A Study on Agile Process and Its Prototypes

Kavya Urs A[1], Ramya S K[2]
[1]*VI Semester, Bachelor of Computer Application, MMK & SDM Mahila Maha Vidyalaya, Mysore, Karnataka, India*
[2]*Assistant Professor, Department of Computer Science, MMK & SDM Mahila Maha Vidyalaya, Mysore, Karnataka, India*

Abstract **- In this Digital Era, with fast growing application developments, maintenance of these applications is of great concern to provide a flexible user environment. With the development of new software's, it is necessary to check every aspect related to the same which may involve enormous operations such as testing the software with randomly generated test cases, monitoring the actions involved in the App's, resolving the failures during the test cases, revision of the software based on current trend user requirements and so on. This paper makes an attempt to brief the evolution of software development and its implementation.**

Index Terms **- Monitoring, Process, Development, Implementation, Maintenance.**

I.INTRODUCTION

In the world of technological dominance and the overwhelming evolution of the Software Industries, need for rapid, tough and fast adaptive software systems is in high demand to attain the extremely progressing customer requirements and market needs. As per the survey of Statista Research Department, during the first quarter of 2021, number of apps in google play store, apple app store, windows store and amazon app store are 3482452, 2226823, 669000 and 460619 respectively [8]. These apps which are growing at an exponential rate, produce bulk codes during the software development process. The software comprises of working modules. These modules are interrelated and its proper functioning plays a crucial role in providing a feasible and secured app over the web. This action needs well versed tools for software development and maintenance. One such tool is the Agile process, which outfits the continuous cycle of iterations and gradational/phased technique for software development. The advancements in the technology inspires the developers to emphasize in enhancing the agility of the applications by deploying several test automations, user trials and many more operations which leads to swift accomplishments and a hassle-free usage. The SDLC (Software Development Life Cycle) process initiates building of Projects/Software's/App's and focuses on providing an optimum, ready to use real time application software by using iterative methods which is the present approach deployed by major MNCs. The rest of the paper is organized as follows. Section II has the Literature Survey. Section III presents the methodologies involved in Agile Process and its predecessor's and finally the conclusion in Section IV.

II. LITERATURE SURVEY

In the paper, scrum: an agile process, the authors have briefed on challenges faced by the IT companies while developing the software. The challenges are complexity of software development, time to market, unclear and evolving customer requirement, requirement of agility for responding to the changing market needs, value centricity and innovation. The agile process overcome most of the challenges by seeking customer feedback in every 3 to 5 weeks during the iteration process. The paper focuses on the comparison between the traditional waterfall model with the present agile methodology. The traditional method is plan driven and accept frozen requirements from the client end, whereas the agile method which uses the frameworks scrum and extreme programming is feature driven and the requirements are not frozen, which means that the client is free to give ideas to the software vendor as per the requirement of present trend during the software development process. The article gives knowledge about the high and low cost of changes, more and less client involvements in traditional and agile methodology respectively.

SCRUM is a technique of agile which accepts incomplete definitions of projects and provides flexibility to add new features during the development process. The characteristics of SCRUM involves non linearity, maximum flexibility, environmental adaptability and so on. SCRUM comprises of sprints which manages to complete a part of the project within 3-4 weeks which assures confidence at the client end. The phases of SCRUM process include pre-game, game and post-game or closure. The SCRUM team includes the stakeholders such as customer, manager, scrum master and the product owner. The SCRUM process is enhancement centric; customer focused and concentrates on higher team productivity [1].

In the paper, Agile Process and Methodologies: A Conceptual Study, the authors mention about the SDLC models such as waterfall, spiral and RAD (Rapid Application Development), after which Agile was introduced. The Agile requires less planning, and the tasks are divided into smaller increments. The characteristics of Agile include iterative, modularity, time boxing, parsimony, incremental, adaptive, convergent, collaborative and people oriented. The paper concerns about the XP (Extreme Programming), SCRUM and FDD (Feature Driven Development) which are the iterative methods defined and executed under Agile methodology. This iterative methodology is adaptive to changing environment, ensures customer satisfaction, needs less documentation and reduces risks of development. At the downside of Agile methodology, repetitive interaction with customer is required to build a successful software, also there is lack of documentation, it consumes time, the wastage of resources is due to constant changes and it is more helpful for management than developer [2].

In the paper, Agile Software Development Methodology, the author signifies iterative approach, project timeline, sprint planning, project progress, stakeholder management, identifying stakeholders, prioritizing stakeholders, stakeholder communication [3].

In the paper, Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management, the authors, have highlighted the other areas of software engineering such as usability engineering (UE), interaction design (IxD), user experience (UX) and extreme scenario-based design (XSBD) which is used in Agile teams. The paper focuses on Agile software development, which uses extreme programming (XP) and SCRUM. In the software usability, design decisions are made using unified modeling language (UML) which help developers to decompose the system. With the progress of human computer interaction (HCI), user experience (UX), user interface (UI) designs were developed. The paper discusses on practices integration, process integration, technology integration, people integration and social integration. The social interaction integration strategy includes the concepts like social capital which is assessed by social network analysis (SNA), network governance and knowledge management. The Agile usability team interaction framework are social capital and the framework, network governance and the framework, knowledge management and the framework. The framework evaluation analyses adaptive software development (ASD), usability, behavior, social capital and network analysis [4].

In the paper, Documentation strategies on agile software development projects, the authors have specified that the 'just enough' documentation on Agile process is a topic of debate by the proponents. The grounded theory life cycle (GTLC) was studied by 58 Agile practitioners from New Zealand and India to overcome challenges with the design of strategies by documenting electronic backups, decisions made by customers, business terminologies, traditional functional specifications and positive customer feedbacks. The duration of data collection varies from 2 to 12 months and it depends on number of people working on the project [5].

In the paper, An Approach for Agile SOA Development Using Agile Principals, the authors emphasize on developing quality software by implementing service-oriented architecture (SOA) methodology with Agile development principles. The SOA methodologies are service oriented modeling and architecture (SOMA), Zimmerman's methodology, service-oriented architecture framework (SOAF), service oriented unified process (SOUP), rational unified process, service-oriented analysis and design (SOAD). To establish core architecture, understanding prioritization of business goals, mapping business goals to business processes, extraction of quality attribute scenarios related to each business goal, determination of CBPS and selecting business processes for current release are required to provide

agility to SOA methodologies. This method was implemented to a system developed by Iran's global distribution organization wherein the stakeholder's feedback was viable and valuable to use this approach [6].

III. METHODOLOGY

The concept of software development life cycle (SDLC) evolved during 1970s with the introduction of waterfall model by Dr. Winston W Royce which is a linear sequential methodology, including phases such as requirement analysis, designing, implementation, testing, deployment and maintenance. During the software development, in the requirements phase, product necessities are acquired, designing phase ponders on software architecture, implementation involves coding, testing includes wide range of classifications which uses several test cases, deployment phase accentuates on execution and maintenance encompasses operations such as installation and proper working of the software. Waterfall methodology has no provision for variations or updating the modules during the software development process. Hence, with the requirement of changing needs in the field of software development, procedures such as spiral model (1986), rapid application development (RAD) (1991) and Agile methodology (2000) emerged in the field of development. The Agile methodology carries out features like short plan, fast delivery and redoing/modification acceptance. The combinational deed of self-organizing and cross functioning gives a quick quality product. The table below gives the evolution of different SDLC models

SL. NO	SOFTWARE DEVELOPMENT MODEL	YEAR OF INTRODUCTION	APPROACH
A.	Waterfall Model	1970s	Linear
B.	Spiral Model	1980s	Iterative
C.	RAD Model	1990s	Iterative
D.	Agile Model	2000s	Iterative

Table 1: Different SDLC models with year of introduction and its approaches

The approaches mentioned in Table 1 rely upon the data/information flow inside the system. The linear approach has limitations wherein the model is unable to admit any extra features that needs to be included during the development phases as per the requirement of the client. The decisions in this approach should be well defined before the commencement of the project phases. The iterative approach provides a broader flexibility to add new things during the development which overcomes the limitations of linear approach. The waterfall model works well for smaller projects. The spiral model which is iterative, accepts new requirements and is suitable for large systems. Since spiral model deals with large projects, the development cost is high when compared with waterfall and RAD models. RAD has prototype cycles which is well suited for medium scale projects. The phases of spiral model, picturized on four quadrants, includes determining objectives, identifying and resolving risks, development and testing, customer evaluation and feedback. The spiral model is suitable for projects that requires continuous risk analysis.

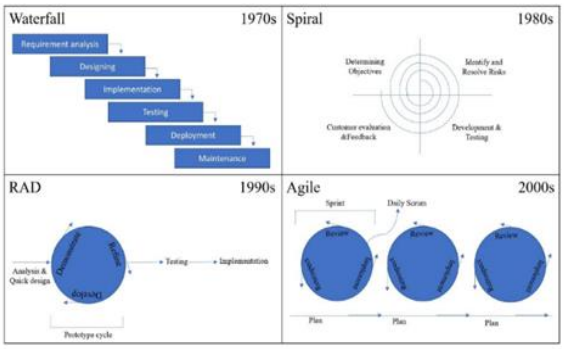

Figure 1: Pictorial representations of SDLC models

The Agile Methodology which emerged during 2000s have lightweight approaches such as scrum, lean software development (LSD), Kanban, extreme programming (XP), continuous integration (CI), continuous delivery (CD), feature driven development (FDD), test driven development (TDD) and crystal frameworks. The fuller approaches of agile include scrum-of-scrums, scrum at scale (Scrum@Scale), large-scale scrum (LeSS), scaled agile framework (SAFe), disciplined agile delivery (DAD), dynamic systems development method (DSDM), agile project management (AgilePM), agile unified process (AUP), open unified process (OPenUP). The lighter approaches have one team, whereas the fuller approaches comprise of multiple teams.

Scrum works on complex adaptive problems, divided into different blocks of tasks known as sprints which are operated one at a time in series of prioritization. LSD follows principles such as total quality management, just in time, six sigma perceptions

(business strategy, vision, benchmark, goal, statistical measure, robustness) and theory of constraints with benefits such as increases profit, fast improvement, improved capacity, reduced lead time, reduced inventory. Kanban is a rapid succession process which depends on Kanban board consisting of TO DO, DOING and DONE tasks. The developers keep a target of one month to parse through the predefined stages in extreme programming (XP) which embraces coding, pair programming (where two programmers work together), unit testing, pair negotiation, standup meeting, acceptance test, iteration plan and releasing plan. Continuous integration (CI) works on agile release train (ART) which has continuous exploration, integration and deployment in the continuous delivery (CD) pipeline. CI works on coding whereas CD is the after process which releases software on demand. FDD process has three planning phases (developing overall model, building feature list, planning) and two iterative phases- one feature per iteration (designing and implementing). TDD employs red, green, blue (RGB) cycle where fault code test (R), testing proper code after update (G) and refactoring (B) tasks are carried out. Crystal framework have variations like clear, yellow, orange, red and maroon wherein 6,20,40,80 and above 80 people work in a team on the basis of project intensity and preferences respectively. With the perspective of philosophical, technical and developmental aspects, the Agile process is in lead. Figure 2 shows the popularity of agile process in various sectors.

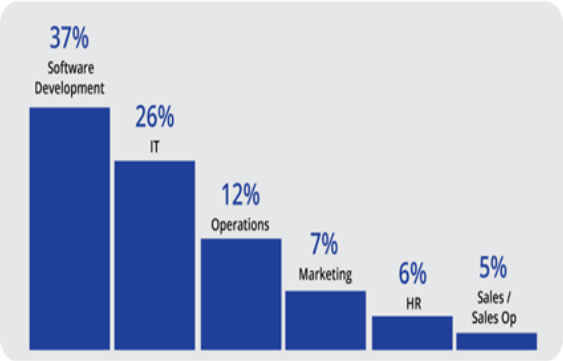

Figure 2: Sectors practicing Agile [7]

IV. CONCLUSION

This paper has made an attempt to examine the overviews of different methodologies that are used in SDLC to foster feasible software products, which is the necessity, as the number of app users are increasing every day. Providing error-free graphical user interface (GUI) applications and updating the software at frequent intervals as per user's requirement in the global market is a challenging task. This task is successful to a greater extent with the implementation of the Agile Process over its prototypes. In this generation of on-going software developments, increasing the efficiencies of the models is an unceasing course for the researchers. The advancements in the paradigm results in improvising the systems expansion. Over the decades, new reforms in the software development process are ongoing and more innovations in the field are awaited.

REFERENCES

[1] K Selvi, Rana Majumdar," SCRUM: AN AGILE PROCESS", IJRET: International Journal of Research in Engineering and Technology, Volume: 02 Issue: 03 | Mar-2013,

[2] Sheetal Sharma, Darothi Sarkar, Divya Gupta, "Agile Processes and Methodologies: A Conceptual Study", International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 05 May 2012,

[3] Charles Edeki, "AGILE SOFTWARE DEVELOPMENT METHODOLOGY", European Journal of Mathematics and Computer Science Vol. 2 No. 1, 2015, Progressive Academic Publishing, UK.

[4] Jeremy T. Barksdale, D. Scott McCrickard, "Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management", Int. J. Agile and Extreme Software Development, Vol. 1, No. 1, 2012.

[5] Rashina Hoda, James Noble, Stuart Marshall, "Documentation strategies on agile software development projects", Int. J. Agile and Extreme Software Development, Vol. 1, No. 1, 2012.

[6] Majlesi Shahrbanoo, Mehrpour Ali, Mohsenzadeh Mehran, "AN APPROACH FOR AGILE SOA DEVELOPMENT USING AGILE PRINCIPALS", International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 1, Feb 2012.

[7] https://www.knowledgehut.com/blog/agile/state-of-agile

[8] https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/
[9] https://airbrake.io/blog/sdlc/rapid-application-development