Empirical Study of Agile Software Development Methodologies: A Comparative Analysis

Gurpreet Singh Matharu
Amity University Uttar Pradesh
Noida, India
mtech.gurpreet@gmail.com

Anju Mishra
Amity University Uttar Pradesh
Noida, India
amishra1@amity.edu

Harmeet Singh
Software Engineer
Delhi, India
harmeet123@gmail.com

Priyanka Upadhyay
Amity University Uttar Pradesh
Noida, India
priyanka.upadhyay0991@gmail.com

ABSTRACT
In today's software industry, technological prowess and ever-evolving customer requirements have led to more complex software demands. Agile based software development is increasingly being adopted by the software practitioners as it assures early software development and high quality software products. Also, it offers responsiveness to changes in user requirements, providing for their quick absorption during software development. In this paper, we elaborate the significance, benefits and increasing adoption of Agile methodologies in context of today's complex and evolving software demands. The main objective of this paper is to conduct an empirical study into the choice among the most popular Agile methodologies, Scrum, Extreme Programming and Kanban. Further, this paper provides for a comparative analysis among various agile software development methodologies. Our survey results reveal higher adoption of Scrum based development in present-day software industry as compared to Extreme Programming and Kanban methodologies.

General Terms
Management, Documentation, Performance, Human Factors, Theory.

Keywords
Software Development Life Cycle (SDLC), Agile Methodologies, Scrum, Extreme Programming (XP), Kanban, Lean.

1. INTRODUCTION
In modern software industry, delivery of high-quality software is the prime objective for software developers. Non-planned and non-systematic approach to software development, if applied for complex software requirements, will certainly result in development of low-quality and high-cost software products. Thus, the approach to software development plays a significant role in deciding the quality of software being delivered. This realization among the software practitioners led to the formation and deployment of several software development life cycle models. According to Kevin Roebuck [1], a Software Development Life Cycle provides the framework for planning and controlling the development or modification of software products, along with the methodologies and models used for software development.

The software industry has shifted from traditional software development models to agile based development in response to ever-increasing software complexity and dynamic user requirements. Unlike the traditional models, agile methods are characterized by shorter development cycles, higher customer interaction, incremental delivery, frequent redesign with accommodation of changes necessitated by dynamic user requirements. Although diverse software development methodologies practice the same set of Agile principles formulated by Agile Manifesto [17], but they differ amongst each other on several parameters.

This study provides a comparison among the most popular Agile methodologies, Scrum, Extreme Programming and Kanban. The survey results would certainly assist the software practitioners in choosing among these Agile methodologies. The rest of this paper is structured as follows. Section 2 undertakes a literature review, followed by section 3, which elaborates on the transition towards agile based software development and its associated benefits. Section 4 discusses the most popular Agile methodologies, followed by Section 5 which provides a comparative analysis among Scrum, Extreme Programming and Kanban methodologies of Agile-based software development, ending up with Section 6 which provides the conclusions.

2. LITERATURE REVIEW
Although several studies have been conducted by individual teams, but little empirical data is available in support of success and higher adoption of agile software development methodologies. The Standish Group's 2011 CHAOS report [2] concludes that Agile methodologies are three times as successful as the traditional software development approaches.

The findings of the 8th Annual State of Agile Development Survey [3] conducted by Versionone.com in 2013 conclude that 92% of respondents believe that agile approach assists them in managing changing customer requirements; 87% of them agree that agile approach helps improve their team's productivity whereas 70% believe that agile software approach accelerates the software development process. Further, the survey indicates a clear trend towards the higher adoption of Scrum-based methodologies with 73% of respondents practicing Scrum and Scrum variants for software development.

The Xebia's Agile Survey 2012 report [4] reveals that 80% of the respondents agreed using Agile methodologies for software development. Further, the survey concludes that 92% of the respondents follow Scrum & Extreme Programming (XP) whereas 30% follow Kanban. The empirical study conducted by O. Salo and P. Abrahamsson [5] point towards higher adoption of Extreme Programming than Scrum methodology by embedded software development organizations across Europe.

Another significant empirical study conducted by Andrew Begel and Nachiappan Nagappan [6] at Microsoft reveals that about 33% of the respondents agreed using Agile software development methodologies, with Scrum being the most popular agile methodology at Microsoft. The survey conducted by G. Azizyan and M.K. Magarian [7] concludes that two-thirds of the respondents used Agile tools, with Scrum being the most widely used agile methodology among the respondent companies.

3. TRANSITION TO AGILE SOFTWARE DEVELOPMENT

Since 1970s, advanced technological interventions and dynamic user requirements have contributed towards more complex software demands which necessitate planned software development approach through development and application of formal software development models, thus eliminating old-fashioned and informal software development approaches. Modern software practitioners employ software development models for producing high-quality software, meeting user requirements and ensuring timely delivery in a cost effective manner. The software development models ensure systematic and organized approach to software development [8].

Heavyweight methodologies, also known as traditional software development approaches are characterized by comprehensive planning, process-orientation, predictive approach and heavy documentation. Unlike traditional software methodologies, lightweight methodologies promise frequent delivery of software increments in small iteration cycles and are team-oriented and adaptive approach. The lightweight methodologies, popularly known as agile methodologies, have made huge inroads into the software industry in the past few years. A comparison between traditional methods and agile methods is presented in Table 1.

Table 1. Comparison of Traditional Software Development Methods and Agile Software Development Methods

Parameter	Traditional Methods	Agile Methods
Adaptability to Change	Change Sustainability	Change Adaptability
Development Approach	Predictive	Adaptive
Development Orientation	Process-Oriented	People- Oriented
Project Size	Large	Small/Medium
Planning Scale	Long-term	Short-term
Management Style	Command-and-control	Leadership-and-collaboration
Learning	Continuous Learning while Development	Learning is secondary to Development
Documentation	High	Low

Agile based software development methodologies offer systematic software production resulting in enhanced quality of software products. Also, Agile based methods are characterized by improved productivity, flexibility, enhanced customer engagement and responsiveness to changes in user requirements. Several surveys have been confirmed rapid adoption of agile approaches in software industry. Such a survey was undertaken by Murphy et al. [9] that demonstrated a boost in practice of Agile methodologies within Microsoft between 2006 and 2012.

4. AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

In Agile approach to software development, work is carried out in small phases, based on collaboration, adaptive planning, early delivery, continuous improvement, regular customer feedback, frequent redesign resulting into development of software increments being delivered in successive iterations in response to the ever-changing customer requirements. Agile methodologies are increasingly being adopted by companies worldwide to meet increased software complexity and evolving user demands.

The Agile software development embodies several methodologies including Extreme Programming, Scrum, Kanban, Lean, FDD (Feature-Driven Development), Crystal, DSDM (Dynamic Systems Development Method).

Figure 1. Benefits of Agile Methods

4.1 Scrum

In recent past, the software industry has been facing several challenges including dynamic customer demands, complex software requirements, tight project schedules and constraints on resources and budget. This calls for a pragmatic approach to software development that delivers high-quality software products within the allocated budget and time schedule. Scrum is one such methodology that manages the software development in various short iterations known as sprints. Each sprint includes all the phases of a software development lifecycle model such as designing, implementation, testing, customer review, etc.

As per the survey conducted by the French Scrum User Group [10] in 2009, 86% of the respondent companies advocated the use of Scrum based development. The findings of the 8th Annual State of Agile Development Survey [3] conducted by Versionone.com in 2013 reveal that 73% of respondents agreed to practicing Scrum and Scrum variants for software development. The survey done by O. Salo and P. Abrahamsson [5] focussed on embedded software development companies across Europe with 27% of them responding in favour of Scrum methodology. The characteristics unique to the Scrum based development are:

4.1.1 Collaboration

Scrum based development promotes collaboration as it is driven by cross-functional teams where every person with his or her skills and experience, contributes towards the best design solution. A cross-functional team includes a mix of programmers, software architects, software analysts and QA experts.

4.1.2 Daily Meetings

Scrum methodology is marked by short-duration daily scrum meetings where the product development team communicates and evaluates the progress status of software development, thus increasing productivity of team members.

4.1.3 Product Backlog

The product backlog captures the requirements for a software product to be delivered successfully. It maintains an ordered listing of features, bug fixes, non-functional requirements.

4.1.4 Sprint Backlog

The sprint backlog records the list of tasks to be performed by the development team during the next sprint. This list is drafted by picking up tasks from the top of the product backlog until sufficient work is arranged for the next sprint, considering the work capacity and past performances of the development team.

4.1.5 Roles

The scrum based development is governed by 3 primary roles:

4.1.5 Product Owner: Responsible for defining, prioritizing and communicating product requirements and guides the product development process.

4.1.5 Development Team: Responsible for executing the tasks allocated by the product owner within the sprint deadline. Usually, a cross-functional team of 3 - 9 individuals implements the product development tasks envisioned by the product owner.

4.1.5 Scrum Master: Responsible for enforcing the rules and principles of Scrum based development. The Scrum Master removes impediments to development and helps improve the process, development team and software product being developed.

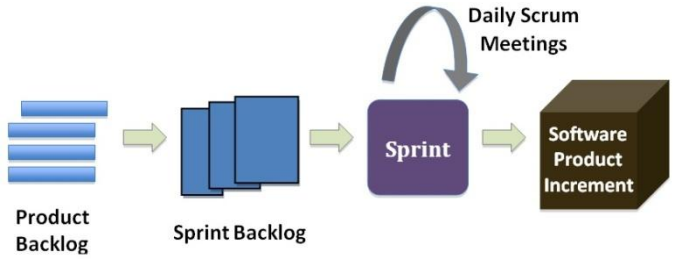

Figure 2. Scrum based Development

4.2 Extreme Programming (XP)

James Newkirk [11] defines Extreme Programming (XP) as a lightweight methodology that facilitates planned and iterative software development by small teams of developers to achieve higher software quality and enhanced productivity, in response to rapidly evolving requirements. Extreme Programming is characterized by intense levels of interaction with customers during the software development process.

The French Scrum User Group [10] conducted a survey on the usage and usefulness of Agile methodologies in France in 2009. The respondents included 230 professionals from over 150 companies, with the survey concluding that 52% of companies advocated the use of Extreme Programming (XP) approach. Another survey undertaken by O. Salo and P. Abrahamsson [5] among the embedded software development organizations across Europe revealed that 54% of the respondent organizations were using Extreme Programming approach. The distinguishing features of Extreme Programming (XP) that make it stand apart from other Agile approaches are [12, 13, 14]:

4.2.1 Requirements as Story Cards

The requirements are represented as scenarios by users, which are then formulated as Story Cards. The developers split each story card into a series of small tasks, which are further prioritized by the customer for being implemented.

4.2.2 Simplicity

XP favours initiation of software development with the simplest design, while additional functionalities can be added as and when required by the customer. Further, a simple design and simple coding can be easily understood by the team developers.

4.2.3 Continuous Interaction

XP includes extreme levels of customer interaction via established feedback loops. The customer engagement occurs in small frequent iterations, ensuring that the customer remains acquainted with the progress of software development. Also, this allows for quick accommodation of changes in software as per customer feedback.

4.2.4 Test Driven Development

XP employs test driven development where test cases for a task are written before its coding takes place. Testing remains an integral part all through the Extreme Programming method.

4.2.5 Refactoring

XP encourages striving for best design and high quality solution by refactoring of existing solutions, thus achieving enhanced code reliability and reduced complexity.

4.2.6 Pair Programming

Another unique concept in XP is pair programming where programmers work in dynamic pairs of two, resulting in enhanced communication and reduction in working hours and workload.

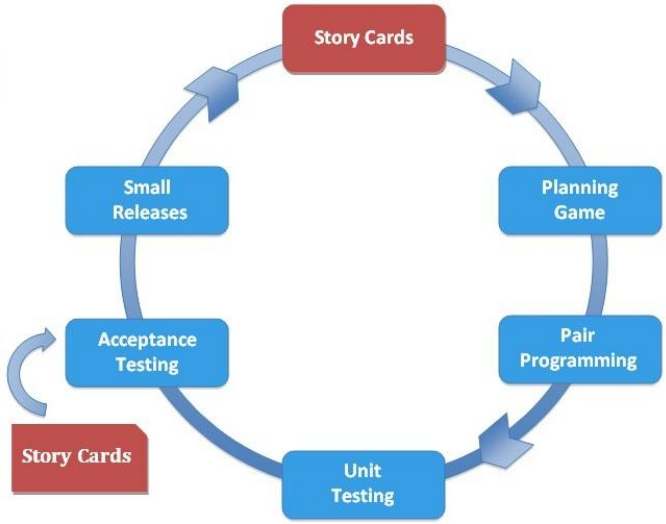

Figure 3. Extreme Programming (XP) Approach

4.3 Kanban

Among the agile group of methodologies, the Kanban methodology is significantly gaining popularity in the software industry. As pioneered by David J. Anderson et al. [15], Kanban provides a means to visualize and limit the work-in progress during software development process. Kanban method lays emphasis on scheduling of work so as to facilitate the delivery of software product just-in-time for implementation. Organizations across the world are adopting Kanban and absorbing it in their present software development processes to better model business agility. The characteristics that distinguish the Kanban methodology from other Agile based methodologies are [16]:

4.3.1 Kanban Board

Kanban board is a workflow visualization tool that enables the optimization of work and guides the workflow by dividing the work into categories, including to-do works, in-progress works and works done.

4.3.2 Maximizes Productivity

Kanban software development approach promises workflow optimization and scheduling, maximizing productivity of the team by reducing idle time.

4.3.3 Continuous Delivery

Kanban methodology is closely related to continuous delivery of software increments instead of releasing functionalities in batches. Release of small parts of product in successive iterations is directed at meeting the dynamic requirements of customers.

4.3.4 Waste Minimization

In Kanban approach, tasks are executed only when they are actually required. This results in elimination of over-production and cuts down on wasted work and wasted time.

4.3.5 Limits Work in Progress (WIP)

The main objective in Kanban methodology is to limit the Work-in-progress so as to optimize the workflow of the system in accordance with its capacity. A WIP constraint can be applied either to parts of the workflow or to the entire process.

5. COMPARISON AMONG AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

Although several methodologies follow the same set of principles as formulated by Agile manifesto [17], but they differ on various parameters of Agile principles. Empirical study of Scrum, XP and Kanban methodologies of Agile software development has resulted in the comparison as presented in Table 2.

Table 2. Comparison among Scrum, XP and Kanban Agile Methodologies

Parameters	Scrum based Development	Extreme Programming (XP)	Kanban Methodology
Design Principle [19]	Complex Design	Simplification of Code & accommodation of unexpected Changes through Refactoring	Limits the amount of Work-in-Progress & ensures Waste Reduction
Nature of Customer Interaction [18]	Not compulsorily on-site	On-site Customer Interaction	Not compulsorily on-site
Design Complexity	Complex design	Simple design	Simple visual design
Project Coordinator [18]	Scrum Master	XP Coach	Team Work
Roles Assigned	3 Pre-defined roles: Product Owner, Scrum Master & Development Team	No prescribed roles	No prescribed roles
Process Ownership	Scrum Master	Team ownership	Team ownership
Product Ownership [18]	Product Owner is responsible for product	Group responsibility of product	Group responsibility of product
Team Collaboration	Cross functional teams	Self organizing teams	Team comprises of specialized resources
Work flow Approach	Iterations (sprints)	No iterations. Task flow development	Short iterations
Requirements Management	Requirements Managed in form of artifacts through Sprint Backlog &Product Backlog	Managed in form of Story Cards	Managed using Kanban Boards
Product Delivery	Delivery as per Time boxed sprints	Continuous Delivery	Continuous delivery
Coding Standards	No coding standards	Coding standards are used	No coding standards
Testing Approach	No formal approach used for testing	Test driven development, including acceptance testing	Testing done after implementation of each work product
Accommodation of Changes	Changes not allowed in sprints	Amenable to change even in later stages of development	Changes allowed at any time

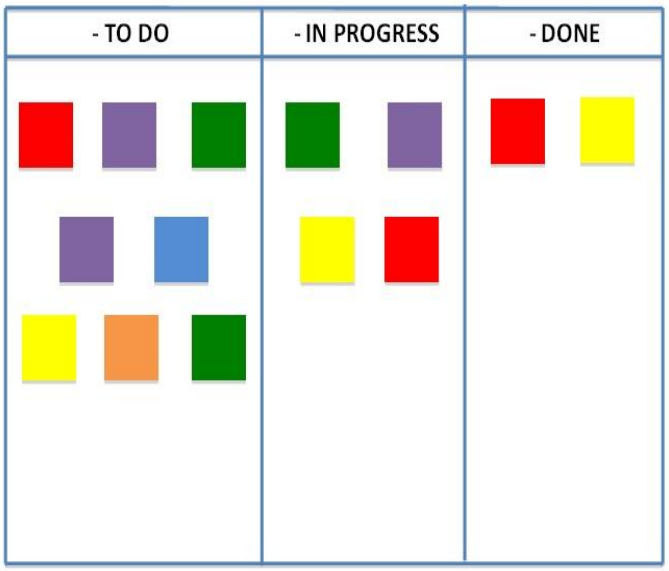

Figure 4. Kanban Board used in Kanban Methodology

A mix of telephonic and email conversations were conducted with the representatives of 15 software development companies. As per the survey results, 10 companies were found to be following Scrum based development, while 3 of them expressed their faith in Extreme Programming approach and the remaining 2 companies have adopted Kanban approach to software development.

Table 3. Companies practicing Scrum / XP / Kanban Agile Methodologies

Agile Methodology	Company Names
Scrum	• Bebo Technologies Ltd. (www.bebotechnologies.com) • Xebia IT Architects India Private Limited (www.xebia.com) • NTT Data (www.nttdata.com) • Bright labs. Ltd. (www.brightlabs.com.au) • Capgemini India Pvt. Ltd. (www.capgemini.com) • Collabnet (www.collab.net) • Stefanini (www.stefanini.com) • Deloitte Touche Tohmatsu Limited (www.deloitte.com) • SAP Software & Solutions (www.sap.com) • Toshiba Software Pvt. Ltd. (www.toshiba-tsip.com)
Extreme Programming (XP)	• Vismaad Pte Ltd (www.vismaad.com) • Thoughtworks (www.thoughtworks.com) • GlobalWorldTech (www.globalworldtech.com)
Kanban	• Stormpath (www.stormpath.com) • Spotify (www.spotify.com)

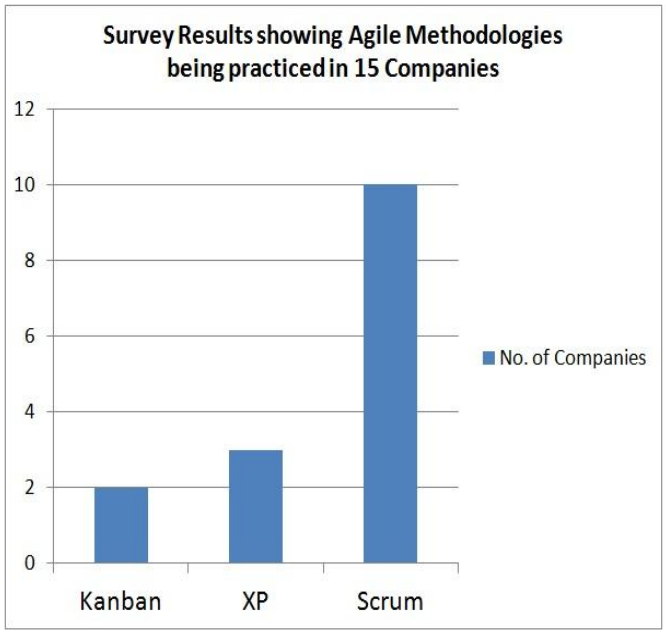

Figure 5. Bar Graph depicting that Scrum is being more practiced in comparison to XP and Kanban

6. CONCLUSION

Agile based software development provides an effective solution to the challenges being presently faced by the software industry including ever-increasing software complexity, dynamic user requirements, low budgets and tight schedules. The sudden increase in adoption of Agile methods across the software industry proves their usefulness and effectiveness. Agile processes ensure frequent interaction between developers and customers, besides offering value addition and improved return on investment, along with quick responsiveness to changing software demands. With our empirical study, we intend to contribute to the already existing body of knowledge about the comparative analysis of agile methodologies. Our survey results demonstrate a clear trend towards the higher adoption of Scrum based development in comparison to other Agile variants such as Extreme Programming and Kanban. The comparative analysis of three Agile methodologies, namely Scrum, XP and Kanban conclude that although the Agile family consists of several software development approaches which share the same set of Agile principles, but they do differ on various parameters. Among all Agile methods, Scrum has the highest adoption whereas Extreme Programming is certainly picking up pace with software practitioners starting to exploit it to their advantage, and Kanban approach to software development is increasingly being explored for addition to existing Agile software development process.

7. REFERENCES

[1] Kevin Roebuck, SDLC Book – Systems Development Life Cycle (SDLC): High-impact Strategies - What You Need to Know:Definitions, Adoptions, Impact, Benefits, Maturity, Vendors

[2] Standish Group 2011. The Crisis in Software: The Wrong Process Produces the Wrong Results, *The Standish Group Report* (2011). www.controlchaos.com/storage/S3D%20First%20Chapter.pdf.CHAOS report

[3] Versionone. 8th Annual State of Agile Survey. (2013) http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf

[4] Xebia. Agile Survey. (2013) http://xebia.com/news/agile-survey-2013/

[5] Salo, O. and Abrahamsson, P. 2008. Agile Methods in European Embedded Development Organizations: a survey study of Extreme Programming and Scrum, *IET Software*, vol. 2 (February, 2008), 58-64. DOI = 10.1049/iet-sen:20070038.

[6] Begel, A. and Nagappan, N. 2007. Usage and perceptions of Agile software development in an industrial context: An exploratory study. In Empirical Software Engineering and Measurement, (Washington, 2007), 255 - 264. DOI= 10.1109/ESEM.2007.12.

[7] Azizyan, G., Magarian, M.K. and Mattson, M.K. 2011. Survey of Agile Tool Usage and Needs. In Agile Conference (AGILE) (August 7-13 2011), 29-38. DOI=10.1109/AGILE.2011.30.

[8] Schach, S. 2007. Software Engineering, Tata McGraw Hill, Ed. 7, 4-6.

[9] Murphy, B., et al.. 2013. Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?, Empirical Software Engineering and Measurement, ACM / IEEE International Symposium,(2013), 75-84. DOI=10.1109/ESEM.2013.21

[10] French Scrum User Group 2009. A National Survey on Agile Methods in France. (June 2009) www.frenchsug.org

[11] Newkirk, J and Martin, R.C. 2001. Extreme Programming in Practice, Addison-Wesley, Ed. 1.

[12] Sommerville, I. 2005. Software Engineering, Pearson, Ed. 7, 26, 418 – 430.

[13] Beck, K. 1999. Embracing Change with Extreme Programming. *Computer*, vol.32, 70-77. DOI=10.1109/2.796139

[14] Shore, J. 2007. The Art of Agile Development, O'Reilly Media, Ed.1, 3, 15 – 44.

[15] Anderson, D.J. 2010. Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, Ed.1.

[16] LeanKit. 2014. Maximize Your Time, Improve Efficiency with the Kanban System. http://leankit.com/kanban/kanban-system/

[17] Beck, K., et al. 2001., Manifesto for Agile Software Development. http://Agilemanifesto.org/

[18] MSDN Library, Agile Principles and Values by Jeff Sutherland, *Microsoft Developer Network White Paper*. http://msdn.microsoft. com/en-us/library/dd997578.as

[19] Qumer, A., Henderson-Sellers, B. 2006. Evaluation of XP and Scrum using the 4D analytical tool (4-DAT), European and Mediterranean Conference on Information Systems (July 6-7 2006).epress.lib.uts.edu.au/research/bitstream/handle/.../2006005 499.pdf?