IN-DEPTH EXPLORATION OF AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

ABSTRACT

Comparative analysis of software development models, also known as the Software Development Life Cycle (SDLC), is a frequent discourse among software engineers. Various software development methodologies, such as prototyping, spiral development, and Rapid Action Development, are recognised within the field. Agile methodologies, including Dynamic System Development Method (DSDM), Scrum, Feature-Driven Development (FDD), Extreme Programming (XP), Kanban, Adaptive Software Development (ASD), Mendix, Lean, and Crystal, have gained traction in delivering software products within designated timeframes. ASD, DSDM, XP, FDD, Kanban, and Scrum emerge prominently among Agile methods utilised by software developers. This study individually examines and contrasts six Agile software models, elucidating their functionalities. The findings aim to assist software industries in making informed decisions regarding software development models for upcoming projects.

KEYWORDS:XP, DSDM, FDD, ASD, ASD, AGILE METHODOLOGY ANDKANBAN

INTRODUCTION

Agile SD has become common in recent times. Agile methodology SD increases quality and practical customer requirements. Traditional SD, such as Waterfall, prototyping Model and Rapid development, do not apply in SD nowadays because of frequent change requirements inthe business real field [1].SD methods and life cycles are terms used in software engineering to define the shapes through which software develops. A software process model abstractly demonstrates the design or characterisation of the software procedure[2].Agile principles focus on customer satisfaction, stakeholder inclusion, and development collaboration[3].XP, FDD, DSDM and ASD have flexible characteristics toaccommodate frequent changes, fast delivery and enhanced quality in software industries.

Agility in SD is a system's ability to select and adapt quickly to numerous variations. Agile SD processes are eagerly embraced, supporting the highest flexibility grade [4].This research gives comprehensive results on applying a critical framework to the agile SD model to determine each model's features, characteristics, and similarities. Agile SD processes are iterative to software requirements, development and software products.Dynamic business and customer requirements are often modified to suit frequent trends. It is necessary to adhere to this flexible model to be compatible with system requirement changes. Agile SD methods tend to change system requirements regularly within a particular period. The agile SD methods develop systems and produce the required customer results. This method delivers functioning software to customers in an agreed period. After the customer proposes,the project team will implement the new requirement in the next iteration.Agile processes focus on agility (rapidly and efficiently responding to changes) for SD. Possible changes in software projects are in needs, budget, schedule, resources, technology and team [5]. In an agile system, the emphasis is on system coding to meet customer demand. In agile methodology, system code is the medium of interaction and documentation between the user and the computer.

LITERATURE REVIEW

Software development includes analysis, design, implementation, testing, maintenance and documentation in delivering software products. There are different definitions of software engineering; the software engineering domain concept was introduced several years back. Most Agilemodels have been extensively embraced by industry, research, and publication within the software engineering domain over recent years. This technique was introduced in 1975 based on iterative enhancement, which became an agile methodology to overcome the heavy nature of developmental processes[6]. Agile methods are evolutionary and centred on iterative improvement andunscrupulous development processes[7].These features characterise this software development model: direct, cooperative, adaptive and incremental. It is collaborative in the sense that the developer works closely with customers. Incremental involves small software deployment with a rapid software development cycle. Because the model is simple and straightforward and can cope with modification.Due to flexibility, it can cope with customer requirements during software development. Traditional software methodologies follow the basic requirements of design, build, and maintenance pattern. According to[2], job design and control

have an optimistic alliance with worker well-being. Additionally, even though the author hypothesises that a worker in a business can be promoted from job monitoring, a high level of monitoring has an unenthusiastic effect on interests. In software engineering, agile development methods focus on the quality design field [5].Due to the lightweight practices of agile, it is the focused approach in many industries, and agile processes aim to handle and effectively manage those requirements iteratively field [8].As a result of frequent changes in technology and business environments, it is a challenge for traditional software development to meet complete basic requirements in advanced fields [9].Agile software methodologyemphasises developer-customer interaction in the software development process. It has been widely approved by developers after introducing in 2001[10].Customers' involvement in this process certify software performs according to customers' needs and expectations.Agile methodology is a family of lightweight software development procedures that include Adaptive Software Development (ASD), Lean Software Development, Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM) Crystal, and Scrum [11].In Agile procedures, interaction with personnel is more significant than processes and tools. Working software is more significant than comprehensive documentation, and collaboration with clients with the requirements is more important than contract negotiation, and being responsive to changes is more vital.Traditional software development methodology and agile software development methodology

OVERVIEW OF THE MODEL

Different Agile Methods have been introduced over a period of time, and these types work within a particular category of software domain:(a) Scrum (b) XP (c) DSDM (d) FDD (e)ASD (f) Kanban

SCRUM

In 1995, Ken Swaber introduced the scrum model methodology. Scrum is part of agile methodology since it entails most agile software development concepts. Scrum is made up of a team where team members collaborate. It delivers software products on time and with reduced charge. Scrum is a mostly used agile software development method and perhaps one of the most

used ones by[12]. It is claimed Scrum perform impeccably for teams of 5-7 people. This approach can be adopted by a developer, too. Scrum exhibits an iterative software model that follows certain roles and responsibilities with an unchanged meeting agenda. Spring usually lasts for two weeks for the team to deliver software productsregularly.The Scrum model is the perfect short-term project software development suite and best for organization-accepted practices and approval. The scope reduces risk failure and enhances transparent, reliable and trusting relations with the software development team and customers. It allows customers to modify requirements during the software development process. SCRUM project management structure ensures a motivated team working on building the customer's most important[13]. Scrum presents the concepts of flexibility, adaptability and productivity. Scrum focuses on how the team members should deliver better software products that are flexible to continuous environmental requirements change [14]. Scrum does not stipulate identifying features, just that there is a list of such features. Iterations can last a week and one month, with three to eight sprints before product release. Scrum also entails client involvement as the "product owner." The Scrum software development method is ideally suited to frequently changing customers' requirements [15].Scrum software methodology manages software procedures in short iterations, known as sprints. Each sprint involves all software development life cycle phases, such as designing, implementation, testing, and customer review[16].

EXTREME PROGRAMMING (XP)

Extreme Programming is commonly referred to as XP. Extreme Programming is a family of agile software methods intended to improve a type of software development intended to improve quality and reaction developing customer requirements. The ideologies behind XP comprise customer feedback, assuming straightforwardness, and acceptance of change in its development processes. Preliminary efforts in the development stemming from the traditional software development model have shown to be unfeasible. The customer doesn't know all the functionalities and qualities of advancedcommunication software. Requirements should always be modified to be corrected, identified, and re-defined as the software is developed. The primary motive for this model is the development idea of designing extreme programming software code -writing and testing.Extreme programming methodology is planned for smaller teams with two

to ten members working on regular or less recognised projects [[17]. Extreme programming is the programming category that helps agile lightweight software development. It became common during these recent modern days[1].All developers work carefully collectively to facilitate better interconnect informally instead of spending time recording designs and decisions. As teams remains lesser, this approach work perfectly. It's faster to converse straight than to document the development knowledge[18].Although software requirements are constantly changingin extreme programming, software development success is the sole priority. The originality of extreme programming is centred on the way the individual performance is composed and lined up to purpose with each other. Some core features of Extreme programming are small iterations with small issues and rapid feedback, close customer contribution, persistent communication and organisation, continuous refactoring, continuous integration and testing, collective code ownership, and pair programming [19]. Design, Code, and Test are four phases of extreme programming methodology, and the prime motives of XP methodology are judicious distribution, inexpensive refactoring, and correctness for developing small-scope software using small-scope teams [20].

FEATURE DRIVEN DEVELOPMENT (FDD)

Feature-driven development was introduced in 1997 and was part of a lightweight, iterative software development process. FDD is an iterative and incremental software development process that merges manufacturing best practices into one method. FDD has five basic features: advance model, construct feature list, plan feature, design by feature, and build by feature. Feature-driven development (FDD) is process-oriented and client-oriented. FDD software development is a process-oriented and client-centred agile method that develops software according to client needs [3]. FDD exhibit an adaptive and incremental nature to implement the essential functionality within short and possible iterations. The primary aim of this FDDmodel is to emphasise designing and building aspects of software development with more stress on quality. The primary aim of this model is to stress quality and deliver frequent, tangible working results at all steps of its delivery of the system. FDD provides precise,meaningful advancement and rank figures with the smallest overhead and disruption for the designer's[21].FDD differs

significantly from the other methodologies in the development context because it strongly emphasises upfront planning and design[4].

DEVELOPMENT SYSTEM DEVELOPMENT METHODOLOGY(DSDM)

The method of software delivery structure used for developing software packages and non-IT solutions normally embraced this model. It reports common practice failures in information technology projects, like going over financial plans, missing targets, and the absence of customer involvement. The primary target of DSDM principles are to focus on the business requirements, on-time delivery of software package, cooperative, uncompromised quality, build incrementally from strong foundations, advance iteratively, continuous communication and demonstrate control. The DSDM approach provides a comprehensive structure for developing and maintaining software systems and meets the duration schedule through incremental, iterative prototyping in an organised project environment[22]. The fundamental notion behind DSDM is fixing the amount of functionality in software products, adjusting time and resources to reach functionality; it is preferred to fix duration and resources and adjust the amount of functionality accordingly[19]. The origins of DSDM are rapid application development models. DSDM is seen as the first truly agile software development method introduced.DSDM addressed to overcome collective difficulties such as late delivery and overcharge[23].

KANBAN

Visual signs or cards signify and designate Kanban. The visual context used to implement Agile displays what to produce, the duration of production, and numbers of production. The Kanban philosophy in software development is Just in Time. Manufacturing Production Company uses this method to produce the needed product of the correct quality in a precise place and at a very exact time. It inspires fewer incremental modifications in the current system and does not involve certain procedures.It initiatives development teams to visualise project workflow, reduce work in progress (WIP) at each workflow stage, and quantity iteration[2].Kanban Methodology has recently gained much recognition in many systems, such as manufacturing, gathering, and supply sequence systems, effectively [24]. Solely concentrations are on the reduction of unwanted in all informs over-production, needless motion, flaws, processing & waiting. This

model provides a means to visualise and reduce work-in-progress during development. It focuses on work scheduling to facilitate exact implementation and issuing software products to customers. Organizations around the globe are adopting Kanban and absorbing it in their present software development processes to enhance model business agility[16]. The Kanban process explicitly presents the most significant tasks that need the greatest attention to lessen the risk and increase flexibility between other tasks in the project[25].

ADAPTIVE SOFTWARE DEVELOPMENT

James A. HighsmithAdaptive introduced Software Development (ASD),which exhibits the features of agile software development methodology with high speed and high change in developing software projects[26]. ASD is a modified approach of the extreme programming model, the most extensively used agile model. ASD focus on issues in complex software development processes, mostly in large systems. This method emphasises progressive and stage-wise development with stable prototyping. ASD supply framework or an approach holding suitable enough guidance to avoid projects.Adaptive software development works within an environment where requirements are not certain. Frequent change in business requirements and rapidly changing markets leads to uncertain requirements. Traditional methodologies are unstable as evolving markets are unstable. ASD's leading software development life cycle characteristics are continuous learning and extreme teamwork among developers, testers, and customers. ASD focuses more on products and their quality than tasks performed. ASD powerfully inspires incremental, iterative development with a continuous prototyping[27].

COMPARATIVE ANALYSIS

Although several agile methodologies have been introduced, each is applicable or applies to a precise set of projects. Numerous factors must be considered to develop a software project, such as project complexity, size, budgetallotted time,etc. Selecting the correct methodology for software development depends on many factors. A comparative analysis of agile methods will help to pick which model to select based on a certain condition and available resources.

DOCUMENTATION

In the agile software approach, one of the important principles of this methodology is to lessen the amount of time consumed on documentation. Although documentation is lessened, it is essential in software development and cannot be ignored completely during the developmental processes. Comparatively, documentation is of the least priority in these methods (Scrum, XP and ASD) compared to other projects under FDD involving more documentation. DSDM and Kanbanrequire moderate documentation, which is still less than FDD.

THE COMPLEXITY AND SIZE OF THE PROJECT

Each of the software development methodologies is appropriate for a specific project. Kanban, XP and ASD are chosen for simple, small and less complex tasks. Kanban, XP, and ASD are suitable for projects where requirements keep changing in the product design. Scrum, FDD and DSDM areproperfor both simple and complex project

CUSTOMER INVOLVEMENT AND INTERACTION

An agile software development methodology gives prime privileges to frequent communication between the user and the customers. Each of the methodologies has the degreeto which it involves its customers in software development. XP and Scrum have greater involvement of customers in the development process. In ASD and DSDM, client or end-user involvement can be seen during the start and end of iteration. Kanban also communicates through the product owner, While FDD communicates with the customers through a report.

MEETINGS

Software developer interaction is very important in its development. The success of agile methodologies normally relies on effective communication between the group members. Meetings are informal in nature, and no documentation maintained is crucial to achieve a better and more comprehensive product. XP usesa programming technique that pairs communication among their team members. FDD and DSDM agile methodscount ondocumentation and reports

for communication. Face-to-face conferences are used for communication on both scrum and ASD. Kanban uses visualising processing in their major practice.

Table 1: Comparison of various Agile Methods

FACTORS	**SCRUM**	**XP**	**FDD**	**DSDM**	**ASD**	**KANBAN**
Suitable project size and complexity	large and complex problems	Small and simple project	Large scale projects	Complex, simple project.	Small and simple project	Small and simple project
Documentation	Simple and Basic	Simple and Basic	More than XP, Scrum, Kanban	Highest among all	Moderate	Moderate
Team work	5 to 7 members	2 to 12 members	4 - 20 but fluctuates with complexity	Not specifically address	Not specifically address	Not specifically address
Changes with an Iteration	Not allowed	Allows within their iterations	Allows continually	Allows and reverse	Expected and welcomed	Anytime
Transparency	Transparent.	Transparent.	Transparent.	Transparent.	Transparent.	Highly Transparent
Approach	Iterative, Incremental	Iterative, Incremental	Iterative	Iterative	Iterative, Incremental	Iterative, Incremental
Iteration cycle period	2-4 weeks	1-6 weeks	2 days-2weeks	In 20% of total time 80 % of product	4-8 weeks	2 to 4 weeks but focus on continues flow
Concurrent feature development	Possible	Possible	Possible	Possible	Possible	Possible
Major Practices	Scrum meetings	Simplicity, Pair programming Test driven developme	Object Modeling, Development by feature, use of	Time boxing, Moscow, Prototyping	Time boxing, Risk Driven, Feature based.	Visualizing processes

		nt	UMl diagram			
User involvement	Through product owner	Actively involved	Through reports	Through frequent releases	Through frequent releases	Through product owner

CONCLUSION

Agile methodologieshave mostly been used for developing software recently compared to traditional software development methodologies. Traditional development practice has countless limitations, which include being unable to adapt to common changes in user requirements and being incapable of working within a specific time frame and budget. Software development methodologies play a significant role in every software project. Change is necessary in software development activity and can occur due to constant changes in user requirements, which makes the agile method the most comprehensive methodology to be adopted.Product requirements must be specified clearly before development in traditional software development because it does not adhere to frequent changes. Considering the changing business environment, it is important that the development methodology used easily adapts to the frequent changes in end-user demands. However, we discussed six agile methodologies: Scrum, XP, FDD, DSDM, ASD and Kanban. We strongly believe choosingthe best method out of the rest for a specific project is paramount.

REFERENCES

[1] S. A. Butt, "Paci fi c Science Review B : Humanities and Social Sciences Study of agile methodology with the cloud,"*Pacific Sci. Rev. A Nat. Sci. Eng.*, vol. 2, no. 1, pp. 22–28, 2016.

[2] S. M. Saleh, M. A. Rahman, and K. A. Asgor, "Comparative Study on the Software Methodologies for Effective Software Development,"*International Journal of Scientific & Engineering Research, Volume 8, Issue 4, April-2017*, no. May. 2017.

[3] Z. Nawaz, S. Aftab, and F. Anwer, "Simplified FDD Process Model,"*I.J. Mod. Educ.*

Comput. Sci. 2017, 9, 53-59 Publ. Online Sept. 2017 MECS DOI 10.5815/ij, no. September, pp. 53–59, 2017.

[4] P. S. Shama, "A Review of Agile Software Development Methodologies,"*Int. J. Adv. Stud. Comput. Sci. Eng. IJASCSE Vol. 4, Issue 11, 2015*, vol. 4, no. 11, pp. 1–6, 2015.

[5] M. R. J. Qureshi, "An adaptive software development process model," no. August 2008, 2018.

[6] M. Ibrahim, M. J. Khan, and A. Salam, "Comparative analysis of scrum and XP in Pakistani software industry," vol. 2, no. 3, pp. 199–215, 2017.

[7] A. Model, "A Survey of Agile Development Methodologies," pp. 209–227, 2007.

[8] A. Khan, "The Impact of Agile Methodology (DSDM) on Software Project Management The Impact of Agile Methodology (DSDM) on Software Project Management," no. March, 2018.

[9] A. B. M. Moniruzzaman and S. A. Hossain, "Comparative Study on Agile Software Development Methodologies," no. May 2014, 2013.

[10] G. E. Iyawa and A. Coleman, "Customer Interaction in Software Development: A Comparison of Software Methodologies Deployed in Namibian Software Firms," pp. 1–13, 2016.

[11] G. Kumar, "Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies," 2014.

[12] M. Blom, "Is Scrum and XP suitable for CSE Development ?,"*Procedia Comput. Sci.*, vol. 1, no. 1, pp. 1511–1517, 2012.

[13] M. Umbreen, J. Abbas, and S. M. Shaheed, "A Comparative Approach for SCRUM and FDD in Agile," vol. 2015, no. 2, pp. 79–87, 2015.

[14] U. Kumari and A. Upadhyaya, "Comparative Study of Agile Methods and Their Comparison with Heavyweight Methods in Indian Organizations," vol. VI, no. June, 2013.

[15] A. Aitken, "A Comparative Analysis of Traditional Software Engineering and Agile Software Development," 2013.

[16] G. S. Matharu, "Empirical Study of Agile Software Development Methodologies : A Comparative Analysis," no. May 2019, 2015.

[17] R. Fojtik, "Procedia Computer," *Procedia Comput. Sci.*, vol. 3, pp. 1464–1468, 2011.

[18] F. Maurer and S. Martel, "Extreme Programming," no. February, 2002.

[19] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New Directions on Agile Methods : A Comparative Analysis," 2003.

[20] M. R. J. Qureshi, "COMPARISON OF AGILE PROCESS MODELS TO CONCLUDE THE EFFECTIVENESS COMPARISON OF AGILE PROCESS MODELS TO CONCLUDE THE EFFECTIVENESS FOR INDUSTRIAL SOFTWARE PROJECTS," no. December, 2016.

[21] J. A. Bukhari and S. M. Shaheed, "A Comparative Approach for SCRUM and FDD in Agile," no. January, 2016.

[22] A. Sani and A. Firdaus, "A Review on Software Development Security Engineering using Dynamic System Method (DSDM)," vol. 69, no. 25, pp. 37–44, 2013.

[23] S. B. Chapram, "An appraisal of agile DSDM approach," vol. 4, no. 3, pp. 512–515.

[24] R. B. Wakode, L. P. Raut, and P. Talmale, "Overview on Kanban Methodology and its Implementation," vol. 3, no. 02, pp. 2518–2521, 2015.

[25] H. Lei, F. Ganjeizadeh, P. K. Jayachandran, and P. Ozcan, "Robotics and Computer-Integrated Manufacturing Full length Article A statistical analysis of the effects of Scrum and Kanban on software development projects,"*Robot. Comput. Integr. Manuf.*, vol. 43, pp. 59–67, 2017.

[26] A. Kaushik, "DSDM and ASD Agile Methodologies," vol. 3, no. 9, pp. 2393–2394, 2016.

[27] S. Merzouk, S. Elhadi, H. Ennaji, A. Marzak, and N. Sael, "A Comparative Study of Agile

Methods : Towards a New Model-based Method,"*Int. J. Web Appl. Vol. 9 Number 4 December 2017*, vol. 9, no. 4, pp. 121–128, 2017.