International Journal of Information Technology and Business Management
29th August 2012. Vol.4 No. 1
© 2012 JITBM & ARF. All rights reserved

ISSN 2304-0777 www.jitbm.com

A DESCRIPTIVE COMPARATIVE STUDY ON XP VS WATERFALL METHODOLOGY

M. Varusai Mohamed
Lecturer in Computing Studies, Gulf College
Muscat, Sultanate of Oman.

S. Balaji
Lecturer in Computing Studies, Gulf College
Muscat, Sultanate of Oman.

Abstract: Many software companies are now developing object oriented software and the people working in the IT companies must have the idea of object oriented CASE tools. Most of the projects in the IT Company go failure due to improper communication with the client. So the communication is the most important value in the IT Company for the project success. Even though if the communication is good with the customer, sometime the customer do not like the project deliverables due to its complexity in programming. In this case, the simplicity is important for the project success and to make the customer satisfied. The simplicity provides the environment to update the projects easily for its future enhancements. It is important for the IT companies to choose a suitable methodology for their project success. Now a day getting requirements from the customer for the system development is a big process in the busy world and providing simplicity in the project deliverables is not easy. But some IT companies are still using some of the traditional methodologies like water fall for the project life cycle. This paper is a comparative study between XP methodology Vs water fall methodology based on some values. This paper describes the importance of XP methodology over waterfall methodology and explains how XP methodology stresses customer communication during the system development.

Keywords: Waterfall [10], XP methodology [7],

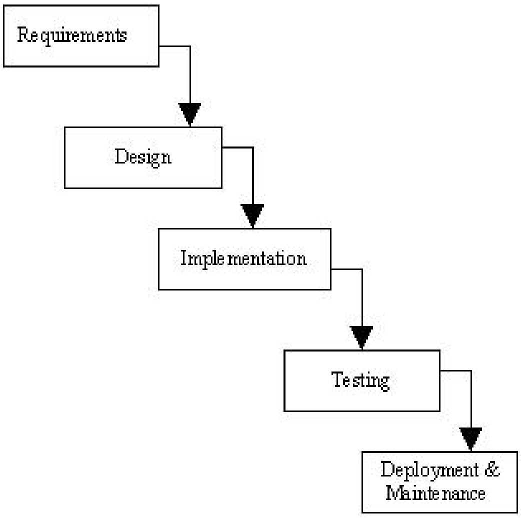
INTRODUCTION

A Software Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, A number of software development life cycle (SDLC) models have been created: waterfall, spiral, V-Model, rapid prototyping, incremental, and synchronize and stabilize. Waterfall model is the Sequential development model. a sequence of stages in which the output of each stage becomes the input for the next. Extreme Programming (XP) [7], is a software engineering methodology, the most prominent of several agile software development methodologies. The main aim of XP is to lower the cost of change. This means that the cost of changing the requirements at a later stage will be high. By applying XP, a system development project should be more flexible with respect to changes.

WATER FALL MODEL

It is a linear sequential model with many stages to software development. It is a stage by stage approach to build the project. It has the stages of requirements, design, implementation, verification and maintenance.

Figure 1: Water fall model life cycle



1. Requirements

All possible requirements are captured in this stage. The requirements are specified as a set of functions and constraints what the end user expect from the system. The gathered requirements from the end user are analysed for their validity. Finally a requirement specification document is created.

2. Design

Before coding all the requirements are translated into system design. Design helps to specify the overall system architecture and to identify the requirements of the system. The design specification is prepared which supports for the next stage.

3. Implementation

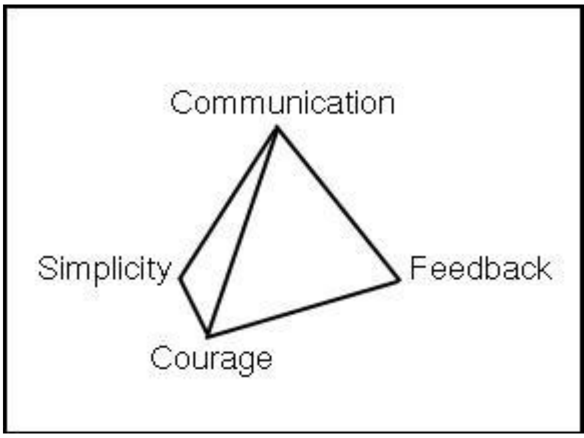
Development phase is started after the design phase. The design documents are translated into functional program. The programs are developed based on the algorithms, diagrams designed in the design phase. The design phase produces a functional specification and a user interface specification. These documents are to be signed off first. Then process of building the application starts. The whole projects are divided into many modules. A program is written for all the modules one by one. Each module is tested for the output. The following diagram shows the modularity of the system and explains how the code is integrated.

Figure 2: Modularity and Integration

System	**Module1->code->Build->Test** **Module1->code->Build-> Test**	**Integrated**

Development is performed by a team consisting of many programmers, interface designers and other specialists. Finally implementation of the product is done.

- There is a target for each programmer in a team.
- It is excellent for large projects.
- It is used in the projects where the requirements are well known.



- Documentation is done in each phase which helps to new programmers.
- It uses version control to track the changes in code.

4.Testing

Each unit is tested for its functionality by unit testing. These units are integrated into a complete system during integration phase and tested to check if all modules/units coordinate with each other. Then the system is tested as per the specifications. The system is delivered to the customer finally.

5. Maintenance

Some problems are identified after practical use of the system and solved after deployment of the system in this stage. It is a never-ending phase. The problems arise from time to time and need to be solved.

WHY NOT WATERFALL

It has various deficiencies, particularly the time taken to deliver a working system and the inflexibility of the approach to requirements change. Once a stage is completed, we cannot go back to change in the previous stage. The problem is fixed in the next development life cycle. The testing is done at the end of the development. If any problems in the results the entire program takes changes. If there is any delay in the release, the customer gets dissatisfied.

XP METHODOLOGY

Extreme Programming (XP) [5] is an agile software-development methodology. Agile methodology has many methods. Some principles of agile methodology are referred by this methodology. It is a light weight methodology. It has dimensions and practices. Dimensions are not adjustable but practices are adjustable depends on the situation of the project.

XP has four dimensions which are to be implemented correctly to improve the software project. They help to enable the team to see where they are and to tune the practices depends on the project situation. Some of the practices are pushed to the extreme level by extreme programming.

1. Dimensions of XP

Figure 3: Dimension on XP

- **Communication**
- **Feedback**
- **Simplicity**
- **Courage**

1.1 Communication

Most of the projects are failed due to communication failure with customers. It is unavoidable situation with some customers but XP stresses the customer communication and give suggestions to improve communication. Communication is important for quick development of the project.

1.2 Feedback

Changing requirements are expected situation with the customer. There is some acceptable risk in changing requirements in real-time development. Getting feedback gives more confidence to the team. Because mistakes are immediately visible and are corrected. The customer sees his requirements corrected within weeks. Programmers update their changes every few hours. They review code and test the results immediately. The customer gets new versions every month.

1.3 Simplicity

It focuses on the design process. The system with the simplest design that supports current functionality. The structure of the design should organize the logic in the system. Proper design is important for long running of the system because it avoids the complexity in the project and project halt.

1.4 Courage

Courage is important for programmers. It is required at all levels. Emotions of people are varying from one to another. XP methodology uses team interactions which provide emotional support between team members in difficult times. This method helps build the bonds of trust by repeatedly exposing people to small successes. XP Programmers are able to overcome their fears and take small steps to promote courage.

BEST PRACTICES OF XP METHODOLOGY

Dimensions of XP are implemented by practices. The whole team is worked to implement the simple practices. Some of the practices are taken by XP methodology to an extreme level. The following are some of good practices and it's environment.

1. Simplicity

XP focus on the solutions to be simple for the known requirements. Simplicity is not easy. Make the design simple. Every project is divided into many components. But some components of the projects are reused in many places. If the design is simple, it is very easy for the programmer to reuse, modify, review and test the program. One of the scientific goals of agile methodology about the project is its simplicity. The developer's objective is to keep the solution very simple which avoids extra code and complexity. The programs pass all the tests. There is no duplicate code and fewest possible classes and methods in program.

2. Refactoring

It restructures the system because it eliminates duplicated functionalities. It improves the communication and enhances simplicity and flexibility. It can happen constantly. Physical environment is important factor to carry out this task. Due to this, the quality is increased in the software production and unpredictable requirements are refined. It is depends on the team size of the project. This gives team members full confidence. It is easy for the programmer to change the system later in the development.

3. Code Review

This is a practice in XP which involves restructuring the system by eliminating duplicated functionalities, improving methods of communication and enhancing simplicity and flexibility. The developers throw away code that is not quite correct.

4. Pair Programming

The code is written by two programmers using one workstation. All production code is written by pairs. The code is owned to two programmers and anyone can change any code. Programmers can share their ideas and experiences to solve the problem. It provides confidence to programmers to add or to change the system. The programs are collected each day for the integration of the system. It provides discipline to test and refactoring always.

5. Testing

Verification and validation testing are occurred to test the individual components and the integrated solution. Test cases are written earlier stages whenever getting the requirements from the customer and are implemented after code to evaluate the requirements. All tests are written in a test framework such as Unit and hence become fully automated. Mostly unit testing is done for all the modules of code and system testing is done when all the modules are integrated. Functional tests are written by the customer with the help of the tester. The acceptance testing is performed with customer to get his feedback. The feedback is given immediately to the development team to fix the issues. The user manual and documentation are produced. In this methodology, testing becomes specification. It gives confidence in the system. It gives courage to change the system. The testing is done starting from the day one.

WHY XP METHODOLOGY?

It highlights the importance of good communication between team members and between developers and users. Programmers are motivated by producing quality work. It embraces the changes in the system development life cycle. It has inherent capability for improving efficiency of business software development. It improves the productivity in the business software development. It emphasizes to deliver quality software. The development is as a conversation and the writing code is the documentation in XP methodology.

WHO BENEFITS FROM XP?

Customers	Programmers
Get most business value first	Get clear requirements & priorities
Get accurate feedback	Can do a good job
Can make informed business decisions	Can make technical decisions
Can change their mind	Don't work overtime

CONCLUSIONS

IT companies are now developing object oriented system by using object oriented software which needs simplicity in its design and easiness to change the system. The companies are developing medium size projects more. The technology is changing day by day. Now the people are interacting with object oriented applications in their hand held devices. So changes are always there in the technology and also in the development. The XP [7] methodology is best suitable methodology for the object oriented system development where the changes and communication are needed in the project releases. XP [7] methodology solves the following risks in the system development.

- Risk of Schedule Slips
- Risk of Cancellation of Project
- Risks of Misunderstanding Business
- Risk of Business Change
- Risks of Additional Unwanted Features

XP methodology addresses the above problems by keeping communication with customer during system development.

REFERENCES

1. Ambler. (2002). *"Agile Modeling: Effective Practices for XP and RUP.".*

2. Boehm, B. &. (2004). *Balancing Agility and Discipline.*

3. Highsmith., A. C. (2002). *Agile Software Development.*

4. Kent Beck, M. F. (2000). *"Planning Extreme Programming".*

5.Little, J. (January 17, 2011). *Impediment Management. Agile Consortium.*

6.McBreen, P. (2003). *Questioning Extreme Programming. .*

7.Rosenberg, M. S. (2003). *Extreme Programming Refactored.*

8.Royce, W. (1970). *"Managing the Development of Large Software Systems".*

9.Sliger, M., & Broderick, S. (2008). The Software Project Manager's Bridge to Agility.Addison-Wesley. p. 46. ISBN 0-321-50275-2.

10.Weisert, C. (2003). *Waterfall.*