



## Process of Moving from Traditional to Agile software Development: A Review

**ShubhinderKaur**

Research Scholar (ME-CSE)  
Chandigarh University,  
Gharuan, India

**Er. Amanpreet Kaur Boparai**

Asst. Prof., Department of Computer Science  
Chandigarh University,  
Gharuan, India

---

**Abstract:** *Waterfall model and Extreme programming are the two software project development methods used for project management. The traditional project development is often referred to as waterfall model and extreme programming is one of the methodologies of agile software development. Many companies are moving to agile software development to improve quality and reduce delivery time. The notion of the paper is to present waterfall model and XP and make a comparison between them to show their features and defects.*

**Keywords:** *Traditional software development, Agile software development, Extreme programming.*

---

### I. INTRODUCTION

SDLC is a framework defining tasks which are performed at each step in developing a software. SDLC defines a methodology for improving the quality of software and the overall development process. It is immensely important to decide on the accurate model in software development. A software development approach generally guides the developer through the software development process. A SDLC selection criteria is crucial as it ensures the organization to maximize the change to deliver software successfully. The first model that popped up was the waterfall model. The other models that kept appearing were prototyping model, incremental model, spiral model, RAD model and finally in 2001 Agile software development was formulated although it was introduced in 1974 by Edmonds in his research paper[7]. Some agile methodologies are Rational Unified Process(RUP), Scrum, Crystal Clear, Extreme Programming(XP), Adaptive Software Development, Feature Driven Development(FDD), Dynamic System Development Method(DSDM). Agile software development is a group of software development methodologies based on iterative and incremental development where requirements and solutions evolve through collaboration between self-organizing, cross functional teams[4]. Agile methodology has an adaptive team which responds to changing requirements and the most important principle is customer satisfaction which is maintained through rapid and continuous delivery of product. It is very useful methodology to be adopted in modern software development process to replace the traditional heavy weight development life cycle[5].

### II. OVERVIEW

#### **A. Traditional software development:**

Software methodologies like waterfall model, V-model, spiral model and prototyping models are called traditional software development methodologies. These methodologies follow a sequential series of steps—Requirements, Design, Implementation, Testing, Deployment and Maintenance. Documenting a stable set of requirements is needed at the beginning of a project. In other words, we can say that traditional software development methods are dependent on set of predetermined processes and on-going documentation.

#### **Waterfall model:**

Waterfall model is a sequential development model where requirement should be clear before going to next phase of development process. Documentation and testing happens at the end of each phase which maintains the quality of a project. Testing is undertaken once the code has been successfully developed. For decades, waterfall project managers have presented stake holders with detailed plans for the entire development process along with documented commitment and regular status reports on how well the project is meeting key milestones. In addition to being familiar it give stakeholders a sense of comfort and control—although if initial assumptions are incorrect or requirements change much time, effort and money can be wasted.

Table 1: Features of waterfall model

	ADVANTAGES	DISADVANTAGES
WATERFALL L MODEL	<ul style="list-style-type: none"> <li>• Simple to implement</li> <li>• Documentation produced at every stage</li> <li>• Requirements are clearly stated before development</li> <li>• Milestones are known and tracked</li> </ul>	<ul style="list-style-type: none"> <li>• Rigid and inflexible</li> <li>• Massive documentation</li> <li>• Less focus on customer values</li> </ul>

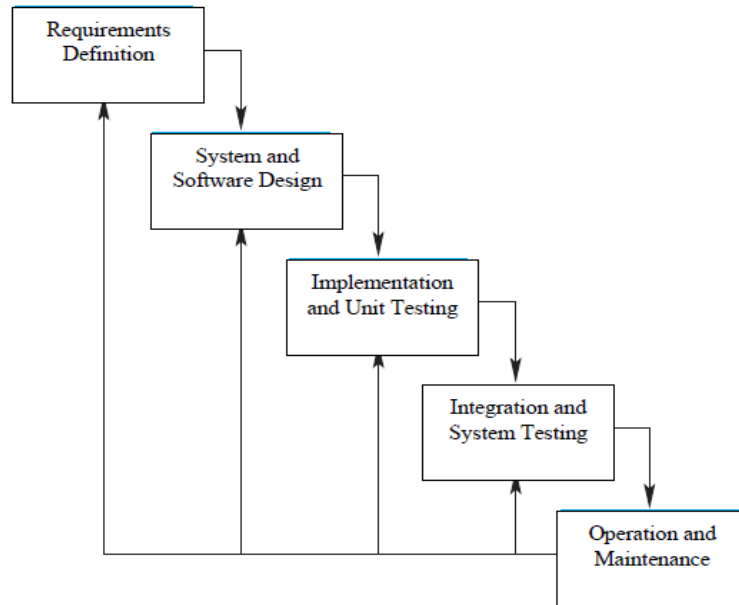


Figure1. Waterfall model[12]

**B. Agile Software Development**

Agility is the capability of software to choose and react expeditiously and fittingly to various changes in its surround and to the demands imposed by surround. An agile process is the one that readily embraces and supports high degree of flexibility. Agile software development is based on the idea of incremental and iterative development in which phase within development cycle are revisited over and over again The agile manifesto was written in 2001 which is based on:

- Individuals and interactions over process and tools
- Working software over comprehensive document
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The major factors according to agile manifesto on which the development is based are:

- Early customer involvement
- Iterative development
- Self-organizing teams
- Adaptation to change

Agile methodology comprises of an adaptive team which is able to respond to changing requirements. There is no guess work between the development team and the customer. The pitfall is that it become difficult to judge the efforts and time required in large projects.

Some discriminative features of agile process are modularity, iterative, time-bound, adaptive, convergent, incremental, people oriented and collaboration. It’s been found that XP is the main forum of research in most of the countries.

**Extreme Programming (XP):**

Extreme programming, a light weight development process, is becoming increasingly influential for rapid development of software applications. XP, created by Kent Beck, argues for real customer involvement and quickly moves towards working code. In XP the development process is split into iterations where the beginning of each iteration includes planning and user stories. After that acceptance tests are done for each user story. The core values for XP are communication, simplicity, feedback, courage and respect. XP consists of 12 practices and is suitable for small to medium sized teams.

Table2: Features of XP

	Advantages	Disadvantages
EXTREME PROGRAMMING	<ul style="list-style-type: none"> <li>• Quick delivery and feedback</li> <li>• Continuous reviewing and testing</li> <li>• Customer satisfaction</li> <li>• Flexible</li> <li>• High level of communication</li> </ul>	<ul style="list-style-type: none"> <li>• No focus on long term goals</li> <li>• Having full-time on-site customer is impractical</li> <li>• scalability</li> </ul>

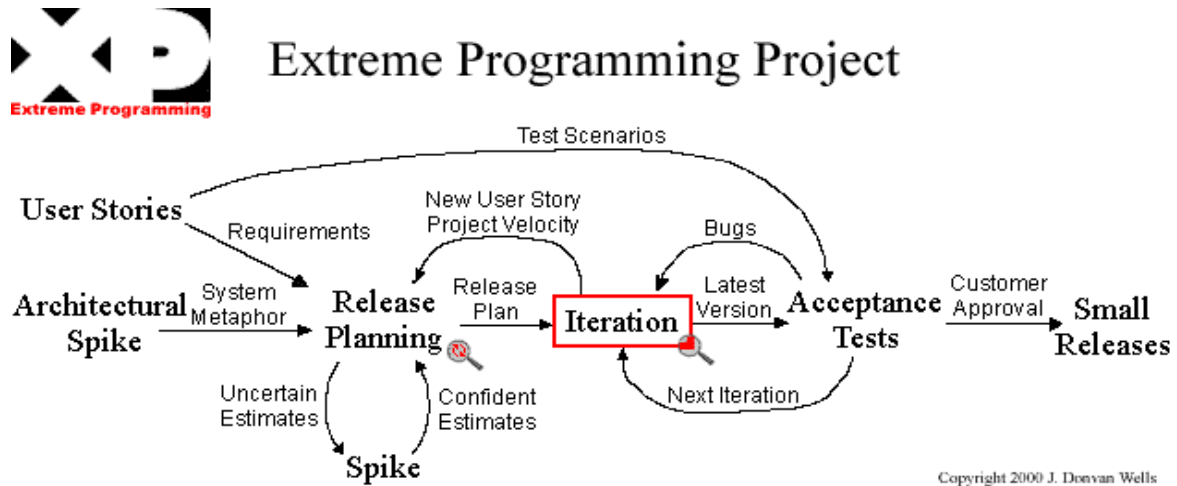


Figure2: Extreme Programming[1]

**XP Practices[8], [11]:**

To increase the productivity while maintaining quality XP proposes set of software development practices.

1. Customer Satisfaction: The main issue is to deal with customer problems. For customer satisfaction XP uses two practices: on-site customer and small releases.
2. Software quality: XP employs various practices to keep software quality high. These practices ensure that the team maintains high quality without slowing down the development process.
3. Metaphor: Metaphor represents unified view of the system that makes sense to both business and technical sides as well as represents what work is being done.
4. Testing: Regression testing is key part of XP. Before the actual code is written programmers write unit tests. Testing includes unit testing, functional testing and acceptance testing.
5. Simple design: As the requirements keep changing from the customers side it is advised to keep the software design simple. It improves team ability to work productively with minimal documentation.
6. Refactoring: The structure of the code is changed to improve its understand ability and maintainability without changing its functionality. Constant refactoring ensures that the design is always as simple as possible.
7. Pair programming: The code is written by two people working at one machine. One person focuses on whether the approach will work and whether it can be simplified. The other person thinks strategically if it is the best way to implement the functionality.
8. Project management: XP includes practices that reduce management overhead to ease the process of project management. XP uses planning game to chart the next releases scope, priority, release date and all technical findings—estimates and consequences.
9. Sustainable development: No one can work consequently for long without affecting product quality.
10. Collective ownership: All the team members own the code. Anyone who can add value to any portion of the code at any time is required to do so.
11. Coding standards: Coding standard makes the code easily understandable and improves consistency among team members. Different team members program different parts of system so coding standards are must.
12. Continuous integration: Developers should integrate code as often as possible. This ensures that there is always an executable version available which can serve as baseline.

**III. RELATED WORK**

For our study, we reviewed several papers on traditional approaches, agile process and extreme programming.

Malik Hneif and SiewHockow[1] explains the evolvement of software development towards agile methodologies, explains various agile methodologies i.e XP and scrum, elaborates the advantages and disadvantages of using agile methodologies. Advantages :satisfies customer, accept changes in requirements and high degree of communication. Disadvantages: not suitable for maintenance, depend heavily on user involvement.

KaushalPathak and AnjuSaha[2] reviewed different agile methodologies and explained how these are diverse from the traditional one as agile approaches are emergent, iterative, flexible and adaptive. There is high level of communication and interaction and the focus is on meeting the customer requirements in each iteration. The paper suggests to review agile practices more in research.

TorgeirDingsoyr, Sridhar Nerur, VenugopalBalijepally and Nils Brede Moe[3] delineate the contribution of authors in study of agile methodologies. They summarized research and introduced contribution of agile software development. Their study suggested that there is a need for more study in the field of agile software development so that it can mature. The paper presents the research on agile software development which included literature survey and it concluded that the articulation of the agile manifesto in 2001 has brought unprecedented changes to the software engineering field. Indeed, the transformation that the manifesto has brought is quite remarkable.

S. Balaji and Dr. M. SundrarajanMurugauyan[4] compared waterfall model, v-model and agile model. The comparative analysis summarizes various steps that an organization should take to make best possible choice for selecting the appropriate model for software development. The paper suggests that if the requirements changes frequently and the project is small then choose agile model, if the requirements are clear in large project then use waterfall model and if requirement changes in larger project and tester is involved in early phases then use V-model.

Yu BengLeau, WooiKhong Loo, Wai Yip Tham and Soo Fun Tan[5] differentiated between agile and traditional approaches. The traditional approaches are denoted as heavy weight methodologies and agile methods are called as light weight methods. The paper differentiates agile and traditional approaches on the following basis- user requirements, rework cost, development direction, testing, customer involvement, extra quality and project scale.

PreetiRai and SaruDhir [6] describes the impact and comparison of various traditional methodologies and new methodologies in their paper. The paper introduces various traditional methods for software development- waterfall, prototyping model, incremental model ,spiral model and finally discusses agile methodology. The process of agile software development consists of six phases: pre-project preparation, start, construction, production, selection and departure.

Aline U. Hakizabera and Koichi Yamada [7] discussed the various software development models- linear models and agile methodologies. The paper discusses a decision criteria to choose the right SDLC model using Dempster- Shaffer theory. The decision process involves a series of information collection, judgement and evaluation and further decision is made according to available data and goals of software project. Dempster- Shaffer theory helps in decision making in situation of uncertainty and incomplete data.

GhazyAssassa, Hassan Mathkour and Hmood Al Dossari [8] explored twelveXP practices in context of Software engineering education. XP is considered as most famous and prominent agile methodology. Some new strategies discussed were- incremental change, small initial project investment, stand up meetings, tracking progress, minimal documentation, teach strategies and experiment. It concluded that XP team produced product with full functionality and less work.

Granville G. Miller [9] elaborated the characteristics of agile software process which makes it different from other processes-modularity, iterative, time bound, parsimony, adaptive, incremental, convergent, people oriented and collaborative. The paper concluded that agile is not a new process but have evolved from the best practices which have refined over the past 30 years.

Rupinder Singh and AmandeepBakshi [10] reviewed important aspects of traditional and agile method of software development and compares both methodologies. The paper discusses various flaws and challenges faced by traditional processes. The traditional methods are presumptuous, they assume many aspects before taking up the work like requirements are stable, etc whereas agile methods include various factors, among them are- early customer involvement, iterative development, self-organizing teams and adaptation to change. Paper concluded that agile have an edge upon traditional methods as it is flexible and promotes adaptive planning.

#### IV. COMPARISON OF AGILE AND TRADITIONAL SOFTWARE DEVELOPMENT

Different studies and surveys have revealed that agile methodologies are being used excessively and a lot of research is going on in this area. The difference between agile development and traditional development methods are that agile methods emphasize on teams, working software, customer collaboration and responding to changes while traditional methods stress on contracts, plans, processes, documents and tools[5].

Although agile methodologies triumph traditional methodologies in many aspects, there exist several difficulties in putting it into practice. Agile methods claim that code could itself act as a document but it becomes difficult for novice developers or new team members to complete task when they could not comprehend the project. Customer involvement for such a long time or full- time is impractical [11]. Another issue is the scalability of the process. But to meet the changing requirements of the software industry and customers agile methodologies are best suited.

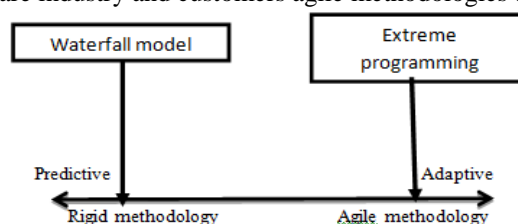


Figure3: Comparison between waterfall model and XP

Table 3[2] [5] [6] [10] shows the comparison between agile process and traditional process. Table 4[1] [4] [7] [8] [11] shows the comparison between waterfall model and extreme programming.

Table 3:

Features	Traditional Software Development	Agile Software Development
Requirements	Stable	Rapidly changes, flexible
Documentation	Heavy documentation	Light documentation
Customer-involvement	Low (only during initial phases)	High
Refactoring	Expensive	Inexpensive
Testing	Done after coding	Done at each iteration
Suitable project scale	Large scale	Low to medium sized

Table 4:

Features	Waterfall model	Extreme programming
Requirements	At the beginning	Frequently available
Changing requirements	Not supported	Supported
Success guarantee	Low	Good
Budget	Generally high	Low
Simplicity	Simple	Simple
Timeframe	Long term	Short term
Approaches	Traditional	Object oriented
User involvement	Only at the beginning	High involvement
Flexibility	Rigid	Flexible
Functionality	Static	Dynamic
Documentation	High documentation	Low (Only when needed)
Reusability	Limited	High

## V. CONCLUSION

Software development methodologies have evolved since 1970's. Presently the process of software development is undergoing a evolutionary shift, from traditional software methodologies like waterfall model towards agile methodologies like XP. Agile development methodologies deliver high quality software products in rapid iterations with high flexibility and adaptability to changing conditions as it becomes quite difficult to plan the from start to end using traditional plan driven approach. From this review we can conclude that Agile methods excel traditional processes but still it is important that the development team selects a process that best suites the project.

If the project size is small, requirements changes frequently and the product is to be delivered in short period of time with skilled resources then agile model(XP) is apt.

If the requirements are clear and project is large then waterfall model is appropriate.

## REFERENCES

- [1] Malik Hneif, Siew Hockow (2009, October). Review of agile methodologies in software development. *IJRRAS*. [Online]. *Volume1(issue 1)*, pp 1-8.
- [2] KaushalPathak, AnjuSaha(2013,February) Review of agile software development methodologies. *IJARCSSE*, [Online], Volume 3(Issue2), pp 270-276. Available: [www.ijarcsse.com](http://www.ijarcsse.com)
- [3] TorgeirDingsoyr, Sridhar Nerur, VenugopalBalijepally and Nils Brede Moe(2012,March) A decade of agile methodologies: Towards explaining agile software development. *The journal of systems and software*
- [4] S. Balaji, Dr. M. SundrarajanMurugaiyan(2012, June) Waterfall vs V model vs Agile: Acomparitive study on SDLC. *JITBM* [Online], Volume 2(issue 1), pp 26-30. Available: [www.jitbm.com](http://www.jitbm.com)
- [5] Yu BengLeau, WooiKhong Loo, Wai Yip Tham and Soo Fun Tan(2012), Software development lifecycle vs Traditional Approaches. *IPCSIT* [Online], Volume 37, pp 162-167

- [6] Preeti Rai, Sarudhir (2014). Impact of different methodologies in Software development process. IJCSIT,[online], volume 5(issue 2), pp 1112-1116. Available: [www.ijcsit.com](http://www.ijcsit.com)
- [7] Aline U. Hakizabera and Koichi Yamada(2010,September). Linear model vs agile models: Making the right model decision. IEEJ[online], Volume 10(issue 1), pp 85-88.
- [8] Ghazy Assassa, Hassan Mathkour and Hmood Al Dossari, Extreme programming: a case study in software engineering courses, unpublished
- [9] Granville G. Miller, "The characteristics of agile software process," in TOOLS'01, 2001
- [10] Rupindersingh, Amandeep Bakshi(2013, March). Need of agile development. IJRTE[Online], volume 2(issue 1), pp 59-61
- [11] Frank Maurer, Sebastien Martel(2002, January). Extreme programming
- [12] 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