analysis class, design class, design subsystem, design package, capsule and data model. Used are deployment model, software architecture document provides a comprehensive architectural overview of the system, analysis model describes the realization of use cases, design model is used as essential input to activities in implementation and test, interface defines set of
operations, use case realization describes how a particular use case is realized within the design model and data model describes the logical and physical representation of persistent data in the system.

- Implementation – It defines the organization of code, test and integrate developed components. Roles defined are software architect, implementer, integrator and code reviewer. Software architect coordinates technical activities and artifacts throughout the project, implementer is responsible for developing and testing components, implementers deliver their tested components into an integration workspace whereas integrators combine them to produce and build, code reviewer ensures source code quality plans and conducts source code reviews. Activities defined are structure the implementation model, implement component, fix a defect, implement test components and subsystem, perform unit tests, plan system integration, integrate subsystem, integrate system and review code. Used are structure the implementation model establishes the structure in which the implementation will reside, implement component produce source code in accordance with the design model, fix a defect for fixing defect occurred, implement test component and subsystem implements test-specific functionality, perform unit tests to verify internal structure of unit, plan system integration plans the integration of the system, integrate system to integrate the implementation subsystems piecewise, review code to verify the code. Artifacts are component, implementation subsystem, integration build plan, build, implementation model. Used are component represents a piece of software code or a file containing information, integration build plan summarizes of detailed integration plan within an iteration, build is an operational version of a system or part of a system, implementation model is a collection of components like executables and source file.

- Test – It is done for assessment of quality. Roles defined are test manager, test analyst, test designer and tester. Test manager has the overall responsibility for test effort’s success. Test analyst defines the required tests, test designer defines the test approach and ensures its successful implementation and tester is responsible for the core activities of the test effort. Activities defined are agree mission, identify test motivators, obtain testability commitment, assess and advocate quality, assess and improve test effort, identify targets of test, identify test ideas, define test details, define assessment and traceability needs, determine test results, verify changes in build, define test approach, define test environment configurations, identify testability mechanisms, structure the test implementation, define testability elements, develop test guidelines, implement test, implement test suite, execute test suite and analyze test failure. Agree mission results in artifact test plan, assess and advocate quality helps in defects
resolution, assess and improve test effort is used for making an assessment of the productivity, effectiveness and completeness of the test effort, identify targets of test identifies system elements that need to be tested, define assessment and traceability needs to identify assessment strategy and traceability requirements, determine test results evaluate product quality and if encounter any defect in quality propose corrective actions, define test approach for identifying techniques for testing, define test environment configuration defines requirements for the evaluation environment to support test effort, structure the test implementation defines the required test suites, implement test is used for implementing tests to provide required product validation, implement test suite is used for assembling collections of tests to be executed together, execute test suite is used for executing test suite for obtaining test results, analyze test failure details and analyze the failures that occurred during test implementation and execution, Artifacts are test plan, test evaluation summary, test script, test log, test ideas list, Test case, workload analysis model, test data, test results, test automation architecture, test interface specification, test environment configuration, test suite, test guidelines, test class and test component. Artifacts applied are test plan defines item targeted, approaches taken, resources required and deliverables produced, test evaluation summary provides general statement of relative quality and recommendations for future test effort, test log represents the output resulting from the execution of a Test Suite, test case specifies set of test inputs, execution conditions, and expected results, for evaluating some aspect of a Target Test Item, test result is a collection of summary information determined from the analysis of one or more Test Logs and Change Requests helps in quality assessment, test environment configuration specifies hardware, software and associated environment setting required for testing and evaluating target test items, test suite is a collections of Test Scripts, both for execution of the tests and to provide a useful and related set of Test Log information from which Test results can be determined.

- **Deployment** – It concerns with software product delivery, Roles specified are deployment manager, course developer, implementer, graphic artist and technical writer. Used are : deployment manager plans the product's transition to the user community and its associated documentation, implementer develops and test components according to project’s standard and integrate it into larger sub system. Activities are develop deployment plan, manage acceptance test, provide access to download site, verify manufactured product, define bill of materials, manage beta tests, release to manufacturing, write release notes, develop training materials, develop installation artifacts, create product.
artwork, develop support materials. Applied are Deployment plan documents how and when the product is to be made available to the user, manage acceptance test ensures that the developed product fulfills its acceptance criteria at both the development, and target installation site, verify manufactured product ensures that the manufactured product is complete, and useable, define bill of material is a complete list of artifacts that will be required to make up the build/product, manage beta test is a pre release test for testing the end product, develop installation artifact Produce all the software required to install and uninstall the product quickly, easily and safely without affecting other applications or system characteristics and develop support material develops the end user support material. Artifacts are deployment plan, bill of materials, release notes, product, installation artifacts, training materials, deployment unit, product artwork and end user support material. Deployment plan is output from activity develop deployment plan, bill of material is output from activity define bill of materials, installation artifact is a output activity of develop installation artifact, end user support material is output activity from develop support material.

- Environment - Focuses on the activities necessary to configure the process for a project. Roles specified are process engineer, tool specialist, user interface designer, system analyst, business process analyst, software architect, test designer, technical writer and system administrator. Process engineer deal with software development process, continuously evaluate process and improve it, tool specialist selects and acquires tools, user interface designer builds user interface prototype, system analyst ensures what actors and use cases exist and how they interact, software architect coordinates technical activities and artifacts, test designer defines test approach and ensures it’s successful implementation. Activities defined in RUP are assess current organization, develop development case, develop project specific templates, launch development case, select & acquire tools, setup tools, develop toll guidelines, verify tool configuration and installation, develop user interface guidelines, develop use case modeling guidelines, develop business modeling guidelines, develop design guidelines, develop programming guidelines, develop test guidelines, develop manual style guide and support development. Develop development case describes website development process, select and acquire tools deal with selecting and acquiring tools for our website development, set up tools deal with setting up tool at client side like in our project xamp server was installed at client side for database. Artifacts defined are development case, development organization assessment, project specific templates, business modeling guidelines, design guidelines, program specific guidelines, use case modeling guidelines, user interface guidelines, test guidelines, manual styleguide, tool guidelines, tools and development infrastructure.
Development case is the output from activity develop development case, Tools is the output from the activity set up tools and select & acquire tools.

- Project Management : It provides a framework for managing software project and their risks, guidelines for planning, staffing, executing, and monitoring projects. Roles are project manager and project reviewer. Used are project manager allocates resources, shapes priorities, coordinates interactions with customers and users, and generally keeps the project team focused on the right goal. The project manager also establishes a set of practices that ensure the integrity and quality of project artifacts. Activities defined are develop business case, initiate project, identify and assess risks, define monitoring and control processes, plan phases and iterations, compile software development plan, define project organization and staffing, develop quality assurance plan, develop measurement plan, develop product acceptance plan, develop problem resolution plan, develop risk management plan, develop iteration plan, acquire staff, initiate iteration, assess iteration, prepare for phase close out, prepare for project close out, monitor project status, schedule and assign work, report status, handle exceptions and problems, project approval review, project planning review, iteration plan review, PRA project review, iteration evaluation criteria review, iteration acceptance review, lifecycle milestone review and project acceptance review. Identify and assess risks identify, analyze and priority risks to determine appropriate risk management strategies, plan phases and iterations estimate scope, cost and effort for the project, develop resource plan, budget, schedule and project plan defining goals to be achieved, acquire staff in this human and available resources are collaborated according to project requirements, initiate iteration starts iteration by allocating staff and other resources defined for project, assess iteration identifies failures and success of the project and any improvement identified, prepare for project close out formalities associated with project acceptance and close-out, reassign project staff, and transfer other project resources, monitor project status captures the current status of project against decided plan, lifecycle milestone review in this project state is reviewed at end of the phase to conclude whether next phase should be initiated or not. Artifacts defined are software development plan, business case, iteration plan, iteration assessment, status of assessment, problem resolution plan, risk management plan, risk list, work order, product acceptance plan, measurement plan, quality assurance plan, issue list, project measurements and review records. Artifacts applied are iteration plan is a time sequenced set of activities and tasks, with assigned resources, containing task dependencies, for the iteration, iteration assessment captures the result of the iteration it is then evaluated for if any change encountered.
Configuration and Change Management: In this configuration items are identified and changes made due to them are managed and restricted. Roles defined are configuration manager, change control manager, integrator and any role. Integrator integrates the tested subsystem to form a unit. Activities defined are set up CM environment, establish CM policies, write CM plan, create deployment unit, report on configuration status, perform configuration audits, establish change control process, review change request, confirm duplicate or rejected CR, create integration workspaces, create baselines, promote baselines, create development workspaces, make changes, deliver changes, update workspace, submit change request and update change request. Set up CM environment creates a baseline for further development by allocating hardware and drivers required for project development, write configuration management plan describe all CM related activities to be performed during the course of the project lifecycle, create baselines ensures that all developed artifacts are captured and archived, at given points in time, as a basis for further product development, promote baseline purpose is to ensure that baselines (individually tested components from various implementers, and development teams, combined together to work together as a product) and then further development is done. Artifacts defined are configuration audit findings, configuration management plan, project repository, change request, workspace (integration), workspace (development). Used are configuration audit findings in which baseline, incompletely tested or failed requirements are identified, configuration management plan details the schedule of activities, the assigned responsibilities, and the required resources, including staff, tools, and computer facilities, project repository stores all the derived data and meta data associated with the files and directories, in integration workspace the workspace is shared by all team members.

b. RUP experiences –

Our project team was small comprising of 15 members, some members were playing more than one role in project development. The roles were developer, project leader/developer/requirement manager, tester. All team members were interviewed about their experience using RUP but none of them have deep thorough knowledge of RUP. Some of them have little experience using RUP but others have just read in literature only. Some people said that RUP is not useful where fast delivery of project is required as it is too document driven and in case of small project it is difficult to distinguish between the activities and artifacts defines as most of them resemble same thing. Some people said that use of software process model is defined according to the client need or their requirements specified. Some said if customer requirements changes frequently then RUP is not better process model to use. The problems specified with RUP usage may be due to lack of RUP knowledge, for this better RUP training should be given to developers for encouraging them to use RUP and if required some tailoring should also be done for meeting their requirements. On the basis of the interview
C. Gather Data –

In this case study we have used postmortem analysis as a tool for data gathering. Dingsøyr’s defines [46], retrospective analysis as a: “collective learning activity, which can be organized for projects either when they end a phase or are terminated. The main motivation is to reflect on what happened in a project in order to improve future practice - for the individuals that have participated in the project and for the organization as a whole.” It is used as a process improvement approach by collecting the positive and negative feedback from each team member using RUP so that it can help in future projects and also help developers for choosing better software development process which lead them to a successful project development. The results obtained are documented in the form of a PMA-report that shows experiences using RUP. During the PMA - sessions, the researchers took personal notes during the work sessions. These notes are also treated as data since they give valuable information about the Post Mortem analyses [8]. The PMA is done through process workshop which was conducted in our office for half a day and all team members experience using RUP were noted. The things that went wrong were listed to find corrective measures for that. The things that went right was also listed with most beneficial thing with high priority.

<table>
<thead>
<tr>
<th>Questions asked in Process Workshop</th>
<th>Developing this software</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is RUP usage beneficial in this project.</td>
<td>3. Do you all have thorough knowledge of RUP.</td>
</tr>
<tr>
<td>2. Do you think coverage of all the phases of RUP are essential for developing this software</td>
<td>4. Does this organization has appointed any RUP mentor.</td>
</tr>
<tr>
<td></td>
<td>5. What benefits you get using RUP.</td>
</tr>
<tr>
<td></td>
<td>6. Do you feel the need of tailoring RUP and if yes then why.</td>
</tr>
<tr>
<td></td>
<td>7. What are the pros and cons using RUP.</td>
</tr>
</tbody>
</table>

Table 2. Showing the questions used to gather data.

D. Interpret data –

In this the data collected using post mortem analysis is analyzed to find a solution for negative feedback. The report made during PMA sessions play a vital role in research and data analyses. The PMA reports in this case study follow a template which ensure that both positive and negative experiences from the project are documented [9]. PMA helps company to analyze what worked well and what had gone wrong during software development which will help team members to create a baseline for other projects by avoiding the things that gone wrong previously. The list created in process workshop is analyzed minutely to find solutions of the problem occurred.

E. Act on Evidence –

On the basis of data collection and analyses what action should be taken to optimize the result is planned, like what changes should be taken to get better result and what changes done can decrease the performance is planned in action planning. The company being researched has mainly handled small size project to medium size project, domain
changed frequently, software development process model vary according to project as the project leader told us, but they mainly use RUP for project development by using use cases and UML. Process workshop was conducted to get a in depth view of what process model is being used and how or to what extend it is implemented. In process workshop team members, project leader, project manager were involved and how they decide which process model to use is investigated. Investigated things were project type, roles defined, models used for training purpose, technology being used, literacy level of people to understand language defined and their learning styles. Based on this investigation actions are taken for developing prototype and test models.

F. Evaluate Result –

Through workshop experience of each and every team member is recorded of using RUP. Though they don’t have thorough knowledge of RUP, but through their working experience they all find that RUP requires too much documentation which is very inconvenient for small project. Thus they all felt that tailoring RUP is necessary with our project scope and requirement. team members also feels that a knowledge management environment should be developed within the organization so that team members can share their knowledge with other members and thus helps in tailoring RUP.

III. Conclusion –

In this case study we presented scenario of a medium sized organization developing a small project with team of 10 members. They found RUP is too much documentation seeking, so for small size project it is difficult to concentrate on coding as many documents have to be prepared first, and focus is mainly on documentation so the coding part suffers as time for developing small project is not much. So for small projects RUP has to be tailored for its effective use. Also for understanding RUP in detail RUP mentor should be appointed so that time is not wasted in understanding RUP by team members on their own.

IV. Future Research –

In this case study only one project is focused; in future we will compare two or three projects using empirical data of each project so that how RUP should be tailored for seeking its benefit will become more clear.

References :
[7] Mard’e Greeff and Louis Coetzee, ” Using Action Research for Complex Research Initiatives,

ISSN: 2277 – 9043
International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)
Volume 1, Issue 8, October 2012

All Rights Reserved © 2012 IJARCSEE


   Conference on Software Engineering and Advanced Applications


[16] Geir Kjetil Hanssen1, Hans Westerheim1, Finn Olav Bjørnson 2, ” Using Rational Unified Process in an SME – A Case Study”. 