

# Revolutionary Testing using Extreme Programming to develop ERP Software in Pakistan

**MUHAMMAD NADEEM ABBAS**

Department of Computer Science, Government College University Faisalabad, Pakistan

**UMER SARWAR**

Department of Computer Science, Government College University Faisalabad, Pakistan

**WAQAS AHMED**

Department of Computer Science, Virtual University, Lahore, Pakistan

**NAEEM ABBAS**

Department of Software Engineering, Government College University Faisalabad, Pakistan

---

## Abstract

*In Pakistan, Evaluation of ERP software projects facing product and project risks due to its beginnings. All these risks must be nipped in the bud, before these risks became the hurdle for software success. Better testing and implementation of ERP software need Planning for identifying risks and testing activities. In this research paper a revolutionary testing technique in collaboration with XP was introduced for the evaluation of ERP software. The XP identified different ERP software risks arise in requirement phases by involving customer and testing technique tackled with these risks. This revolutionary testing technique involved (TDD), (ATDD), (TFD), (TLD), (BDD), Pair Programming and gray box testing. Resultantly a Revolutionized XP Software model RXP was developed with better testing solutions for the successful development of ERP software in Pakistan. To get a larger nationwide view on the RXP (across Pakistan), a survey was conducted through a questionnaire and the results were analyzed to show whether the provided solution meets the organization goals.*

**Key Words:** 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 Extreme Programming (RXP).

---

## Introduction

Now-a-days, the business organizations are facing challenges in the market, as they are competing their competitors in their daily routine. Every company wants to promote its products among customers and manage its sales. That's why; they are facing abruptly changing requirements to meet their goals successfully. In such situations,

Information Technology has lent to the world of business application. With the passage of time, new technology with more functionality is being introduced in the market. Software is part and parcel for such technologies to do work. Reliable error free software can guarantee this notion that the production will long lasting error free and will confirm the user requirements. ERP is the integration and control of the entire organization of all business processes enterprise-wide information systems.

ERP is "bundled business programming framework that permits the organization by providing a total, integrated solution for the organization's information processing requirements, efficient management and operative use of assets (material, human resources, finance etc.). The product is anything but difficult to utilize, if appropriately actualized, incorporated data on every one of the elements of the stream of data "over the association into a solitary bundle to a typical database. In this way, it permits simple and moment access to data about stock, item or client information and former history of data (Shehab *et al.*, 2004). A process of software validation and verification is known as Software Testing. 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. Software Testing is essential to detect errors in the early stages of Software Development Life Cycle (SDLC). It ensures that the product is error free. It paves way for the developers to anticipate about the Quality, Time and Cost of the product in advance (Saini, 2013). In this modern era, the ERP Software is being used by large organizations to fulfill their volatile requirements. An ERP Software consist of several modules including manufacturing process, finance, human resources management, supply chain management, customer relationship management, depository management etc. Such software can be implemented in all types of business and organizations. The testing of an error prone software and the method used for testing economically affect the overall cost of a product. Software Testing is an integral part of every software development life cycle and it is required to introduce new testing techniques to make this process more efficient. Every software house chooses different testing techniques according to their requirements. It is essential to find out defects during early phases of software development life cycle to develop a successful product with less cost and high quality within limited time box. Testing phase is conducted to improve various parameters of the product such as reliability and performance (Devi, 2012).

**Software Testing**

Software testing may be defined as a process that is performed to evaluate the software to identify errors and 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.

Quality Attribute of Testing Technique

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

**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 is done 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 product documentation at this stage.

### 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.

### Testing in XP

Testing in agile method continuous until a project finishes. As there is not enough time for detailed testing, each sprint is separately developed and tested before integration testing of all sprint. An agile team contributes its expertise. Testing takes place in real time focuses on quality of the software. Due to less time, tester cannot plan a detailed testing criterion, that's why only essential corrections are done. Every Sprint is designed, developed, tested and deployed before moving to the next Sprint. Then all Sprints are tested on the integration level. Testing in Extreme Programming (XP) conducted throughout all phases. Testing throughout all phases of Extreme Programming (XP), have become a major feature of this methodology. This feature makes the difference among various software development methodologies. By adopting this feature in software development process, a project manager becomes able to develop a product successfully to deploy with fewer defects. Testing artifacts produced at various phases of development process can help in maintenance phase in future after deployment. The execution of tests and prototype give the surety about mitigation of all possible risks involved in the project. Software Development Life Cycle (SDLC) provides guide line and a systematic approach to the developer of the software to deliver software within limited time duration with high quality. There are different methodologies which claim to do so. All the development methodologies are comprises on phase of requirement engineering to collect requirements, analysis of business to fulfill all end user's requirements, design phase, implementation phase and Quality Assurance Testing Phase.

It is required to introduce a hybrid model of all methodologies, because is not even a single methodology which can claim to thoroughly grip all aspects with perfection (Mishra, 2013).

### Revolutionized Extreme Programming

The Revolutionized Extreme Programming (RXP) was designed to present an efficient methodology which could overcome all the problems efficiently faced by all methodologies mentioned above. Agile does not pay attention towards testing and documentation of the product development process. Agile Management always compromise on the quality of the product. Water fall model pays attention towards documentation but not towards testing phase. In water fall model, the testing phases become too late, and due to this problem these models become risky because at the end after consuming effort and resources the product can fail. Testing in Water Fall Model becomes costly. There is no customer involvement throughout the development process in water fall model. Extreme Programming (XP) is a tailor able and iterative, Incremental methodology. Due to these characteristics, it becomes a good choice to develop Enterprise Resource Planning Software in Pakistan. In the Revolutionized Extreme Programming (RXP), latest Testing strategies are being introduced which makes the Traditional Extreme Programming (XP) more efficient by increasing the customer involvement, developing comprehend able documentation can be used for maintenance phase and by using extreme testing techniques such as Acceptance Test Driven Development and Pair programming etc. It is necessary to collect authentic requirements of the customer in the Planning Phase. To enhance this capability Revolutionized Extreme Programming (XP) used Acceptance Test Driven Development (ATDD) technique.

This technique 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 Methodology. 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. In the proposed software Pair Programming is also being introduced to enhance the testing process. It is a 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 Extreme Programming (RXP). This makes the testing more efficient and less costly. State Transition Testing and Exploratory Testing Techniques can also help to improve the Testing phase. While the software is being tested, the tester learns a thing that together with experience and creativity generates new good tests to run. State Transition testing, a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. In other words, tests are designed to execute valid and invalid state transitions. Grey Box Testing in Revolutionized Extreme Programming (RXP) can replace the working of black box testing and white box testing separately. In this regard, it can be said that Revolutionized Extreme Programming (RXP) is a revolutionary methodology 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.

## Research Methodology

The purpose of this research was to study existing 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 XP. This hybrid testing technique in collaboration with XP enhanced defect finding ratio. 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.

## Extreme Programming (XP)

Extreme Programming (XP) is an engineering methodology consisting of practices that ensure top-quality, focused code. XP begins with four values:

### Simplicity

Simplicity means that the software is developed using the simplest possible design and constructs, but it means more: simplicity pervades the entire process. XP eliminates, as much as possible, the unnecessary elements of building software. One of our rules is “You aren’t going to need it,” which reminds us to add software or process only when we really need them, not in anticipation of the need.

### Communication

Communication is a key to rapid development and to customer satisfaction. XP embodies communication with a focus on simplicity: Use person-to-person communication instead of written documents wherever possible. It also stresses continual communication between the customer and development team members by having an on-site customer while development progresses. The on-site customer decides what will be built and in what order.

## Feedback

Feedback is important for any development process, but when you are trying to eliminate everything you can, you need feedback to be sure you're on track. Software testing is a major source of quality feedback in XP but we include resource, scope, and time feedback as well.

## Aggressiveness

In moving forward is possible when you take the simplest possible approach and employ a process high in communication and feedback. It means being honest about what you can and cannot do.

## Practices of Extreme Programming (XP)

Twelve XP practices support the four values. These practices tend to keep the team on track while they build up a grasp of the principles that form the basis for the practices. In XP, every contributor to the project is an integral part of the whole team formed around a business representative – the “Customer”. There are twelve XP practices that the team should depend on when adopting XP.

### The Release Planning

A practice in which customer present the desired features to the programmers in the team who in return estimate their difficulty.

### Iteration Planning

A practice whereby the team is given direction every couple of weeks building a software in 2-weeks “iterations” and delivering running useful software at the end of each iteration.

### Small Releases

XP teams put a simple system into production early, and update it frequently on a very short cycle. The team releases running, tested software, delivering business values chosen by the customer. The most important aspect is that the software is visible and given to the customer at the end of every iteration. This keeps everything open and tangible.

### Testing

Also known as customers test XP teams focus on validation of the software at all times. Programmers develop software by writing tests first then software that fulfills the requirements reflected in the tests. Customers provide acceptance tests that enable them to be certain that the features they need are provided. The best way for success is that once the test runs the teams keep it running correctly thereafter.

### Pair Programming

Programming is a technique for agile methods. In this step, 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 works 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.

## 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 (Pugh, 2012). 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 RXP test capability.

### 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.

### 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 and XP are dependent or independent and the results shows that by applying proposed methodology i.e. RXP 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 XP introduced Progressive testing techniques for quality assurance of ERP software products.

### 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 XP 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 XP 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.

## Future Work

Progressive testing technique in collaboration with XP 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

- Devi, R. 2012.Importance of Testing in Software Development Life Cycle. *International Journal of Scientific & Engineering Research*. 3(5): 1-5
- 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.
- Shehab, E., Sharp, M., Supramaniam, L. and Spedding, T. 2004. Enterprise resource planning: An integrative review. *Business Process Management Journal*, 10 (4): 359-386.
- Madina, S. 2012. Effective Pair Programming Practice- An Experimental Study. *Journal of Emerging Trends in Computing and Information Sciences*. 3(4):471-479
- Mishra, A. 2013. A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. *International Journal of Advance Research in Computer Science and Management Studies*. 1(5): 64-69
- William, L. 2000. Strengthening the Case for Pair Programming. *Journal of IEEE software Engineering*. pp: 19-25.
- Pugh, K. 2012. Introduction to Acceptance test-driven development. A Net Objectives Essential White Paper. Pp: 3-8