An Evolutionary Testing Solution of Rational Unified Process Model to Develop ERP Software in Pakistan

Sabahat Tasneem, Mr. Akmal Rehan

Abstract---In Pakistan, The evaluation phase of Enterprise Resource Planning (ERP) Development Life Cycle is facing Product Risks and Project Risks due to its infancy. All of these risks must be nipped in the bud, before they become monstrous and threat about the software. In this research paper, a systematic technique based upon Rational Unified Process (RUP) Model and hybrid testing technique based upon Acceptance Test Driven Development (ATDD), Pair Testing, State Transition Testing and Exploratory Testing Techniques, which can be adopted in Rational Unified Process (RUP) to mitigate these risks, involved in Enterprise Resource Planning (ERP) Testing, effectively in a tight time frame, has been presented to use in Pakistani Software Houses. Quantitative and Qualitative Techniques were used to gain better results in research.

Index Terms— Enterprise Resource Planning (ERP), Rational Unified Process (RUP), Revolutionized Rational Unified Process, Acceptance Test Driven Development (ATDD), Software Development Life Cycle (SDLC), Specification by Example (SBE), Behavior Driven Development (BDD), Example-Driven Development (EDD), Story Test-Driven Development (STDD), State Transition Diagram (STD)

1 INTRODUCTION

In this era of competition, the competitor organizations all over the world are using ERP Software to fulfill their volatile requirements of manufacturing process, finance, human resources management, supply chain management, customer relationship management, depository management, etc. to promote their product in market. The development method and testing techniques used for ERP software development or up gradation affect the overall cost of the product. Software testing as a process of validation and verification of the product, paves way for the developers to anticipate about the Quality, Time and cost of the product in advance. It must start from the early phases of SDLC to improve quality, reliability, and performance and to decrease cost consumption and time box [3], [6]. According to the rule "1:10:100" the project manager can save Rs.10 on appraisal and 100 on failure costs, by spending Rs.1 on prevention.

Figure 1: illustrates that defects are frequently originate during requirement elicitation and specification phase. The first runner up is requirement elicitation, second runners up is design phase, third runners ups is others phase (maintainability) and the fourth runner up is coding phase. Figure 2: illustrates that commonly most of the testing resource and efforts are used during testing phase in traditional SDLC Process Model, according to a past survey. The major problem which arises here is that the product becomes costly and risky, due to short testing time and resources.

Software Development Life Cycle is a frame work to be followed by developer, designer and tester to manage their allocated resource to develop efficient software with in tight time frame and to meet expectations of all stake holders [2]. The suitable methodology is selected on the basis of some conditions such as, budget, time and risk mitigation techniques [5]. However, there are different types of methodologies to develop software; but unfortunately, there is not even a single methodology which can claim to achieve above mentioned goals. This research paper is an endeavor to achieve those goals.

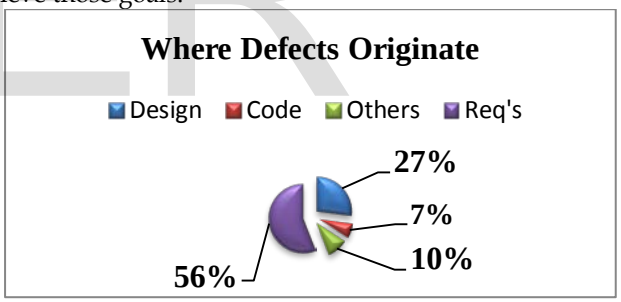


Fig. 1. Illustrates where does the defect originates the most

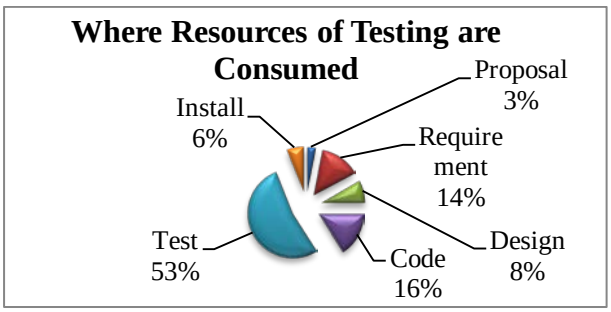
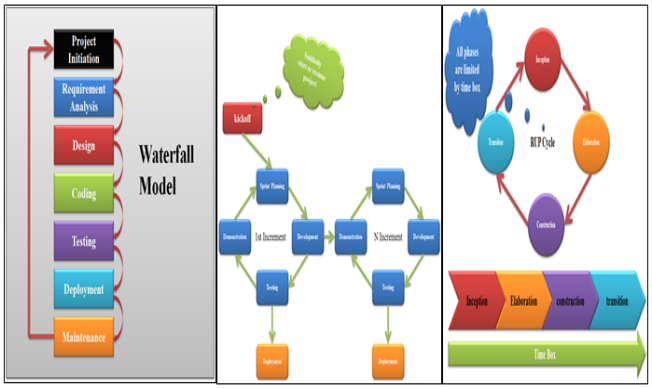


Fig. 2. Illustrates where does the resources of testing are consumed mostly

The Figure 3: Illustrates the comparison among thrice models known as Waterfall Model, Agile Model and Rational Unified Process (RUP). Waterfall model follows the famous proverb, "Define before design and design before code" and

believes to end previous phase before to chase next. Agile Model as an incremental model breaks down a task into small iterations without long term planning. While Rational Unified Process (RUP) as a tailor-able, customizable and iterative model, it can be defined as a best mixture of traditional Waterfall Model and modern Agile Model. This model pays more attention towards the collection of feedback of all stakeholders and produces a prototype at the end of each iteration.



Fig, 3. Illustrates the comparison between Water Fall Model, Agile and RUP

1.1 Pros and cons of Waterfall Model:

Pros:

- Comprehendible for Experts and Novice
- Help from previous documentation
- Deliverables from each phase to the next
- Supports stable requirements only
- Suitable for smaller projects
- Execution of one phase at a time
- Focus on quality

Cons:

- Uncertain and risky
- Testing before deployment
- Less customer involvement
- No support for volatile requirements
- Not suitable to develop ERP Software
- Delay in deployment due to serious error
- No focus on time and schedule

1.2`Pros and cons of Agile Model:

Pros

- Rapid and continuous deliverables
- More Customer involvement
- Face-to-face meeting among stakeholders
- Supports volatile requirements
- Supports late changing requirements

Cons:

- Lack of time for designing and testing
- Lack of documentations
- Ambiguous requirements spoil the product
- No place for newbie

1.3 Pros and cons of Rational Unified Process:

Pros:

- General terms allow local customization
- Focus on proper documentation
- Iterative and incremental model
- Evolutionary approach
- Use Case Driven
- Manageable
- Traceable
- Tunable

Cons:

- General Terms provide slight guidance
- Requires heavy toolsets
- Heavy weight model on process side
- Expensive documentation

1.4 Testing Phases of Thrice Models:

- In waterfall model testing occurs in the 5th level, while in Agile model and RUP, it is a continuous process till the end of project.
- There is enough time for detailed testing in both waterfall model and RUP, but not in Agile.
- In waterfall model, the whole product is tested, in Agile a sprint is tested individually and in RUP an increment is tested at various levels.
- In Waterfall Model and RUP, testing artifacts are produced for guidance in maintenance phase, but not in Agile Model.
- In RUP the prototypes are produced for the surety of risk mitigation, while there is no concept of prototype in both Waterfall Model and Agile.
- RUP is not risky due to testing at various levels, while waterfall and Agile Model are.
- Waterfall model is more costly than Agile and RUP, RUP is less costly than Waterfall and slight more than Agile and Agile is less costly than both Waterfall and RUP.

2 Revolutionized Rational Unified Process (RRUP):

The Revolutionized Rational Unified Process Model is the developed Model, which is comprises on four phase of traditional Rational Unified Process, such as Inception, Elaboration, Construction and Transition, and testing is conducted throughout the four phases of RRUP. . In inception phase, software initiation and planning was conducted along with testing phase. In Elaboration Phase modeling, designing and planning was conducted along with testing. During construction Phase the design which was planned during elaboration phase was implemented along with testing phase. During transition phase the product can be deployed after testing that product thoroughly to reduce the chances of defects occurrence. Testing Technique of RRUP comprises on additional Acceptance test Driven Development (ATDD further comprises on Specification by Example (SBE), Behavior Driven Development (BDD) Example-Driven Development (EDD), and Story Test-Driven Development (STDD) techniques to enhance customer understanding and the initial stage of Revolutionized Model), State Transition Diagram (STD), Pair Programming, and Grey Box Testing. All these mentioned techniques can reduce cost, increase documentation for maintenance phase, can achieve goals and user requirements in limited time box with better quality.

The RRUP adopts the traditional software development methodology when starting from scratch. Changes can be incorporated by adopting agile methodology with changing environment in order to achieve time saving and cost effective product [8]. Reliability is one of the most important quality attributes of commercial software. In order to increase the reliability, we need to have a complete test plan that ensures all requirements are included and tested. In practice, software testing must be completed within a limited time and project managers should know how to allocate the specified testing-resources among all the modules. The main purpose should be to minimize the cost of software development when the fixed amount of testing-effort and a desired reliability objective are given. An elaborated optimization algorithm based on the Lagrange multiplier method can be used [4].

2.1 Acceptance Test Driven Development (ATDD):

This Technique enhances the testing capability of RRUP. comprises on Specification by Example (SBE), Behavior Driven Development (BDD) Example-Driven Development (EDD), and Story Test-Driven Development (STDD) techniques to enhance customer understanding in the initial stage of Revolutionized Model. These techniques used the graphical techniques to enhance customer understanding, so may he or she can impart his/her complete requirements in the initial phases. Because it becomes a major problem if customer does not able to impart his / her requirements in initial phase, and when he/ she understands what he/ she expect from the software to do, it becomes too late.

2.2 Pair Programming:

The pair Programming follows the famous proverb, "One is alone, but one and one make eleven". In pair programming one person writes coding and the second person continuously observe the coding to find out errors frequently during coding phase. Both of them can share their ideas to improve software quality. In this way, error occurrence reduces in Revolutionized Rational Process mode. This makes the testing more efficient and less costly.

2.3 State Transition and Exploratory Testing Technique:

These techniques also help to improve the testing process. State Transition testing is a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. . Grey Box Testing in Revolutionized Rational Unified Process Model can replace the working of black box testing and white box testing separately. In this regard, it can be said that Revolutionized Rational Unified Model is a revolutionary Rational Unified Process Model with latest testing techniques which helps the Project manager to develop an Enterprise Resource Planning Software within the time box with better quality and less cost.

2.4 Basic Principles:

. The developed methodology follows the Six (6) basic principles of Rational Unified Process (RUP), which are just like commandments for software developers The software developer can have the right to break these commandments only at inevitable risk [1]. Those basic rules are as follows:

- **A**_____Adaption of the Process
- **B**_____Balance the requirements of all stakeholders on the basis of their priorities.
- **C**_____Collaboration among all team members
- **D**_____Demonstration of all values iteratively
- **E**_____Elevation of abstraction level
- **F**_____Focus on quality of the product continuously

3 Phases of Methodology:

3,1 Inception:

Inception is the first phase comprises on feasibility report, the Vision Document, the Use Case Model, Initial Project Glossary, An initial business Case, Business Model, Prototypes, Various phases and iterations to develop a project.

3.2 Elaboration:

Elaboration Phase comprises on Design of the system, Use case, Prototype of architecture, Analysis Model, a list of revised risk and user manuals are produced in detail. The goals and development methodology is planned in detail.

3.3 Construction:

This phase comprises on design planning, Bulky coding, Beta testing, Design model, Software components, Test plans and test cases and Supportive Manuals.

3.3 Transition:

In transition phase, the produced product is deployed to the end user for implementation. The training programs are arranged. Beta testing is used test whether the product fulfills the expectations of the end user or not. The end user is asked to give feedback.

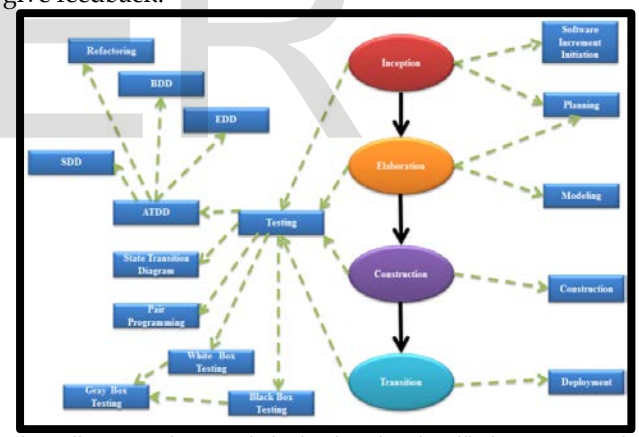


Fig.4. Illustrates the Revolutionized Rational Unified Process Model (Developed Model) with testing specification

4 Rational Unified Process Discipline:

Disciplines of Rational Unified (RUP) are also distributed along with various phases and iterations. They can be categorized into two basic disciplines, Engineering Discipline and support discipline.

4.1 Engineering Disciples:

The six Engineering disciplines of Rational Unified Process are requirements, business modeling, analysis & design, tests, implementation, and deployment.

4.1.1 Business Modeling:

In this process the vision of the organization is defined, which is further used to plan development process and to assign roles and responsibilities to all stakeholders.

4.1.2 Requirement:

It is the process explains the requirement elicitation process and the method to transform those elicited requirements into an organized set of requirements.

4.1.3 Analysis and Design:

It is used to produce the design model to provide blueprints to the programmer.

4.1.4 Implementation:

In this process, Classes and objects are implemented by using design model. Unit testing occurs at this stage.

4.1.5 Testing:

Basic purpose of testing is to verify and validate the product quality. This is the discipline starts working from initial software development process and continuous till the end of the process (Inception to Deployment).

4.1.6 Deployment:

The basic purpose of this discipline is to deliver the developed product to the end user after successfully releasing it.

4.2 Support Discipline:

Support disciplines of Rational Unified Process are configuration and change management, project management and environment. Their details are given below:

4.2.1 Configuration and change management:

Its basic purpose is not only used for tracking versions but also used to control changes.

4.2.2 Project Management:

In it a plan phase designates the whole project and an iteration plan defines the iterations in detail.

4.2.3 Environment:

It pays attention towards processes and tools to provide an environment of software development to make it easy, light weighted and inexpensive.

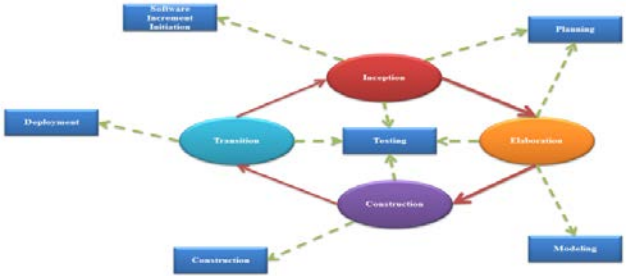


Fig. 5. Rup Disciplines workflow during all phases

5 Best Practices implemented in Rational Unified Process (RUP)

- **Iterative Software Development:**

This practice is implemented by planning frequent iterations for software development results in executable requirements.

- **Requirement Management:**

Use cases are used to capture functional requirements with detail.

- **Component Based Architectures:**

Architecture is defined by using different components.

- **Visual Software Model:**

Unified Modeling Language (UML) is used to develop model of software.

- **Quality Verification:**

Testing is used for verification of quality throughout life cycle

- **Software Change control:**

Change isolation and procedure is used for change control.

6 Software Testing Goals:

- **Validation & Verification:**

Verification means the product works as desired and validation means that the software fulfills the conditions laid down.

- **Priority Coverage:**

Testing should be within budget and schedule limits.

- **Balanced:**

Testing should balance the technical specifications, user requirement and expectations.

- **Traceable:**

The documents should be prepared at every level of success and failure of testing to save time.

- **Deterministic:**

Tester should be able to understand the targets and possible outcomes.

7 Roles and Responsibilities in Software Testing Process:

- **The Moderator:**

The moderator (or review leader) directs the Testing process by plan monitors and manages all testing activities. He determines the review's type, schedules the meeting, classifies the documents for review process and stores them.

- **The Author:**

The author writes the document under discussion.

- **The Scribe:**

The scribe is also known as recorder.

- **The Reviewer:**

He is also called inspector or checker responsible to inspect defects.

- **The Manager:**

The manager allocates a time frame for testing process and has check and balance on results.

8 Software Testing Techniques:

8.1 Manual Testing:

It is, also known as Static or verification testing, used to test manuals, requirement specifications and testing plan.

8.1.1 Walk Through Testing:

In this testing, the author conducts meetings to explain the product for the team members.

8.1.2 Informal Review:

The reviewer just reviews the Work documents and gives informal comments about it.

8.1.3 Technical Review:

Experts review the technical specification of product and focus on Testing Plan, Testing Strategies and documents of requirement specification.

8.1.4 Inspection:

The project team plans meeting in which each member is assigned a fixed role to perform. A team member reads the code of a specified product and other team members start to inspect that product and finds defects. A scribe records the defects. Moderator coaches all the team members for better results.

8.2 Automated Testing:

A testing technique determines the source code's quality attributes by using a predefined input set. It is also known as Dynamic Testing, Execution Testing or Validation Testing.

8.2.1 Performance Testing:

It is performed to determine how a system does responds and to what extent it remains stable under a particular workload. It measures scalability, reliability and utilization of resource.

Stress Testing:

It is used to test the ability of the software to maintain its effectiveness level when it faces unfavorable situations.

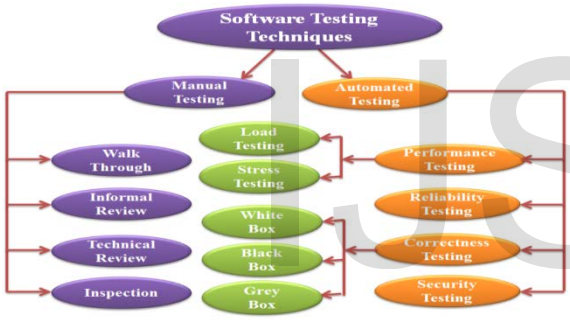


Fig. 6. Types of Testing Techniques

8.2.3 Load Testing:

It is used to measure resource-utilization, time of response, rate of throughput, and to identify product's breaking point.

8.2.4 Correctness Testing:

It is used to test the minimum requirement that the new product must possess.

8.2.5 White-Box Testing:

It requires deep knowledge of code and the test cases chosen verifies if the system is implemented as expected.

8.2.6 Black-Box Testing:

In this testing technique, without having internal knowledge, tester tests how software gives output in respond to an input.

8.2.7 Grey-Box Testing:

A combination of Black-Box Testing and White-Box Testing is known as Grey Box Testing.

8.2.8 Reliability Testing:

It is used to make sure that the failures of the product have been removed before the deployment of that product.

8.2.9 Security Testing:

Software Security testing is done to check whether the software is secured or not. It also checks whether the software can be attacked or hacked by anyone or not.

9 Software Testing Life Cycle:

9.1 Requirement Analysis:

In this phase, The team of Quality Assurance interacts with a number of stakeholders and study their requirements by categorized them into Functional and Non Functional a requirement.

Deliverables:

- Requirement Traceability Matrix
- Automation Feasibility Report

TABLE 1: ILLUSTRATES THE TESTING TECHNIQUE ACCORDING TO QUALITY FEATURES [7]

Testing Techniques According to Quality Features

Features	Testing Technique
Performance	Performance Testing
Compatibility	Compatibility Testing
Reliability	Stress and load testing.
Vulnerability	Penetration testing
Consistency	Database testing
Correctness	Database testing
Portability	Portability Testing
Recovery	Recovery testing
Completeness	Condition and Path Testing

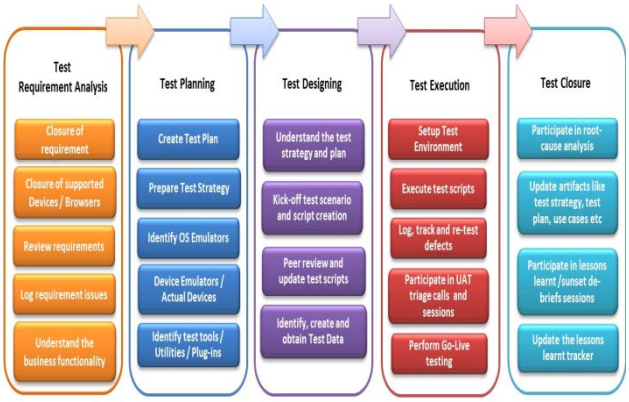


Fig. 7. Testing Phases

9.2 Test Planning:

In this stage, a Senior Quality Assurance manager determines the required effort and estimates the expected cost for the project to be completed.

Deliverables:

- Test plan document
- A document of required effort estimation

9.3 Test Case Development:

The test cases and test scripts are produced, verified and reworked.

Deliverables:

- Test cases and test scripts
- Test data

9.4 Test Environment Setup:

It determines the conditions of software and hardware under which a software product is tested.

Deliverables:

- Environment
- Results of Smoke Testing are calculated

9.5 Test Execution:

In test execution, testing team carries out the process of testing the product by using testing plans and test cases designed during previous phases of Software Testing Life Cycle.

Deliverables:

- Requirement Traceability Metrics (RTM)
- Test cases and Defect reports

9.6 Test Cycle Closure:

In this phase, the testing team sit together to talk over and analyzed testing result to identify best practices and strategies to implement in future testing programs and to obtain lessons from recent testing life cycle.

Deliverables:

- Test Closure Report
- Test metrics

CONCLUSION:

Software testing is required to find the defects and errors during the development phases. The Developed Revolutionized Rational Unified Process Model introduced revolutionized testing techniques for quality assurance of ERP software products. It can efficiently mitigate product and project risks. As RUP is tailor-able methodology that's why Acceptance Test Driven Development (ATDD), pair Programming, state transition Testing and exploratory Testing Techniques can be tailored inside it. Because this hybrid testing technique can achieve all testing goals and software houses can adopt it to develop quality ERP Software.

FUTURE WORK:

As I have discussed above I will present this draft version for testing and will request feedback on it so that amendments can be made if necessary. Furthermore, I think that there are a lot of chances of improvements in testing phase. It is not written on a stone, it can be improved. In testing more work is required to refine this model.

References

[1] Ashbacher, C. 2008. IBM Rational Unified Process Reference and Certification Guide Solution Designer. Journal of Object Technology. 7(6): 53-54

[2] Bassil, Y. 2012. A Simulation Model for the Waterfall Software Development Life Cycle. International Journal of Engineering & Technology (IJET). 2(5): 2049-3444

[3] Devi, R. 2012. Importance of Testing in Software Development Life Cycle. International Journal of Scientific & Engineering Research. 3(5): 1-5

[4] Huang, C. Y. 2006.Optimal resource allocation for cost and reliability of modular software systems in the testing phase. Journal of System and Software. 79(5): 653–664

[5] Massey, V. 2012. Evolving a New Software Development Life Cycle Model (SDLC) incorporated with Release Management. International Journal of Engineering and Advanced Technology (IJEAT). 1(4):25-31

[6] Saini, G. 2013. An Analysis on Objectives, Importance and Types of Software Testing. International Journal of Computer Science and Mobile Computing. 2(9): 18-23

[7] Tuteja, M. 2012. A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models. International Journal of Soft Computing and Engineering (IJSCE). 2(3):251-257

[8] Veerish, B., K. Lyytinen and N. Berente. 2012. Iteration in Software Development: A Comparison of Agile and Waterfall Software Development Project. 7th – Pre ICIS

- *Author name is Sabahat Tasneem (2008-ag-851), Student of MS Computer Science in University of Agriculture, Faisalabad, Pakistan, PH-0092-3336670445 & 0412659783. E-mail: sb_tasneem@yahoo.com*
- *Co-Author name is Akmal Rehan, Lecturer in Computer Science in University of Agriculture, Faisalabad, Pakistan, PH-0092-3006676515, E-mail: akmalrehan@uaf.edu.pk*