

Overview of the Agile Rational Unified Process (Rup) in the Context of Software Development Projects

Paul Dragos

(Management Department, FEAA Faculty of Economics and Business Administration, West University of Timisoara, Romania)

Abstract: One of the processes of software engineering is the AGILE methodology, which is famous for its iterative method of development. Its well-known types include Rational Unified Process (RUP), Scrum, Extreme programming (XP), and Adaptive software development. RUP is an incremental software development process which delivers releases over time with improvements per release while following the best practices along the way. The four phases of RUP, Inception, Elaboration, Construction, Transition, provide a sequential and iterative approach to the development of a product. The workflow is divided into logical activities that are nine disciplines, and are maintained and performed throughout the four phases of RUP. RUP is not only a system development process but is also a mindset, which can be implemented with any process or project, by following certain recommended practices which will be discussed in the present paper.

Key words: agile; rational unified process; software development

JEL code: Y20

1. Background

The Agile methodology is one of the processes of software engineering, which is famous for its iterative method of development. Agile is especially useful in the context of a continuous evolution of the requirements and solutions, creating a solid foundation and helping the team to respond to the changes in a natural and faster way. Its famous types include Rational Unified Process (RUP), Scrum, Extreme programming (XP), and Adaptive software development. From the mentioned types, the Rational Unified Process (RUP) will be discussed in the present paper.

RUP stands for a Rational Unified Process; it focuses on creating and maintaining models, rather than on producing large amounts of documents. It is developed and maintained through Rational Software. The people responsible for RUP work in close quarters with clients, partners and product groups to ensure that the RUP is continuously updated and improved to keep up with the new times.

RUP guides in the usage of Unified Modelling Language (UML). UML is a standard language that allows users to display their requirements, designs and architecture. UML was originally developed by the Rational Software (Christensson, 2006) as well, but is now handled by the Object Management Group (OMG).

RUP is also supported by tools which help in the automating of large parts of the process. These tools are

used to maintain and create various artefacts, like models, for the software engineering process, such as visual modelling and designing, programming and testing. These all are the reasons that RUP provides the best practices in software development which are suitable for a wide variety of projects and organizations.

2. The Dimensions of RUP

There are two crucial dimensions of RUP, along with 4 phases and nine disciplines and their correspondence is indicated by Figure 1 (Anwar, 2014).

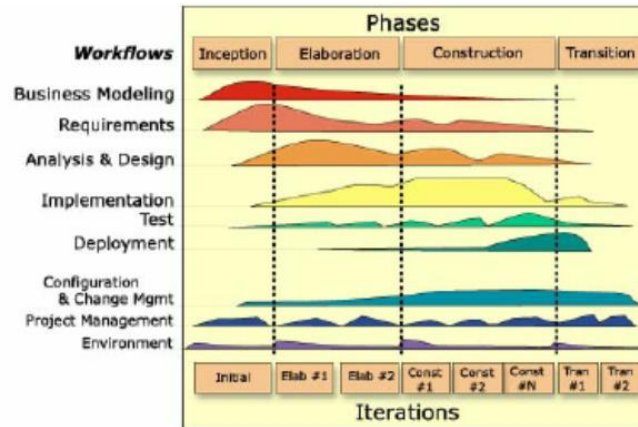


Figure 1 Overview of RUP

Source: Anwar, 2014

2.1 The Disciplines of RUP

There are nine disciplines defined by RUP (Hirsch, 2002):

a) **Business Modelling:** both the business processes and its internal structures are defined to get a better understanding and to be able to come up with requirements for the desired software. Business Modelling involves exploring and evaluating potential strategies used for the reengineering of the business processes.

b) **Requirements Management:** obtaining, organizing and processing and finally documenting the requirements. Working along with the project stakeholders closely to understand their requirements and needs is part of requirement discipline.

c) **Implementation:** understanding the design model and transforming it into a source code by implementing components and finally integrating it together with sub-systems and into the complete build.

d) **Test:** the three types of tests: integration, system and acceptance test are part of test discipline activities. That means both defining, planning and developing use cases to run tests on the project and, also, reporting the bugs/error found.

e) **Deployment:** finalizing the project, packages, scripts and end-user documentation for it, along with any other tasks required to supply the user with the complete end product software. That also means to develop support and operational material along with all the necessary packages for deployment and training the end-users as well.

f) **Project management:** planning and monitoring tasks for better management of the project through initiating a new project, management of staff, management of risk factors, planning and scheduling sub-tasks, doing iterations and completing the phase/project.

g) Configuration and Change Management: it includes everything about the version and after release and change request management and also consists of the configuration controls and its status by keeping track through monitoring. Baselines and releases management is also part of it.

h) Environment: processing materials for individual project teams, adapting to the project's or organisation's needs and introducing tools according to them and making sure all the tools etc. are appropriately installed to support the development of the project.

2.2 The Phases of RUP

Each software lifecycle is broken into smaller cycles, and each cycle represents a new generation of the software. RUP divides each development cycle into four phases. And each phase ends with a milestone. A milestone is a point or place in the process where certain decisions must be made, and certain goals are achieved.

RUP is a software development process that combines linear and iterative structures. It's four phases are Inception, Elaboration, Construction, and Transition (McManus, 2004). Every stage is done in many iterations except Inception. All other development process stages like requirements, design, testing, etc. are done corresponding with these four stages of RUP. Although the intensity is different, it helps build stable and flexible solutions. However, this software process is not the same as other Agile models like SCRUM and XP. It is not as efficient as other models of Agile. It is not as quick and adaptable as others. Customer involvement, iterations and their intensity vary depending on the project requirements.

The details of each phase are given below (McManus, 2004):

1) Inception Phase: the process of RUP starts from the first stage of Inception. In this phase, the idea of the project is proposed. Some of the milestones for this phase are a requirement document, general use case models and risk factors assessment.

2) Elaboration Phase: in this stage, the team developers evaluate the architecture and requirements of the project. Some of the milestones for this phase are use cases which have their actors identified, more in-detailed risk assessment and a general manual for users.

3) Constructions Phase: during this process, all the remaining features and functional components are developed and integrated into the build model to combine into a complete product. In this phase, all the processes, costs, quality, operations and resources are optimised. Some of the milestones of the phase are manuals, a complete product and description of the current software.

4) Transition Phase: in this phase, the product is released to the users. At the time of deploying to the users, issues might arise, which needs debugging and correcting the problems. Because of this, sometimes the new release is produced, and new features are added to the existing product. Some of the milestones are completed testing, properly operational databases, training of end-user and management and finally deploying the product to the market.

3. Best Practices Deployed by RUP

RUP effectively applies proven approaches on the software development process and the team responsible for it successfully obtains its objective. These practices are observed in successful organisations in the industry and they are given below (IMQS et al., 2015):

1) Develop the software iteratively: due to the changing times, no software can be designed, built and implemented through the sequential process methodology without there being a need to retouch some part of the

software due to some new update in the technologies or in the operating system etc. for this reason RUP allows the developers to iteratively go over the software in the process of development which helps in getting a deeper understanding of the particular problem through successive refinements and incrementally provide an effective solution over each iterative.

2) Requirements Management: RUP provides a way to obtain, organise and document the specified requirements and constraints, and make trade-offs and decisions. This is achieved through scenarios and use cases, which have been proven to provide the best way to capture functional requirements. These also help in tracing the requirements throughout the whole development process.

3) Component-Based Architecture: RUP provides support for component-based software development. Components are sub-systems that help in fulfilling a requirement. Through the use of new and existing components, RUP provides a systematic approach to defining an architecture.

4) Verification of Quality: one of the most important parts of software development is ensuring that the client receives a quality product. That means reviewing the functional requirements as well as the non-functional requirements to make sure that everything is in working state.

5) Model Visual Software: different models are used to communicate different aspects with different stakeholder. Through the usual visual models, we can capture the behaviour and structuring of the components of the required software.

6) Control Changes: because of iterative changes, it is necessary to be able to trace each change in the software. It is also needed to make sure that each change is accepted. Controlling, tracking and monitoring the changes to be able to integrate each iterative change successfully, are all part of this process.

4. Conclusion

Rational Unified Process (RUP) is an incremental software development process which delivers releases over time with improvements per release while following the best practises along the way. The four phases of RUP provide a sequential and iterative approach to the development of a product. The workflow is divided into logical activities that are nine disciplines, and are maintained and performed throughout the four phases of RUP. RUP is not only a system development process but is also a mindset, which can be implemented with any process or project.

References

- Anwar A. (2014). "A review of RUP: Rational unified process", *International Journal of Software Engineering*, Vol. 5, No. 2, pp. 8-24, available online at: <http://www.cscjournals.org/library/manuscriptinfo.php?mc=IJSE-142> .
- Christensson P. (2006). *RUP Definition*, accessed on 28 October 2020, available online at: <https://techterms.com>.
- Hirsch M. (2002). "Making RUP agile", p. 1, doi: 10.1145/604252.604254.
- IMQS et al. (2015). "Rational unified process best practises for software development teams", *Development*, Vol. II, pp. 378-382.
- McManus J. (2004). "Managing stakeholders in software development projects", *Elsevier Butterworth-Heinemann*.