

## PM WORLD TODAY – FEATURED PAPER – DECEMBER 2009

## Test Driven Development in Extreme Programming

*By Pavan Kumar Gorakavi*

## 1. What is Extreme Programming?

Extreme programming is a software engineering methodology which has gained its importance in the arena of agile software development methodologies. Extreme Programming was initially formulated by Kent Beck.

Extreme Programming is a disciplined methodology which stresses on customer satisfaction. It basically underlines on the concept of 'deliver when needed'. Unlike conventional practices, agile practices maintain re-factoring cost minimal. Agile process primarily focuses on high interaction, working model, customer collaboration, regular feedbacks, and flexibility for changing requirements. Extreme programming encourages high degree of interactions between team players, which includes developers, testers, managers, business owners and others. This practice advices the team to possess a lucid information system. Extreme programming emphasizes just not on testing, but testing well. Test cases and tests are created at all stages of coding. As Extreme programming provides early feedback, it will reduce failure cost. It provides a flexible environment to both customers and developers for changing requirements.

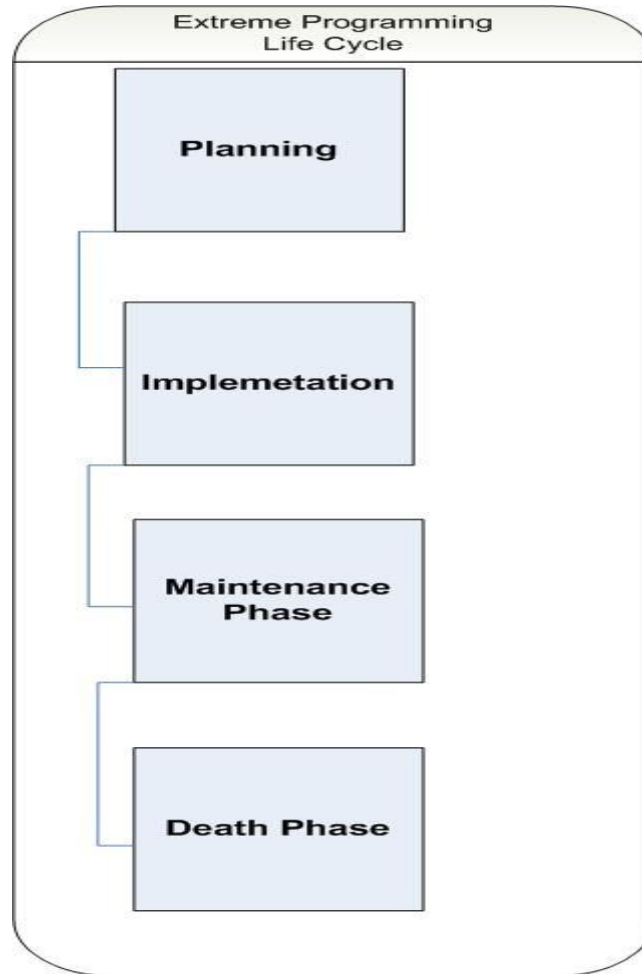
Extreme programming needs effective team with size of 5-20 team players. Small teams handling smaller module yields an efficient output in XP practice. XP address the problem of huge development cost by providing early prototype feature. These practice advices developers, testers, managers, customers and other key players to have periodic interaction. Extreme programming can be efficiently applied when team environment is highly communicative, has small flexible groups, de-fragmented modules, proper testability and productive workgroup.

## 2. Life cycle of Extreme Programming

Extreme Programming can be implemented by disciplined methodology which focuses primarily on customer satisfaction. Life cycle of extreme programming includes Planning, Iterative development phase, maintenance phase and death phase. In planning phase user stories are written, Iterative implementation planning is performed, efforts are estimated, and priorities of the features are decided. In Iterative development phase, functional modules are developed using standard software development life cycle process: analysis, design, development and testing.

This is iterative in nature. 'Pair Programming' is a unique feature associated with XP. Iterative development phase includes continuous reviews, continuous integration and quick feedback. Maintenance phase includes new small releases with customer experience. Once Extreme

programming reaches a maturity level where there are no user stories, then product life cycle reaches a death phase. This can be illustrated in figure 1.

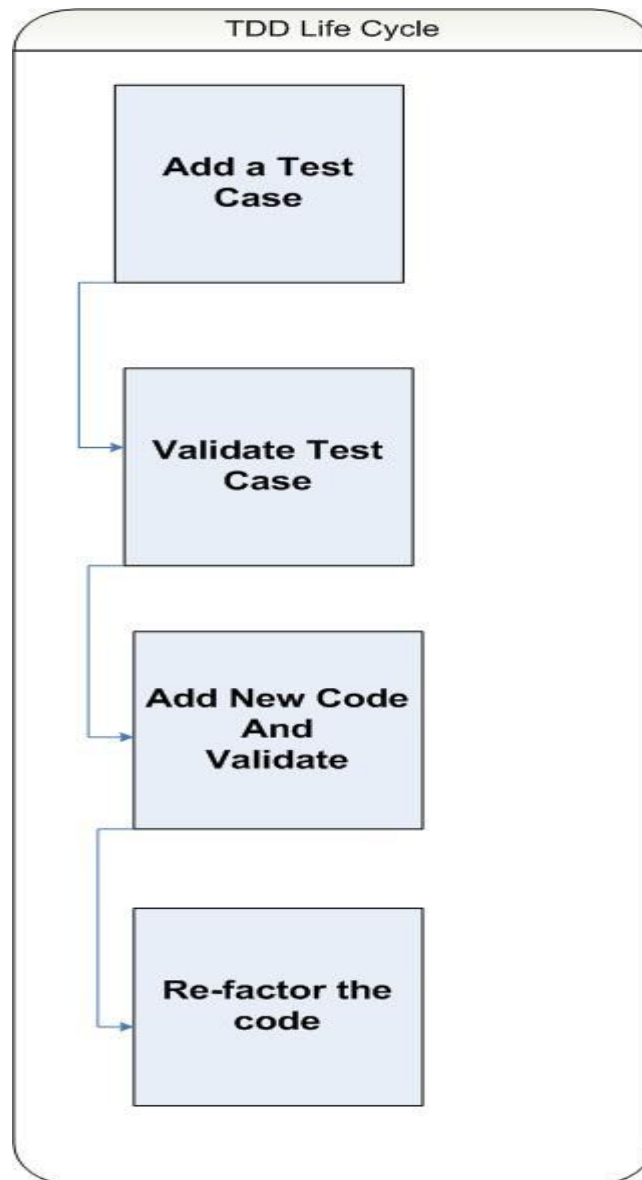


**Figure 1: Life cycle of Extreme Programming**

### **3. What is Test Driven Development?**

Test Driven development is a software development practice widely used in Extreme Programming. Test driven development advises to build the software in small increments. This approach widely relies on the tests to drive the design of the module. It is a simple evaluation of software from test cases and can also be proclaimed as test-driven approach. Test driven development advises to create automated unit test cases that define design and code before writing the code itself.

Test Driven Development is a unique practice which relies on unit testing and refactoring. Test Driven Development is a methodology which proclaims incremental steps: designing a unit test case, developing a functional module for the unit test case, make the test case pass, and re-factor the code. Test driven development approach is illustrated in Figure 2.



**Figure 2: Life cycle of TDD**

***Step 1: Add a Test case***

TDD recommends initiating the process by creating a unit test case. TDD assumes an automated unit test is available during its implementation. The test written should be initially failed as there is no corresponding functional modules been developed. This approach of writing test cases prior to coding develops developer's ability to understand requirements, features specification and come with better design.

***Step 2: Test cases failed for newly added code***

As TDD proclaims to write test cases before development of the code, all newly added test cases must fail. This validate that test driven approach is working correctly.

***Step 3: Developing functional code***

Functional code will be developed in this phase for the test cases which are written in step1. The functional module developed may or may not be fit in the existing unit test case design. A good functional module only addresses the test case which was developed in step1, but no further functionality is predicted.

***Step 4: Re-run the test cases***

After developing the newly added functional module, we try to validate the test cases by re-running the unit test cases. This allows the development team to verify the quality of the newly developed functional modules.

***Step 5: Re-factor the code***

After successful implementation of functionality, developers need to perform periodic refactoring of the code for easy maintenance of software.

**4. Benefits of TDD**

Development mechanism can gain following benefits by implementing Test Driven Development approach.

- 1) Better and simple design resulted because of test driven approach.
- 2) Test driven approach generally reduces complexity of the design.
- 3) As development is based on unit test cases, functionality is developed in small intervals and incremental nature.

- 4) Test driven approach can yields a better communication flow among the team.
- 5) Project velocity can be calculated based on the consideration of number of test cases succeeded.
- 6) Test driven development approach relies on continuous execution of unit test cases, this facilitates constant regression.
- 7) TDD encourages better understanding of requirements and better analysis by the developers.

## 5. Conclusion

Some times there is a myth among management folks that TDD approach is considered as a waste of time. In practice, Implementers finds TDD produces a qualitative product when compared to conventional methodology. Test driven development approach is difficult in scenarios where complex functional test cases are required to validate functionality. In practice, TDD is a good approach to follow if the project permits.

## References:

1. Building your project using XP, Pavan Gorakavi, asapm.
2. Gold, Russell, Thomas Hammell, and Tom Snyder. Test Driven Development: A J2EE Example. Apress. © 2005
3. A. Anderson, Beattie, Ralph, Beck, Kent et al., "Chrysler Goes to "Extremes","
4. *Distributed Computing*, vol. October 1998, 1998, pp. 24-28.
5. L. A. Williams and R. R. Kessler, "All I Ever Needed to Know About Pair Programming I
6. Learned in Kindergarten," in *Communications of the ACM*, vol. 43, no. 5, 2000.
7. Stephens, Matt, and Doug Rosenberg. Extreme Programming Refactored: The Case Against XP. Apress
8. Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, Juhani Warsta. Agile software development methods, VTT Publications.

**About the Author:*****Pavan Kumar Gorakavi****Author*

**Pavan Kumar Gorakavi.** M.S, M.B.A, G.M.C.P, C.A.P.M is currently working as Sr. Software Developer in Dallas, TX. He is settled in Dallas with his small family [wife Swapna Gorakavi and son Anish Gorakavi]. He is VP - programs for asapm-young crew. Pavan did his bachelors in computer science from Jawaharlal Nehru Technological University in India and masters in computer science from Lamar University. He did his MBA from University of Texas at Dallas and GMCP from Southern Methodist School in Dallas. Pavan holds SUN, IBM and PMI certifications. Pavan Gorakavi is the author of books on '*Artificial Intelligence*' published by Rahul publications - India, and '*Digital Electronics*' published by Subhash publications, India. His research interests are Artificial Intelligence, Agile methodologies, and Software development in Ada, Prolog and Java. You can reach Pavan at [gorakavi@gmail.com](mailto:gorakavi@gmail.com) .