Analysis of Linear Sequential and Extreme Programming Development Methodology for a Gaming Application

Pooja Sharma, Nitasha Hasteer

***Abstract*—There exist various software development methodologies under agile software development method. Extreme programming (XP) is known as one of the agile development methods that has gained popularity in the recent past. Waterfall model which is termed as linear sequential development model has been the traditional model of development. Both extreme programming and waterfall focus on different aspects during the process of software development and management. In this work we have analyzed the linear sequential software development methodology and the extreme programming method with regard to time constraints and risk mitigation. Our study shows that though there is no single methodology suited in all scenarios of development yet extreme programming gives fast delivery and less exposure to risks.**

***Index Terms*—Waterfall model, Extreme Programming (XP) model, Time constraint, Risk, Risk mitigation.**

I. INTRODUCTION

With the rapid boost in the economical development, the development of software is getting increasingly complex. There is no one single well defined and implemented development methodology that is suitable in all scenarios of the development but there has been the effective transition from traditional development to agile development [1-4]. Extreme programming is a type of agile software development methodology. This is capable of overcoming the limitations of the traditional software development approach in terms of implementation, faster delivery (small releases), easy change adaptability even late in the development cycle and increased customer satisfaction. Some of the XP practices are that XP focuses on early testing and automated testing. Testing in XP starts very early in the development phases. XP keeps short development cycles with incremental design and planning. It keeps the continuous involvement of customer in each phase of the development. It also involves continuous integration and daily deployment. Our study focuses around the analysis of traditional and agile practices of the software development. We have compared Linear sequential commonly termed as Waterfall model with the Extreme programming model of agile.

The rest of the paper is organized as follow: Section II defines the classic Waterfall model Vs. XP model. Section III elucidates the review of existing literature. Section IV illustrates our application and methodology used. We present our result in Section V and a conclusion is given in Section VI.

II. CLASSIC WATERFALL MODEL Vs. XP MODEL

Waterfall is the classic software development method [2]. It is the oldest software development strategy that is being traditionally used for software development and management comprising of five phases. It is linear sequential in nature due to which the output of one phase will be taken as the input of the next phase. This method also restricts to backtrack to the previous phase which makes it difficult to make corrections. This method conducts the intensive planning and documentation in the starting and shows its inflexibility in the face of changing requirements. There are five phases in the waterfall model. First phase of the waterfall model is the requirement gathering and specification in which software requirements are collected, analyzed and software requirement specification (SRS) document is prepared. Second phase is design phase where software architecture and rough design of the software is presented for implementation. Third phase is the implementation phase in which the software design is implemented into actually working software and individual modules are tested. Fourth phase is the testing phase where interface of the modules with each other and system as a whole is tested and debugged. Fifth phase is the operation and the maintenance phase in which the working software is deployed into real system and is maintained.

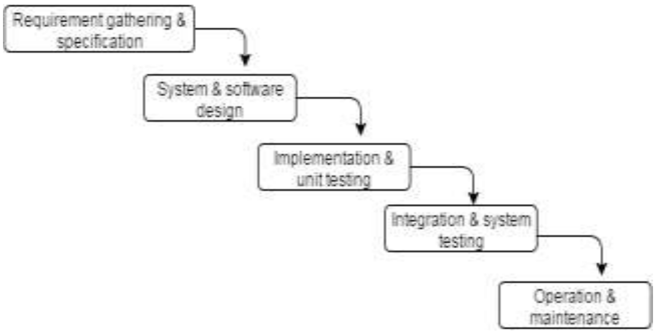
Requirement gathering & specification
System & software design
Implementation & unit testing
Integration & system testing
Operation & maintenance

Fig. 1. Five phases of Waterfall Model [31]

Due to the linearity and risk propagation of waterfall model the XP came in to use which is iterative in nature and propose the software project in small releases. This makes the XP model more popular than waterfall model.

Extreme programming is all about the social change in the field of software development [1]. It is a light weighted methodology for any size of software. XP is a software methodology that focuses on the pair programming, doing extensive code review, simplicity and clarity in code, flat management structure, keeps frequent track of communication with customer and programmer. It intends to improve software quality and responsive to changing customer requirement even late in the development cycle.

It improves quality, productivity and introduces check points at which new customer requirements can be adopted. It is incremental in nature based on continuous customer feedback. It has ability to flexibly accommodate the implementation of

Pooja Sharma is with the IT Department, Amity University, Uttar Pradesh, India (e-mail: poosharma7448@gmail.com)

Nitasha Hasteer is with the IT Department, Amity University, Uttar Pradesh, India (e-mail: nitasha78@gmail.com)

978-1-5090-0396-9/16/$31.00 ©2016 IEEE

functionality even adaptive to changing business requirements. XP works on the 'mentality of sufficiency' that is you may or may not have enough money, time, resources or skills in your team but it's always best to act as if there is going to be enough XP explains that you can do best even when there are constraints.

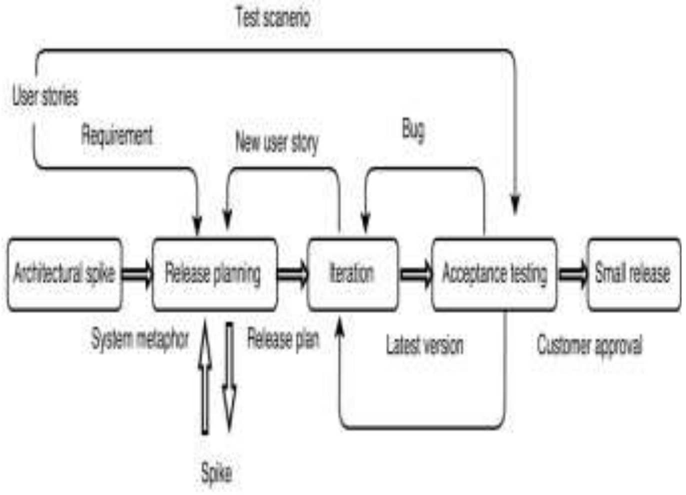

Fig. 2. XP Development Cycle [1]

Extreme Programming also encourages a high degree of interactions between team members, including managers, developers, testers, stakeholders and others. It also provides faster development and frequent releases. Extreme programming is a software development discipline that addresses the risks at all the level of the development process. It enables us to deliver on time and on budget project by small releases in many iterations that indicate the flexibility and risk minimizing capabilities of the process. XP adapts the changes in the project even late in development phase which may delay the completion time of the project and may increase the complexity of the project. Though it has many benefits yet according to the customer's changing need (volatility), completeness and the complexity of the project is at suffering. Because of accommodation of new arising needs of the customer, it is possible that project may not be completed at time, and there may be increased complexity in the project. Due to the above mentioned reasons there are some risks which are likely to occur. Table I explains the likely risks.

TABLE I
SCORE VALUE FOR THREE TYPE OF RISKS [11-32]

Risk type	Description	Score		
		0	1	2
Completeness	How complete do we know the details?	Complete	Incomplete	Unknown
Volatility	How likely are the changes?	Low	Medium	High
Complexity	How hard is it to build?	Simple	Standard	Complex

III. REVIEW OF LITERATURE

In this area relevant case studies has been conducted for the comparison of the classic waterfall model with the XP methodology [2-11]. Table II. Below shows some of the earlier works reporting the comparison of Waterfall with XP.

TABLE II

EXISTING WORK COMPARING WATERFALL AND XP

Year	Author	Work done
2010	Nabil Mohammed Ali Munassar et.al[2]	This paper has done a comparative of five different software development models for different sizes of the model, talked about their advantages and disadvantages.
2011	Jennifer Dorette Jacob [10]	This thesis focused on investigating a suitable software development process to be followed for start-up companies by comparing agile(XP) and waterfall.
2011	Feng ji et.al [11]	This paper at silicon valley have done comparison among waterfall and XP and shows the effective transition from traditional development to agile development

In the above given case studies though the comparison between traditional waterfall and XP has been done which defines the effective transition from linear sequential to XP of agile but there was no on ground real time implementation which actually demonstrates the comparison between traditional waterfall and XP of agile specifically on the basis of time constraint and the risk mitigation.

There are number of studies which gives the implementation methodology of the XP. Some of them from the digital library IEEE are given in Table III.

TABLE III
RELATED STUDIES

Year	Author	Work done
2015	Kholid Haryono [3]	This paper used XP for financial management in public sector and found that low level user is highly resistance to change, that demanded some better strategy to cope up with such problem.
2014	Rajendran Swamidurai et.al [4]	This paper done comparison among two light-weight and low cost collaborative inspection methods, traditional pair programming and peer code review, that shows effective improvement in the quality of software.
2014	Takaaki Goto et.al [5]	This paper focuses on the XP method for Innovative Software based on systems Design
2014	G. Sivanageswara Rao et.al [6]	This paper shows that how software development limits the possibilities of adapting the development process to the changes and update requests. This problem might be solved by adoption of Extreme Programming (XP).

2012	Linghui Liu et.al [7]	This paper talks about the traditional method lacking in adapting requirement variety, discuss the advantages and disadvantages of XP.
2012	Putu Edy Suardiyana Putra et.al[8]	This paper focuses on the industrial practices of XP for improved software quality.
2011	Zhai Li-li et.al [9]	Aims at the core values of XP and the barriers to the demand communication Process, study of kano model for quality characterstics of software.
2011	Xin Dong, Qiu-Song Yang, Qing Wang, Jian Zhai	This paper focuses on selecting the right user stories and planning their implementation for the next iteration for success of (XP) and risk analysis.

The above given table (Table III) gives the better understanding of the XP and detailed idea that how XP is implemented for software development. Study gives the Application of XP in various sectors with innovative designs and knowledge of industrial practices of XP.

IV. METHODOLOGY

For our comparison we took an existing project of developing a gaming application to be developed using the waterfall methodology in C^{++} and another similar application to be developed using the extreme programming approach which used java. Considering that existing application in waterfall is taken as P_1 and similar application developed in XP as P_2.

TABLE IV
PROJECT STATISTICS

Method Parameter	Waterfall model (P_1)	XP model (P_2)
Language	C^{++}	Java
No. of team members	6	4
Time (In days)	90	75(First release)

While working on to the waterfall model for the application development two of the team members were into requirement gathering and the documentation which took 25% time of the total development process as traditional waterfall requires high documentation. Among all the members of the team one was associated as the designer consuming around 16%, two team members were associated with the implementation (30%) of over-all time and one was taking the account of the testing with 12% time consumption and maintenance took 17% time of the process.

On another hand we undertook a similar project of gaming application to be developed with XP development methodology where the XP team was having was having 4 members, where one team members was into requirement gathering (user stories) and a simple documentation which consumed only 14% of the complete process as XP demands for very less documentation and kept a regular communication which helped the team in updating changing needs, two of the team members were in release plan with 16% of the time duration and carried out the iteration process with 20 % time duration and one member worked as a tester with 25% time and for small release it took 20% process time of the development process.

V. RESULT ANALYSIS

While working with both the methodologies result is analyzed on time variation among different phases of the development and risk factors are identified with their impact.

A. TIME CONSTRAINT

Among the time distribution to the various activities Fig. 3 & Fig. 4 shows the time taken by the both Waterfall and XP methodology respectively. It is found that the XP methodology dedicates its maximum time to the testing phase.

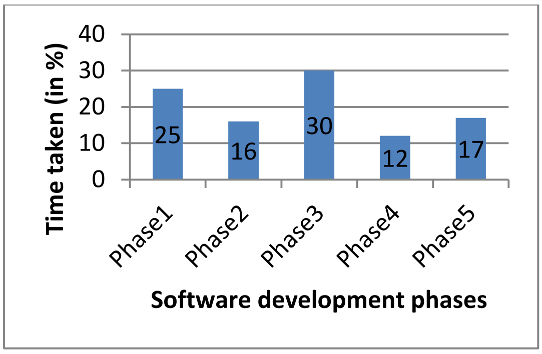

Fig. 3. Time taken in execution of Project P_1

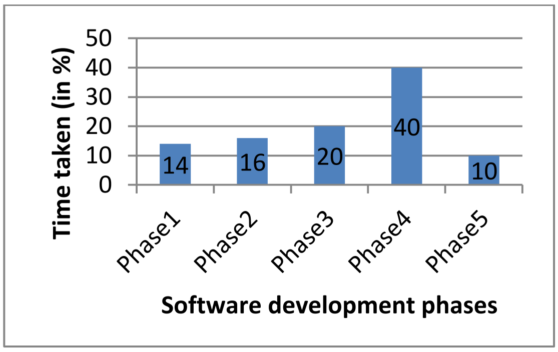

Fig. 4. Time taken in execution of Project P_2

The given Table IV shows the building of the same software application where the study shows that the Project P_2 takes less time and there was 16.66% reduction in time.

B. RISK MITIGATION

XP follows an automated testing strategy and restricts the propagation of the risk in the later stages. It keep on checking the risk occurrence after each phase of the development and resolve it at that very stage of development cycle.

During the work process under the both traditional waterfall and XP it is found that waterfall team did not keep the communication process on throughout the development process due to which requirement once misunderstood finally tends to develop a faulty system. Waterfall focuses to find the bugs in the later stage only due to which risk propagation in next phases of the development cycle increases that finally ends up costing more than 50 times the amount to fix it.

For risk mitigation we calculated risk exposure for both waterfall and XP teams for which we identified the risk with their probability of occurrence and their impact on a scale of 1-5. Risk identified were Requirement being misunderstood, Staff turnover, Faulty design and Bug propagation which are given in Table V as R_1, R_2, R_3, R_4 respectively. For the below given table probability of occurrence of identified risks are on the basis of the survey administered to the team members and the expert opinion has been incorporated for the same.

TABLE V
RISK TYPE WITH THEIR OCCURRENCE PROBABILITY AND IMPACT

Risks identified	Probability of occurrence in P_1	Impact of risk	Probability of occurrence in P_2	Impact of risk
R_1	4	5	3	2
R_2	3	4	3	4
R_3	4	4	2	3
R_4	5	4	2	2

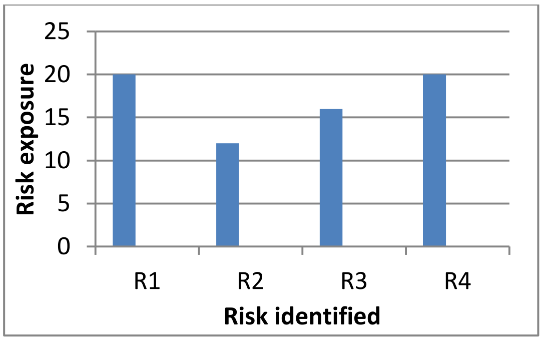

Fig. 5. Risk exposure in Project P_1

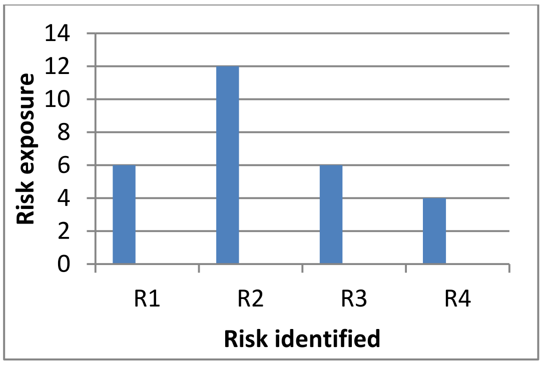

Fig. 6. Risk exposure in Project P_2

While calculating the risk exposure for both the projects it was analyzed the XP methodology is less prone to risks as it starts testing early and dedicates maximum time to testing in development cycle. But due to less documentation, staff turnover is a major risk in XP methodology because new staff may not have a formal documentation from where to take development forward and what to do.

VI. CONCLUSION

After the study was conducted it was concluded that among Classic Waterfall and XP of agile, there is no single methodology that can be suggested best suitable in all scenario of software development. Here XP methodology gave better result with fast delivery and reduced risk exposure in development of application P_2. It is found that XP better suited for the small to medium size software project and when it comes to large scale software project waterfall model can be taken as the base. Though Waterfall can be adopted as the base for the various software projects yet due to its linearity in nature Waterfall fails to give fast delivery with changing needs and high accuracy.

REFRENCES

[1] Kent beck, Extreme programming explained 2^{nd} Edition, Addison Wesley, November 2004

[2] Nabil Mohammed Ali Munassar and A. Govardhan, "A Comparison Between Five Models Of Software Engineering", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010

[3] Kholid Haryono, "The extreme programming approach for financial management system on local government", RMUTT, *International conference on science and technology*, pp 29-34. *IEEE*,2015

[4] Rajendran Swamidurai, "*Investigating the Impact of Peer Code Review and Pair Programming on Test-Driven Development*", pp 1-5. *IEEE* 2014

[5] Takaaki Goto, Kensei Tsuchida, Tetsuro Nishino, "EPISODE: An Extreme Programming Method for innovative Software Based on Systems Design" IIAI *3rd International Conference on Advanced Applied Informatics, IEEE* 2014

[6] G. Sivanageswara Rao, Ch. V. Phani Krishna, K. Rajasekhar Rao, *"Extreme Programming for Service-BasedApplication Development Architecture" IEEE* 2014

[7] Linghui Liu, Yao Lu, *"Application of Agile Method in the Enterprise Website Backstage Management System", IEEE* 2012

[8] Putu Edy Suardiyana Putra, Arlisa Yuliawati, Petrus Mursanto, "Industrial Extreme Programming Practice's Implementation in Rational Unified Process on Agile Development Theme, ICACSIS, *IEEE* 2012

[9] Zhai Li-li, Hong Lian-feng, Sun Qin-ying, *"Research on Requirement for High-quality Model of Extreme Programming", International Conference on Information Management, Innovation Management and Industrial Engineering, IEEE* 2011

[10] Jennifer Dorette Jacob, " Comparing Agile XP and Waterfall Software Development Processes in two Start-up Companies", Master of Science Thesis in the Programme Software Engineering and Technology, Göteborg, Sweden, November 2011

[11] Feng Ji, Todd Sedano, "Comparing Extreme Programming and Waterfall Project Results" ,Carneige mellon university, Conference software engineering education and training, 2011

[12] Francisco Macias, Mike Holcombe, Marian Gheorghe "A Formal Experiment Comparing Extreme Programmingwith Traditional Software Construction",Proceedings of the Fourth Mexican *International Conference on Computer Science*, pp73-80, *IEEE*,2003

[13] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad,Wolfgang Slany, Martin Umgeher, Thomas Vlk "Optimizing Extreme Programming", *Proceedings of the International Conference on Computer and Communication Engineering, IEEE*, 2008

[14] Markus Hummel "State-of-the-Art: A Systematic Literature Review on Agile InformationSystems Development", *47th Hawaii International Conference on System Science, IEEE*, 2014

[15] Jim Q. Chen, Dien Phan, B. Wang, Douglas R. Vogel *"Light-Weight Development Method: a Case Study" IEEE*,2007

[16] Samireh Jalali, Claes Wohlin "Agile Practices in Global Software Engineering – A Systematic Map" *International Conference on Global Software Engineering, IEEE*,2010

[17] Kai Stapel,Daniel Lübke, Eric Knauss, "Best Practices in eXtreme Programming Course Design**"** ICSE,ACM ,2008

[18] Sara Shahzad, "Learning From Experience: The Analysis of an Extreme Programming Process", *Sixth International Conference on Information Technology: New Generations, IEEE*, 2009

[19] Yael Dubinsky,Orit Hazzan, "eXtreme Programming as a Framework for Student-Project Coaching inComputer Science Capstone Courses", *Proceedings of the International Conference on Software—Science, Technology & Engineering, IEEE*, 2005

[20] Scott E. Carpenter, Aldo Dagnino, "Is Agile too Fragile for Space-Based Systems Engineering?", *International Conference on Space Mission Challenges for Information Technology,IEEE*, 2014

[21] Jitender Choudhari, Dr. Ugrasen Suman, "Extended Iterative Maintenance Life Cycle Using eXtreme Programming", SIGSOFT Software Engineering Notes, ACM, 2014

[22] D. Woit, K. Bell, "Do XP Customer-Developer Interactions Impact Motivation? Findings from an Industrial Case Study", CHASE, ACM, 2014

[23] Zhai Li-li, Hong Lian-feng, Sun Qin-ying, "Research on Requirement for High-quality Model of Extreme Programming", *International Conference*

on Information Management, Innovation Management and Industrial Engineering, IEEE, 2011

[24] Anupriya Tuli, Nitasha Hashteer, Megha Sharma, Abhay Bansal, “Emprical investigation of agile software development: Cloud perspective”, SIGSOFT Software Engineering Notes, ACM,2014

[25] Jukka Kääriäinen, Juha Koskela, Pekka Abrahamsson, Juha Takalo, “Improving Requirements Management in Extreme Programming with Tool Support – an Improvement Attempt that Failed”, *Proceedings of the 30th EUROMICRO Conference,IEEE*, 2004

[26] Xu Bin, Yang Xiaohu, He Zhijun*, “Extreme Programming In Reducing The Rework Of Requirement Change”, CCECE, IEEE*, 2004

[27] D.M. Woit, “Requirements Interaction Management in an eXtreme Programming Environment: A Case Study*”, ICSE, ACM,2005

[28] Darshan Domah / Frank J. Mitropoulos, “*The NERV Methodology: A Lightweight Process for Addressing Non-functional Requirements in Agile Software Development”,SECON, IEEE,* 2015

[29] Danh Nguyen-Cong, De Tran-Cao, “A Review of Effort Estimation Studies in Agile,Iterative and Incremental Software Development”, *International Conference on Computing & Communication Technologies -Research, Innovation, and Vision for the Future,IEEE,* 2013

[30] Amani Mahdi Mohammed Hamed, Hisham Abushama, “Popular Agile Approaches in Software Development: Review and Analysis”, *International Conference On Computing, Electrical And Electronic Engineering, IEEE*,2013

[31] Roger S Pressman, Software engineering-A practitioner’s approach 6th edition, TATA McGRAW-HILL

[32] Xin Dong, Qiu-Song Yang, Qing Wang, Jian Zhai, Günther Ruhe, “Value-Risk Trade-off Analysis for Iteration Planning in eXtreme Programming”, *18th Asia-Pacific Software Engineering Conference, IEEE*,2011