Computer Engineering and Intelligent Systems www.iiste.org
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.6, 2011

Exploring Documentation: A Trivial Dimension of RUP

Bashir Ahmad
Institute of Computing and Information Technology
Gomal University, PAKISTAN
bashahmad@gmail.com

Muhammad Javed
Institute of Computing and Information Technology
Gomal University, PAKISTAN
javed_gomal@yahoo.com

Shakeel Ahmad
Institute of Computing and Information Technology
Gomal University, PAKISTAN
shakeel_1965@yahoo.com

Imran Ali Khan
Department of Computer Science, COMSTS, Abbotabad,PAKISTAN
imran@ciit.net.pk

Muhammad Ahmad Jan
Institute of Computing and Information Technology
Gomal University, PAKISTAN
mr_ahmadjan@yahoo.com

Sheikh Muhammad Saqib
Institute of Computing and Information Technology
Gomal University, PAKISTAN
saqibsheikh4@hotmail.com

Abstract

The Unified Process (UP) methodology is a commonly used methodology which can be followed by that entire process model where perfectly documented and well defined structure of team is needed, like Rational Unified Process (RUP) model which follows the UP methodology. During documentation, the defect rate of software can be reduced and software quality can be improved. Quality is the sole objective which is pursued by stakeholders throughout the whole life cycle of software development. Quality is not the outcome of an accident; it is the fruit of the continual labor of devoted professionals. As the size of software increases, it is natural for the number of errors and defects to increase. The Cleanroom Software engineering process is a process for software development. The basic objective of Cleanroom Software engineering is to produce high quality of software emphasizing to increase the level of reliability to its utmost efficiency. Moreover, the Cleanroom process is involved in each and every phase of software development life cycle i.e. planning; measurement; specifying design; verifying code; testing; and certifying to mold the entire engineering discipline that the end product should result 'ideally' in zero defect-rate.

Keywords: Cleanroom software Engineering process, Documentation, Defect rate, Rational Unified process, quality and reliability.

1. Introduction

Software development life cycle model is either a descriptive or prescriptive categorization of how software is or should be developed. Usual practice suggests that Descriptive models are the main tools in improvement of software development process by providing grounds for empirical results and hypothesis whereas Prescriptive models deal with the proper arrangement of activities or phases to be carried out in order to achieve the goal. This structuring of activities can yield better results by establishing more natural sequence [1].

The Cleanroom software engineering process is intended to apply different statistical, mathematical and empirical techniques to improve the quality of software process whereas the Rational Unified Process (RUP) is a well disciplined approach to assign responsibilities and services with in development organization to ensure the production of high quality software with the specified needs of its users. In every Rational Unified Process model, as the target is to achieve the desired level of quality, two aspects are considered as prime parameters: first, development of high quality software is achieved by providing techniques and second, in order to improve different quality attributes different techniques are provided which are derived through the basic model [2]. The Cleanroom process was originally developed by Dr.Harlan Mills with great contribution of his several colleagues, the founder of Software Engineering Technology [3]. As the less number of errors always result in low cost, the same is the objective of Cleanroom software engineering process which aims to reduce errors up to maximum limit or even zero errors in deliverable goods. The Cleanroom process provides a complete engineering discipline with which software team should have like planning, designing, controlling (verify code and testing) and finally certificate should be issued on successful completion of each and every iteration of each phase of software life cycle for verification at zero defect rate which means no need of rework. The main advantage of Cleanroom software engineering over the traditional software development is, the traditional software development methods are wide open to the errors which takes a valuable amount of time in exploring and correcting these errors whereas in contrast this approach Cleanroom methodology is less error prone which means less debugging and rework so there is higher level of productivity [4].

Here, author proposed a strategy to embed principles of Cleanroom software engineering in RUP to enhance the software documentation and to improve software quality.

2. Rational Unified Process Model (RUP)

The Unified Process (UP) is a use-case-driven, architecture-centric, iterative and incremental maturity process which is influenced by the Unified Modeling Language (UML, provided by the Object Management Group, OMG) and is fully compliant and amenable with the OMG's Software Process Engineering Metamodel (SPEM) [5].

UP is in fact a concept and a layout which provides the infrastructure and standards for the successful execution of the projects without knowing all the details. It is basically software development process framework and a lifecycle which involves context and background, collaboration and alliance, and interaction and communication of its stakeholders.

It was Philippe Kruchten, Ivar Jacobson and few others who conceptualized the RUP as a process harmonized to UML.

Rational Software Corporation provides the RUP as the process product for execution of the projects along with its standards, guidelines and templates [6].

RUP can be viewed using both the static and dynamic aspects.

2.1. Static Aspects of RUP

It involves Role, Activity, Workflow and Artifacts.

a. Role

It is the responsibility and dependability which is assigned to persons in the project e.g. Project Manager, System analyst, programmers etc.

b. Activity

It is the unit of work that an individual performs in the project. An activity has a transparent intention and is expressed in terms of manipulating any artifact. The granularity of an activity is generally a few hours to a few days and normally involves the workers and their associated artifacts.

c. Workflows

Significant sequences of activities that can generate valuable results, and to confirm collaboration and interactions between workers.

A workflow represents a sequence of activities that turn outs an outcome of noticeable significance. Sequence diagram, collaboration diagram, or activity diagram are the variations of workflow in UML.

d. Artifact

Information that is produced, modified, or used by a process is known as Artifact. Artifacts are tangible and concrete in nature; these are the elements produced or used by the project. They are used both as input for any activity and are output of any activity . In object-oriented designs, the activities are functions of an active object (the worker) and artifacts are the parameters and considerations of these activities.

The whole working of RUP can be classified into two aspects:

2.2. Dynamic Aspects of RUP

It involves various phases and their iterations.

a. Phases

RUP establishes four phases of development, each of which is organized into a number of separate iterations that must satisfy defined criteria before the next phase is undertaken: in the inception phase, developers define the scope of the project and its business case; in the elaboration phase, developers analyze the project's needs in greater detail and define its architectural foundation; in the construction phase, developers create the application design and source code; and in the transition phase, developers deliver the system to users. RUP provides a prototype at the completion of each iteration [12].

Phases of RUP are shown in Fig-1.

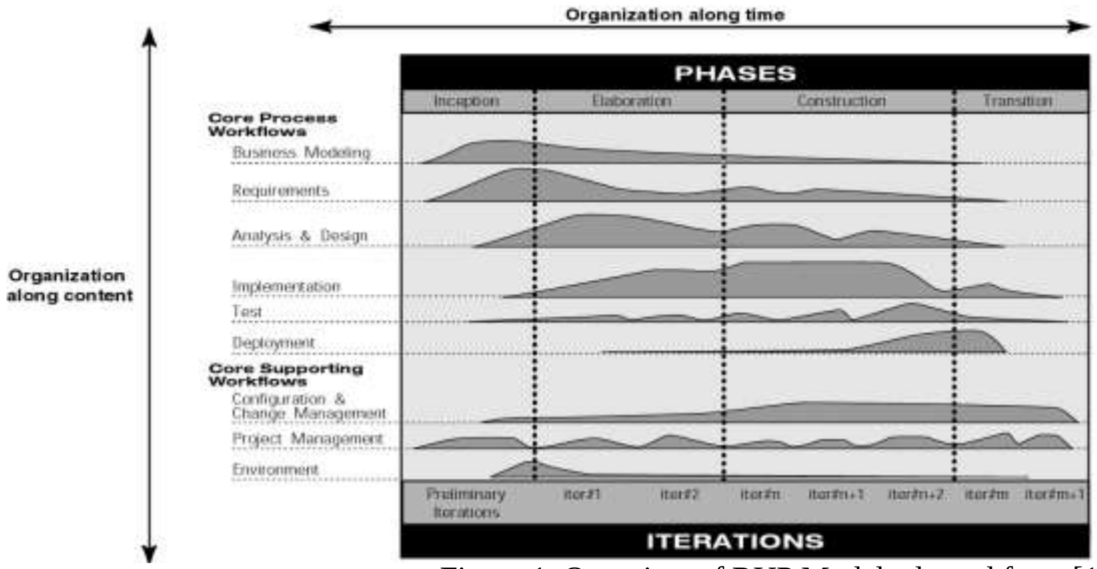

Figure 1. Overview of RUP Model adopted from [12]

b. Iterations

The implementation of RUP standards vary from case to case and the number of iterations for any product development may differ significantly. Each iteration does have milestones and set deliverables. At the end of each iteration, the decision of moving into next iteration is made to check whether it is feasible or not to do so[7].

3. Cleanroom software Engineering

The main intent of this process is to make the final deliverable completely error free as this itself is the standard of a low cost product [3]. This process of making error free product yields an output of highest quality and at the same time critical for decision making. The Cleanroom process is an end to end disciplined approach for software development which involves all the stages like planning, designing, testing. At the end of a successful development increment, a certificate is issued which certifies that the increment is completely error free and is at zero defects rate, confirming there will be no need of rework [8].

4. Cleanroom as Process Model

The Cleanroom process combines a variety of techniques such as incremental development; formal specification and design; correctness verification and statistical testing. The incremental work and certification process of Cleanroom software engineering is shown in Fig-2.

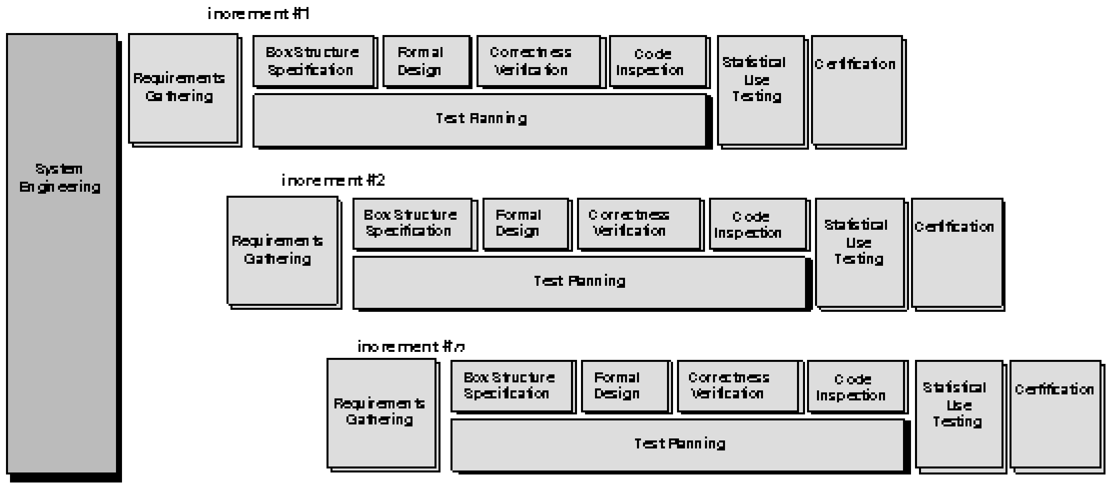

Figure 2. The Cleanroom Process Model adopted from [8]

In Cleanroom process model each increment or phase consists of requirements gathering; Box structure specification; formal Design; correctness verification; code inspection; Statistical testing and certification.

a. Requirement gathering

Requirements gathering phase involve all comprehensive details of the requirements of the customer which will be analyzed and designed in each increment.

b. Box structure specification

Box structure involves the separation and isolation of behavior, data and procedures defined in the functional specification phase.

c. Formal Design

Various specifications, also known as “black boxes”, keep on being refined and polished in every upcoming increment so that they become equivalent to the architectural and procedural designs, which are also known as “state boxes” and “clear boxes,” correspondingly [9].

d. Correctness Verification

The correctness verification is carried out starting from the box structure which is at the highest level and then progressing towards the detailed design and coding by means of “correctness questions”. If by any chance these practices do not certify the correctness of the specifications, then as a result of this more ceremonial mathematical routines are used for authentication.

e. Code Generation, Inspection and Verification

The box structure specifications which are symbolized using their appropriate languages are transformed into suitable programming languages.

f. Statistical Test Planning

It involves the planning and designing of a series of test cases which put into effect the “probability distribution”.

g. Statistical Usage Testing

The Statistical Usage Testing accomplishes a set of tests which are originated from a statistical sample possible program by all users. It involves the creation of test cases, their respective execution and collection of error data [10].

h. Certification

As all the verification, inspection and usage testing is successfully accomplished plus all possible and potential errors are corrected, then the increment is given a clearance certificate which guarantees that it can be integrated easily.

5. RUP w.r.t trivial Documentation and defects

Quality is the challenging factor which can be faced by stakeholders though out the whole life cycle of software development, to achieve a full quality product is not so easy, it is the promising result of intelligent effort and also as the volume of software increases, then it is natural for the number of errors and defects may also increase. In RUP reworking process of the existing development system is to be performed if any iteration is to be failed at in any phase which is entirely time consuming process. Software documentation is an attempt which is adopted by different developers to overcome above mentioned problem. In RUP process model, documentation is performed by different roles during each iteration of all disciplines. During RUP software development process roles are specified for documentation but still defect rate and quality remains challenging for software developers.

6. Incorporating Cleanroom principles in RUP

The support of software quality in a software development process may be regarded under two aspects: first, by providing techniques, which support the development of high quality software and second, by providing techniques, which assure the required quality attributes in existing artifacts. Both approaches have to be combined to achieve effective and successful software engineering [11]

Four sequential phases, are included in RUP each phase is concluded by major milestone, at the end of each phase assessment is performed whether the objectives of the customer are met , whether the quality of phase is best , I embed the Cleanroom software Engineering principles whether all these principles are applicable in that phase of RUP or not , when all these factors related to quality , working , zero defect rate and customer needs are met then certificate will be issued after satisfactory assessment of that progressed phase then move to the next phase of RUP.

In this paper , authors define a strategy to enhanced the documentation in RUP to increase quality of RUP process model and decrease the defect rate, this will be done to issue a certificate of acceptance after each iteration. Each activity in each phase of RUP can be performed in iterations and after completion of each iteration there is need to issue a certificate to carry on work successfully and verified the customer needs.

Each activity in RUP may be completed in one or more iterations, when the activity is completed then next activity is assigned and so on. According to authors when the iterations are completed for any activity then certificate will be issued to verify reliability as shown in Fig-3.

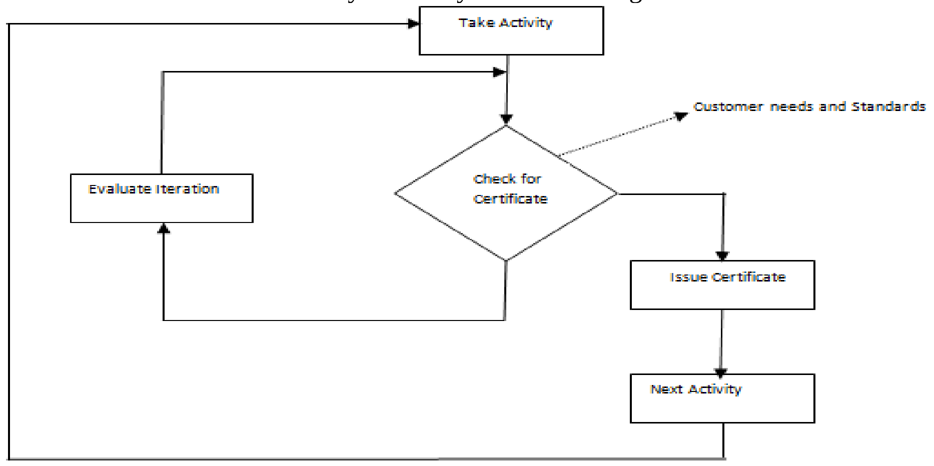

Figure 3. Issue Certificate to verify reliability

7. Mapping of RUP Process model with Cleanroom software Engineering

When the activity is going to be completed iterative manner in RUP then it must be notify that the completion of iteration is successful or not , it can be check by mapping Cleanroom software engineering principles whether customer needs are fulfilled; whether the iteration has zero defect rate means no need of rework if all these checks performed well then certificate should be issue otherwise suggest other mechanism to complete that iteration by other source the mapping of RUP with Cleanroom software engineering is shown by Table-1. The suggestion column of Table-1 shows the states of iteration which can either sound or required some improvement in the given iteration. Similarly the source column directs towards the kick-off points of errors, which can be either the flaws on statistical testing or other errors in component and so on.

Once the iteration completion considered as successful, it left no space for suggestion. The Table-1 leads towards the issuance of certificate for iterations, only when the defect rate is zero and customer needs are not existent i.e. all possible requirements of customer are successfully satisfied.

The verification of statistical test for any component also leads towards the minimization of defect rate, it is to be noted that certificate can not be issued unless the defect rate is equivalent to zero although all the customer requirements are fulfilled and vice versa. So in Table-1, the same analogy will be followed for all activities.

Table-1. Mapping certificate format

Activity No	**Roles**	**Iteration No**	**Customer Needs**	**Leads to defect rate**	**Issue Certificate**	**Suggestion**	**Source**
A1	R1,R2	I1	x	✓	X	✓	✓
	R1,R2	I2	✓	X	✓	X	✓
A1	R1,R2	I1	x	✓	X	✓	✓
	R1,R2	I2	✓	✓	X	✓	✓
	R1,R2,R3	I3	✓	X	✓	X	✓
A2	R1,R2	I1	✓	✓	✓	X	✓
A3	R1,R2	I1	x	✓	X	✓	✓
	R1,R2,R3	I2	x	X	X	✓	✓
	R1,R2	I3	✓	X	✓	X	✓
.		.	.	.	.	.	.
.		.	.	.	.	.	.
.		.	.	.	.	.	.
.		.	.	.	.	.	.
.		.	.	.	.	.	.
.		.	.	.	.	.	.
An		.	.	.	.	.	.

8. Conclusion

The reason behind the popularity of RUP are its well defined documentation and iterative approach, but still documentation and well defined iterative approach is challenging to overcome the defect rates and quality improvement. According to authors suggestion there should be proper certification process at the end of each iteration. Due to this, only right iteration for each activity will well documented and it causes to increase the quality of RUP products and can guaranty for zero defect.

References

Walt Scacchi, Institute for Software Research, University of California and Irvine, “Process Models in Software Engineering”, 2001.

Wolfgang Zuser at al.” Software Quality Development and Assurance in RUP, MSF and XP- A Comparative Study”. 2005.

Mills, H.; M. Dyer and R. Linger (September 1987). "Cleanroom Software Engineering". IEEE Software 4 (5): 19–25. doi:10.1109/MS.1987.231413.

Paul Murphy ,”A Review of Cleanroom software Engineering”,1996.

Rumbaugh, J., Jacobson, I., & Booch, G. (1998). The Unified Modeling Language reference manual. Reading, MA: Addison Wesley Longman,mInc.

Sinan Si Alhir,” Understanding the Unified Process (UP)”, Methods & Tools- An international software engineering digital newsletter, March 2002.

Philippe Kruchten, A Rational Development Process, CrossTalk, 9 (7), STSC, Hill AFB, UT, pp.11-16. July 1996.

David C. Wofford,” Report to the Manager: Cleanroom Software Engineering”, 1997.

Michael James Banks,” Formal Modeling of Secure Systems Qualifying Dissertation”, 30th July 2009.

Trammell, C. ,” Quantifying the reliability of software: statistical testing based on a usage model”,Experience and practice’,Proceedings.,Second IEEE International,Aug 1995.

Wolfgang Zuser at al.” Software Quality Development and Assurance in RUP, MSF and XP- A Comparative Study”. 2005.

Rational Software Corporation," Rational Unified Process Best Practices for Software Development Teams”, Rational Software white paper ,TP026B,Rev 11/01.200