

A Progressive Testing Solution with Spiral Process Model to Develop ERP Software in Pakistan

Iqra Mahmood, Akmal Rehan, Sana Saeed

^{1,2,3}Department of computer science university of Agriculture Faisalabad, Pakistan

Abstract: In Pakistan, Testing and implementation of ERP software projects facing product and project risks due to its initial stages. All these risks whether related to project or product must be tackled before these risks became the threat for software success. Better evaluation of ERP software required road map for Planning, identifying risks and testing activities. In this research paper a Progressive testing technique followed by spiral process model was introduced to develop ERP software. The spiral process model identified different ERP software risks and progressive testing technique tackled with these risks. This Progressive testing technique involved (TDD),(ATDD),(TFD),(TLD),(BDD),Pair Programming and gray box testing. Resultantly a Revolutionized Spiral Software model RSSM was developed with better testing solutions for the successful development of ERP software in Pakistan. To get a larger nationwide view on the RSSM (across Pakistan), a survey was conducted through a questionnaire and the results were analyzed to show whether the provided solution meets the organization goals.

Keywords: Enterprise Resource planning (ERP), Test Driven Development (TDD), Acceptance Test Driven Development, Test First Driven (TFD), Test last Driven (TLD), Behavior-Driven Development (BDD), Software Development Life Cycle (SDLC), Revolutionized Spiral Software model.

I. INTRODUCTION

Business organizations are now facing market challenges as they compete in their day-to-day competitions. Every company wants to promote its products among its customers and manage its sales. This is why they are faced with a sudden change in the demands to successfully achieve their goals. Enterprise resource planning (ERP) is the integration and control of enterprise-wide information systems across the entire organization of all business processes. ERP is a "bundled business planning framework that allows organizations to provide a complete, integrated solution for the organization's information processing needs, efficient management and operational use of assets (materials, human resources, finance, etc.). The development process and testing techniques required for ERP software effect overall cost of the product [1]. Testing is the key attribute for the software success. It is a process to identify bugs, to mitigate bugs before they become monstrous and threat for cost consumption, and to check either the product fulfills its functional and technical requirements or not and either it does what the customer expects from the product to do or not. It is necessary to detect errors in the early stages of SDLC to produce defect free and Reliable software in defined time period [2], [3]. According to the rule

“1:10:100” A defect that is identified early in the development cycle can be significantly 100% time cheaper to rectify than those that are not found until much later in the production of the software [2]. Fig 1. depict that the ratio of defect origination is high in requirement elicitation and specification phase than design, code and other phases of SDLC. Fig 2: shows that most of the testing resources are utilized during the testing phase of SDLC according to the past survey. Due to the limited development time and resources the product becomes expensive and risky which leads the software towards failure.

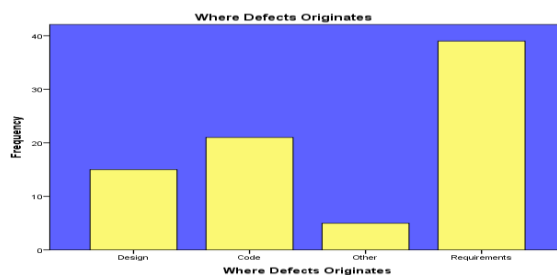


Fig. 1. Where defects originates

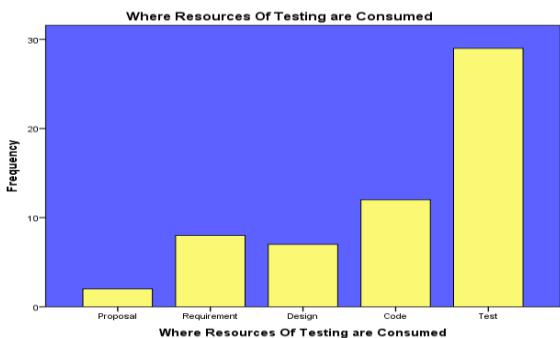


Fig. 2 .Where resources of testing are consumed

SDLC is the scaffold for software development followed by the designer, developer and tester to supervise their endless efforts and resources in defined time frame to develop reliable software [4]. The choice of method depends on parameters such as time, budget and risk mitigation. There are different approaches to software development, each with its own advantages and disadvantages. There is not even a way to claim to fit all situations. Therefore, the software developers according to the requirements of the situation to choose a method are the biggest challenge. In this case, it was observed that there should be a hybrid approach with the latest testing techniques to produce defect-free, high-quality products over a specified time period and to save cost in the maintenance phase. Due to product risks and project risks, the product may fail and may disrupt the reputation of the Software houses. This research paper is an adventure to implement these parameters.

A. Software testing:

Software testing may be defined as a process that is performed to evaluate the software to identify errors therein, to evaluate the capabilities and functions of the software, and to determine the quality of the software. The most common quality factors are availability, efficiency, reliability, portability, integrity, security, maintainability, efficiency, and capacity. In other words, software testing is the software validation and verification process.

Table 1. Illustrates the Software Testing Goals in detail

B. Testing in Water Fall Model:

The testing took place in a 5-level of waterfall model. The requirements defined in the early stages are used to validate and validate the product during the testing phase. Unit Testing, Acceptance Testing, and System Testing Evaluate the end product using the test cases defined by the quality assurance team to ensure that it meets the requirements of all stakeholders and end

users (as defined in the initial phase). Unit tests are used to evaluate software modules. Conduct a system testing to see how the modules work together at the integration level. The performance testing by the customer. If a defect is detected at this level, the deployment of the product is delayed. The tester detects errors and requires the coding team to make corrections. Testing at this level becomes expensive and increases the risk. Prepare, review, and distribute

Software Testing Goals		
No.	Goal	Description
1	Validation and Verification	Verification means that the product works as needed, and verification means that the software meets the specified conditions.
2	Priority Coverage	Testing should be within budget and scheduling constraints
3	Balanced	Testing should balance technical specifications, user requirements and expectations
4	Traceable	The file should be prepared at each level of test success and failure to save time by not wasting time to test the same thing again.
5	Deterministic	Testers should be able to understand the goals and possible outcomes

product documentation at this stage.

C. Testing in Agile Process:

Testing in the agility method continuously until the project is complete. Since there was not enough time for detailed testing, each sprint was developed and tested separately before the integration tests for all sprints. Agile teams contribute their expertise. Testing is done in real time, focusing on the quality of the software. Due to the short time, testers can not plan detailed test criteria, which is why only the necessary corrections are made. Each Sprint is designed, developed, tested and deployed before moving to the next Sprint. Then all Sprints are tested at the integration level.

D. Testing in Spiral:

In Spiral process model testing were performed at all stages. Testing of all stages across a spiral process model has become a major feature of this approach. This feature makes the difference between various software development methods. By adopting this feature in the software development process, the project manager can successfully develop the product with fewer defects for deployment. Test artifacts generated at various stages of the development process can be aided in future maintenance phases after deployment. Execution testing and prototyping gives assurance of mitigating all possible risks involved in the project.

E. Comparison of Testing Phase of Tri Process models

Agile is not concerned with testing and documentation of the product development process. Water fall model focuses on the document, but does not pay attention to the testing phase. In the water fall model, the testing phase becomes too late, and because of this problem, these models become risky because the product may fail after consuming effort and resources. Water fall model testing becomes expensive. The spiral process model is a custom and iterative, incremental model. As a result of these features, it becomes a good choice to develop enterprise resource planning software in Pakistan due to its risk assessment function.

II. Revolutionized Spiral Software Model (RSSM):

The revolutionary spiral process model is designed to provide an effective way to overcome all of the problems faced by all of these approaches. Agile is not concerned with testing and documentation of the product development process. Water fall model focuses on the document, but does not pay attention to the testing phase. In the water fall model, the testing phase becomes too late, and because of this problem, these models become risky because the product may fail after consuming effort and resources. The spiral process model is a custom and iterative, incremental model. Because of these characteristics, it has become a good choice for enterprise resource planning software development in Pakistan. In the revolutionary spiral process model, the latest testing strategies are being introduced to make traditional spiral process models more efficient by increasing customer engagement, and understandable documentation developed for the maintenance phase, and by using extreme testing techniques such as acceptance Test-driven development and pair programming. It is necessary to collect the customer's real requirements during the planning phase. The spiral process model for improving this capability innovation uses acceptance test-driven development (ATDD) techniques. This technology includes examples in the specification (SBE), Behavior Driven Development (BDD), Example Driven Development (EDD) and Story Test Driven Development (STDD) techniques to enhance customer understanding in the initial stages of the revolutionary model. These techniques use graphical techniques to enhance customer understanding so that he or she can communicate his / her complete requirements in the initial stages. Because it becomes too late if the customer cannot communicate his / her requirements at the initial stage and when he or she understands his or her expectations of the software.

In the proposed model, programming has also been introduced to enhance the testing process. In pair programming, one person writes the code, and the second person observes the encoding continuously to frequently find errors during the encoding phase. They can share their ideas to improve software quality. In this way, errors are reduced in the revolutionary spiral process model. This makes the test more efficient and less costly. In this regard, it can be said that the RSSM is a revolutionary spiral software model with the latest testing techniques to help project managers develop an enterprise resource planning software within the time frame, with better quality and lower cost.

III. Research Methodology:

The purpose of this research was to study exiting software quality assurance methods and to present a quality model which should be appropriate for customized development models. There is need of new methodologies with latest testing techniques, because there are a lot of companies which are working with their customized process models with less focus on testing phase and compromised on quality of products. Due to the testing problems in waterfall, agile and spiral process model a hybrid testing technique which is a combination of static and dynamic testing introduced in spiral process model. This hybrid testing technique in collaboration with spiral process model enhanced defect finding ratio in each phase of spiral process model and cope the overall project and product risk.

For this tenacity a Questionnaire has been conducted from different software houses and responses were collected from different persons such as testers, designers etc. after that statistical tool named as SPSS has been used and chi square test has been applied to observe that whether the Hybrid testing technique and spiral process model are dependent or independent.

A. Pair Programming:

Programming is a technique for agile methods. In this technique, a pair of programmers work together to develop a module of the product. One is the driver that writes the code, the other is the observer, and navigates the entire encoding for early detection of errors. In programming technology, a pair of programmers work together to develop a work piece. They work together, sharing innovative ideas with each other, improving at the design stage, designing and coding algorithmic algorithms to define design and coding. There are three variations of pairing which are described as follows:

1). Expert–expert

The pairing of two specialists is called expert-expert pairing. In this technology, the two experts are more focused on high-quality productivity. The two experts work together and propose new ways to solve this problem, as both are responsible for the reliable development of the product.

2). Expert–novice

In expert-novice pairing techniques, there is an expert and a beginner. Expert guidance novice. More new ideas have been introduced in this technique. Because the novice by thinking about his fresh ideas and fresh knowledge to give new ideas. The expert also put forward his ideas according to his established practice. However, in this technology, the novice may hesitate in the face of experts, and follow the experts.

3). Novice–Novice

In the novice-novice pairing technique, two novices work together. This pairing technique yields better and more significant results than the two novice work alone [5]. It can be seen that programming techniques are an effective and reliable technique for developing software with improved quality. This has a lower defect rate for the software produced by the programmer, less time boxes and more efficient cost [6]. As a result, programming techniques reduce the origin of errors in the RSSM. This makes the test more reliable and less costly.

B. Acceptance Test Driven Development:

Acceptance testing Driver development is a method of developing software that focuses on communication between all stakeholders, such as customers, developers, and testers. This approach also includes several other approaches, such as example driven development (EDD), specification examples (SBE), story test-driven development (STDD), and behavior-driven development (BDD). Acceptance Test Driven Development (ATDD) follows the characteristics of Test Driven Development (TDD). It emphasizes the collaboration between developers, testers, and customers. Acceptance Testing Acceptance testing in Test Driven Development (ATDD) is designed from the end user's perspective. These tests examine the observable effects of the software. It attempts to specify the software output in response to the input. It shows how the software switches its state. These tests also show how the software interacts with other software interfaces [7]. Test-Driven Development (TDD) helps software developers develop high-quality and reliable software. Acceptance Test Driven Development (ATDD) helps provide products that have all the features and functionality

that customers require in their software. This ATDD technology enhances the RSSM test capability.

1). Advantages of ATDD:

Acceptance Test Driven Development (ATDD) involves testing the customer during the design process. Moderators try to improve their understanding of the customer by using brainstorming techniques and visualization techniques such as Mind Mapping and Story Mapping. The results of brainstorming and visualization techniques were interpreted by the testers to design the executable test. The programmer begins writing code to automate the test. Then use navigation tests and boundary tests to explore behavioral aspects. Once the automatic test passes, the probing test testers are used to clearly define the desired and undesirable behavior.

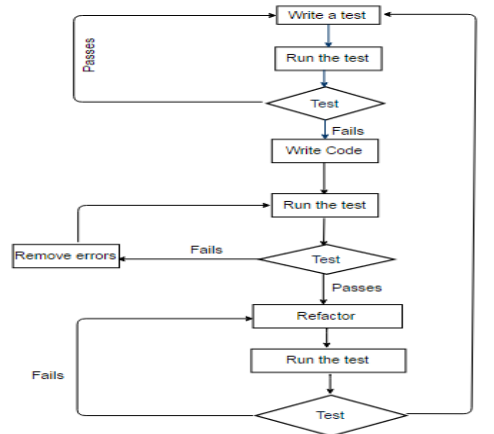


Fig. 3. TDD Flow Chart

C. State Transition and Exploratory Testing Technique:

These techniques improve test capability State transition testing is a black-box testing technique in which a change in input condition results in a state change in the application under test (AUT). Gray box testing in RSSM can replace black box testing and white box testing, respectively. In this regard, it can be said, RSSM is a revolutionary model with the latest test technology to guide project managers to develop ERP software at a defined time effective quality and lower cost.

D. Phases of Methodology:

1). Planning Phase: Planning is the first stage of the spiral process model. It takes into account potential market and other factors to estimate the expected revenue to determine the feasibility of the product. Feasibility report to determine the feasibility of product development. Collect requirements at the planning stage. Requirements such as "BRS" are

"Business Requirements Specification" and "SRS" is "System Requirements Specification". Tests are performed at this stage to eliminate defects at an early stage.

2). Risk Analysis: In the risk analysis phase, the process of identifying risks and alternative solutions. Generate prototypes at the end of the risk analysis phase. If any risk is found during the risk analysis, an alternative solution is recommended and implemented. Tests are performed at this stage so that the error cannot be moved in a further stage.

3). Engineering Phase: At this stage, software development, and testing at the end of the phase. Therefore, at this stage, development and testing are complete. Tests are performed at this stage to eliminate defects at an early stage. The test technology cooperates with ATDD, BDD, and state transition testing to meet test objectives.

4). Evaluation phase: This phase allows the customer to evaluate the project's output before the project proceeds to the next spiral. During the evaluation phase, the products produced are deployed to the end user for implementation. Training programs are planned for training end-users and software maintainers. Beta testing is used to test whether a product meets the end user's expectations. Tests are performed at all stages of the RSSM, so the software is reliable, efficient and defect-free.

and spiral process model are dependent or independent and the results shows that by applying proposed methodology i.e. RSSM testing process has been improved for the development of ERP software which gave efficient results. The effective performance of a software application or product needs to be tested. Software testing is very important to ensure that the application does not cause any failure because it can be very expensive in the future or late in the development. Traditional process models are designed to work in a linear fashion, and it is difficult to manage rapid changes using these traditional process models. Although there is not enough time in the agile method to test the software and lack documentation, this becomes a major issue in the maintenance phase. To meet this need, custom testing procedures are needed to test products with limited time boxes, resources, and to successfully achieve the end user's goals. Unfortunately, no process model can claim to overcome all of the identified problems. No one paid attention to the software testing phase to produce Pakistani better quality ERP software. In Pakistan, software developed through custom process models cannot produce good quality software because software factories are less concerned with software testing. The proposed Revolutionized Spiral Process Model introduced Progressive testing techniques for quality assurance of ERP software products.

V. Conclusion

In Pakistan, Testing and implementation of ERP software projects facing product and project risks due to its initial stages. All these risks whether related to project or product must be tackled before these risks became the threat for software success. Better evaluation of ERP software required road map for Planning, identifying risks and testing activities. The proposed Revolutionized Spiral Process Model introduced Progressive testing techniques for quality assurance of ERP software products. ERP software plays a vital role in businesses to handle all aspects of that business. The Product and project risks can be managed by using latest software testing techniques. As Spiral 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 will be able to develop quality ERP Software.

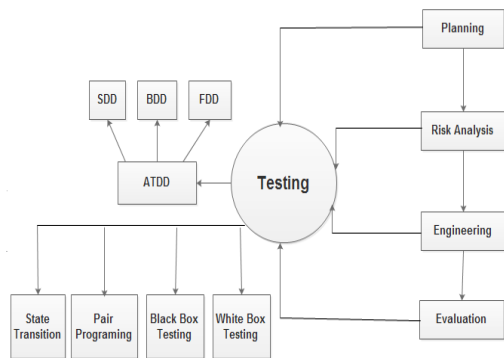


Fig.4. Illustrates the Revolutionized Spiral Process Model (Proposed Model) with testing specification.

IV. Results of the study

In order to take results from different persons working in software house a Questionnaire has been conducted and responses were collected from different persons such as testers, designers etc. By using chi square statistical method has been applied to observe that whether the Hybrid testing technique

VI. Future Work:

Progressive testing technique in collaboration with spiral process model gave efficient results. As I have discussed above I will present this draft version for testing. Furthermore, I think that there is 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] Shehab, E., Sharp, M., Supramaniam, L. and Spedding, T. 2004. Enterprise resource planning: An integrative review. *Business Process Management Journal*, 10 (4): 359-386.
- [2] 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.
- [3] Devi, R. 2012. Importance of Testing in Software Development Life Cycle. *International Journal of Scientific & Engineering Research*, 3(5): 1-5
- [4] Bassil, Y. 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (IJET)*, 2(5):2049-3444
- [5] Madina, S. 2012. Effective Pair Programming Practice- An Experimental Study. *Journal of Emerging Trends in Computing and Information Sciences*, 3(4):471-479
- [6] William, L. 2000. Strengthening the Case for Pair Programming. *Journal of IEEE software Engineering*, pp: 19-25.
- [7] Pugh, K. 2012. Introduction to Acceptance test-driven development. A Net Objectives Essential White Paper. Pp:3-8