

## SOFTWARE DEVELOPMENT AND ITS ITERATIVE DELAY

Syeda Rubab Zahra\*

Sidra Mushtaq\*

### Abstract

The adaptation of software development methodology is essential to build software. Typically there are two models for development: traditional and agile method. These models have further different methodologies. The agile methodologies are used to make the development process flexible. Here in this paper I have discusses waterfall and extreme programming (XP) model. The extreme model overcomes the problems of waterfall model. It produces the software timely under time box technique but still there are some problems in XP e.g. incomplete iterations during time box so this paper discusses some reasons and their solution.

**Keywords:** Traditional, Agile, Extreme Programming, Time boxing, Iterations.

---

\* Department of computer science, University of Agriculture, Faisalabad, Pakistan

## 1. Introduction

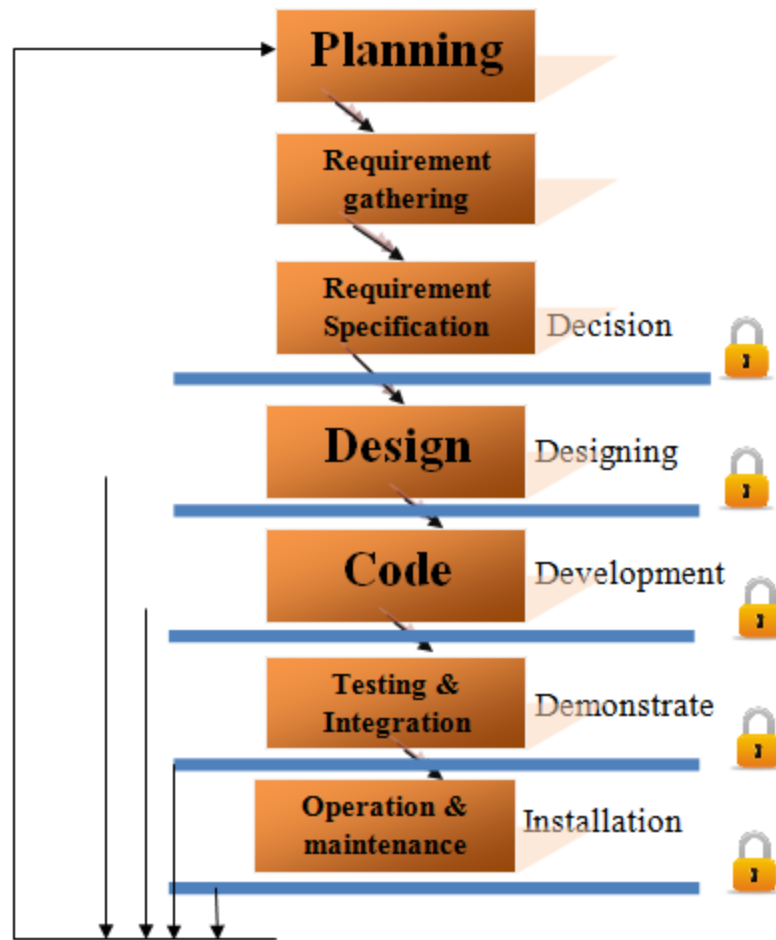
Selection of Software Development Life Cycle (SDLC) methodology is most important because it is a process to build or maintain software products [7]. Normally it includes many phases from preliminary investigation to software testing, implementation and maintenance. Each methodology forms a framework for scheduling and handling the whole process of development. But it is a cruel and tiresome job to govern the development process to provide a high quality end product that involves high degree of integrity and robustness and meet user requirements and acceptance. Thus a systematic development methodology is essential which must understand the complexity and scope of the required product and achieve the successful system. [3]

Here in this paper and compare the two different models of software development will be discussed as i) Waterfall and ii) Extreme Programming. And the reasons will be described that why and how their working is different and which is better and how it can be improved further.

### 1.1 Waterfall

The waterfall model is a traditional sequential model in which each stage of development cycle has been locked when development team moves to next stage. If changes required in previous stages then developer must have to start work from initial stage even if he is on later stages. It strictly requires that the system requirements must be defined and documented at the project beginning. [3]

This framework is divided in five major phases. The first phase is to take decision about project. In this phase the required project is intended to make then requirements are



gathered and specified for development. Cost is estimated. Then the client approved this documentation and development process starts. The next phase is architectural design and then the coding / actual development of software begins. After developing code in components, they are tested and integrated. And in last installation phase the system has installed and operated on user site and maintained when required.

### 1.1.1 Advantages of Waterfall Model

As waterfall model is traditional software development methodology and its idea was depicted from other engineering domains e.g. civil engineering, so it uses the same sound rules as those used for a good effect- design the right and then build it right. It provides all requirements gathered from and signed off by the user. Complete documentation is another good feature which provides a strong base for future developments and maintenance. Waterfall model usually focuses on completeness, on correctness and functionality. [2] Development environment under waterfall model is “Command and Control”. Development teams do not work like self organized team they have to work under the commands given by project director/ manager.

### 1.1.2 Disadvantages of Waterfall Model

Waterfall model provides a sequential and slow & steady flow of work. Track records of this model shows that this development team fails to meet the deadlines of time. Budget estimation is also not good. The main focus of team is on documentation completeness rather than development which cause delay in timelines to complete project. After decision phase customer is not involved but the presence of customer in later phases with developer is also necessary. This model does not support this facility. Code structure usually not refined. The “Command and Control” environment does not provide the chance for self development to developers. Often they don't know that why they are doing something. All these problems were covered by extreme programming development model. Here I will discuss all aspects, merit and demerits of extreme programming model.

## 1.2 Extreme Programming

As Waterfall model is inflexible in order to change requirements so extreme programming (XP) provides the features which cover the demerits of waterfall model. The XP methodology is important part of agile method. The agile method treats the unbalanced and fickle requirements well with the use of a number of methods e.g. less documentation, small iterations, early testing and customer partnership [1]. Some key factors for the success of projects under extreme programming are pair programming, high level of code review, unit testing, avoiding programming of features until they are actually needed, iterations, time boxing, clear and simple code, refactoring, flexibility to changes requirements and collaboration of client during development process. These features also cover the weak points of waterfall model. [4]

### 1.2.1 Practices of Extreme Programming

Extreme programming has been implemented with its 12 practices which are called its life cycle. Actually these practices are lifeblood of extreme programming. [8]

1. Planning
2. Small Releases
3. Metaphor
4. Design
5. Testing
6. Refactoring
7. Pair Programming
8. Collective Ownership

9. Continuous Integration
10. 40- hours week
11. On site Customer
12. Coding Standard

Later on the the testing was divided into two further categories. [5]

## 2. Problem and Its Proposed Solution

The extreme programming is now a trendy method to develop quality software than traditional system development methods but still there are some dark sides in this methodology but our focused problem is its delay in the completion of iteration during its time box which causes delay of software delivery. Here I will see some reasons and then propose its solution to overcome this delay.

One of the main reasons for this problem is the elimination of command and control environment from development phase as it is pair programming and self organized team. As in this development method there are two persons for development one is coder and second is reviewer. These persons can swap their duties after short span of time. [6]

But the problem arises when one is serious and dedicated for his duty and second is not. There can be many causes against this problem like their personal conflicts etc but there can be multiple problems for example iteration's time boxing expires before iteration completion because of careless behavior of developers, quality of code and software can be affected and it also can degrade moral and behavior of development team which can be resulted in the loss of organization from different aspects.

Here in this research paper our main focus is to reduce the number of delayed iterations and to complete them in time box limit. For this purpose I have presented a solution to supervise the paired development team with command and control environment but with less restriction so that it cannot affect the self learning ability of developers.

Another reason for the delay is that the extreme programming is code centric rather than design centric. It does not give any significant importance to design but if we design it properly before coding then it can help to complete iterations, mostly within time boxing.

## 3. Conclusion

This paper proposed a solution to overcome the time delay in the delivery of software specially the delayed iterations while time box for such iteration over. The peoples of the world are more

dependent on software so timely delivery of software is necessary. Time to market for software is now an important factor so the late development can cause disastrous for organization. So it is the basic need to improve software development methodology to strictly meet the timelines.

#### 4. References

- [1] Ji, F., & Sedano, T. (2011, May). Comparing extreme programming and Waterfall project results. In Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on (pp. 482-486). IEEE.
- [2] Rapley, K. (1995, December). RAD or TRAD or both? The future of software development. In Will Tickit and ISO 9000 Survive Rapid Application Development? IEE Colloquium on (pp. 3-1). IET.
- [3] Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software Development Life Cycle AGILE vs. Traditional Approaches. International Proceedings of Computer Science & Information Technology, 37.
- [4] Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. Z. (2010, June). Agile software development: Impact on productivity and quality. In Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on (pp. 287-291). IEEE.
- [5] Vanderburg, G. (2005, October). A simple model of agile software processes--or--extreme programming annealed. In ACM SIGPLAN Notices (Vol. 40, No. 10, pp. 539- 545). ACM.
- [6] Wray, S. (2010). How pair programming really works. IEEE software, 27(1), 50-55.
- [7] Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999). The unified software development process (Vol. 1). Reading: Addison-Wesley.
- [8] Newkirk, J. (2002, May). Introduction to agile processes and extreme programming. In Proceedings of the 24th international conference on Software engineering (pp. 695-696). ACM