

A Comparative Study on Utilization of Scrum and Spiral Software Development Methodologies: A Review

Akanksha Mathur
Computer Science and Engineering
Govt. College of Engineering and Technology
Bikaner, Rajasthan, India

Archana Acharya
Computer Science and Engineering
Govt. College of Engineering and Technology
Bikaner, Rajasthan, India

Abstract—Software development life cycle (SDLC) is a framework that consists of detail illustration of developing, designing, maintaining and deploying a high quality software system. In SDLC project can be developed using traditional and agile structures. Spiral process model is referred as the traditional software development methodology and scrum is one of the agile software development methodologies. For reducing delivering time and improve the quality of software many companies are moving towards agile software development. The goal of agile is to translate the requirements to the form the customer will easily relate and gain more reliable feedback. The objective of this paper is making a comparison between spiral process and scrum methodologies to show their features and defects. Also examine the circumstances of using either spiral process model or scrum methodology while developing software project.

Keywords—SDLC; traditional; Agile; spiral model; scrum methodologies

I. INTRODUCTION

“The SDLC is a framework defining tasks performed at each step in software evolution process. International organization for standardization / International Electro technical Commission 12207 is an international standard for software life cycle process [1]. SDLC is a process followed for a software project, within a software organization. It comprise of outline describing how to develop, prolong, replace and alter or strengthen specific software. Life cycle is a methodology for improving the quality of software and overall development process. A selection criterion in SDLC is crucial as it ensures the organization to maximize the change to deliver software successfully.

SDLC introduces various traditional process models including waterfall model, iterative model, incremental model, prototype model, RAD, V model, spiral model. Traditional methodologies are plan driven in which work begins with selection and documentation of entire set of requirement specification followed by architectural and high level designed development and inspection. These methodologies found to be very complex due to these heavy aspects and when rates are relatively low in software development, practitioners posses difficulties in it. As a result, several consultants independently discovered

different methodologies and practices to embrace and respond to inescapable change they were experiencing. These methodologies and implementation are based on iterative improvement, a technique that was discovered in 1975 and introduced as agile methodologies.

The name Agile came in 2001 when 17 process methodologies discussed future in software development. Developers noticed that their method has many characteristics in common so they decided to name these processes Agile which means both “light and ample”. Therefore agile organization and its issuance for agile software development came into being. The Agile methods claim to place more emphasis on propel, interaction, working software, customer, collaboration and change rather than on processes, tools, contracts and plans. However empirical studies are urgently needed for evaluating the effectiveness and probability of using agile software development methods.

In today’s world problems are changing, community are changing, concepts are changing. This tends to move towards organizations in which we have more control over how we work and how we interact with peers, customers and management. This paper will investigate heavy weight and agile methodologies for their sustainability in software development and creates a review of data to determine which methodology suits best. Additionally tried to fill the gap by conducting a detailed review of both spiral and scrum methodologies.

II. TRADITIONAL V/S AGILE METHODOLOGIES

This section describes overview of traditional and agile software methodologies.

A. Traditional Software Development

Process models based on traditional software development methodology are waterfall model, iterative model, incremental model, spiral model, prototype model etc. These methodologies follow a sequential series of steps including requirement analysis, planning and documentation, designing, coding, maintenance and deployment. There are many different methodologies but this reference paper limits the discussion to the most significant methodology: Spiral Methodology.

Spiral Model:

Spiral model is traditional development model, which combines the elements of both designing and prototyping in stages. This was defined by Barry Bohem [1] based on experience with various refinements of the waterfall model as applied to the large software products. There are four different phases of spiral process model

1. Objective setting: Specific objective for the project phase are identified.
2. Risk assessment and reduction: Key risks are identified, scrutinize and information is obtained to reduce these risks.
3. Development and validation: An appropriate model is chosen for the next phase of development.
4. Planning: The project is reviewed and plans are drawn up for the next round of spiral.

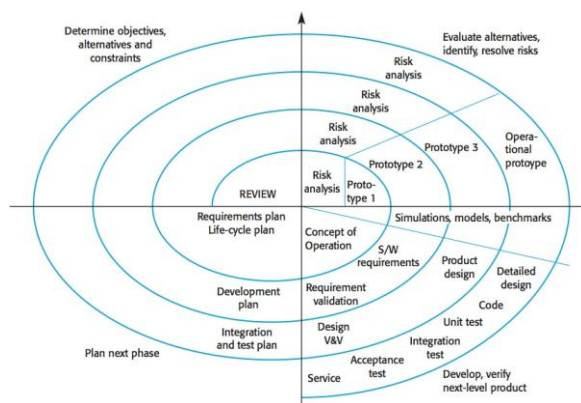


Figure 1: Spiral Model

Spiral process model also called “meta-model” because it consists of other models of SDLC. Waterfall and prototype both models are used in it. So we perform software development systematically over the loops and at the same time we make prototype and show it to the user after completion of various phases. In this way we are able to reduce risks, as well as we follow organized approach. Below discussed table1 for advantages and disadvantages of spiral model in detail.

TABLE 1: FEATURES OF SPIRAL MODEL

Advantages	Disadvantages
Good for large mission censorious projects.	Doesn't works well for smaller projects.
Software is produced early in software lifecycle.	Highly customized limiting reusability.
It is suitable for high risks projects where business needs may be unstable.	Cost involved in this model is usually high.
Strong approval and documentation control.	It is complicated approach specially for projects with clear SRS
More and more features are added in systematic way.	Skills required evaluating and review project from time to time need expertise.
Estimates (i.e. budgets, schedule etc) become more realistic as work progress because more important issues are discussed earlier.	Rules and protocols should be followed properly to implement effectively this model. Doing so throughout the span of project is

	tough.
Easy to change requirements and documentation, if any change happens in middle of development.	Due to various customizations allowed from the client using the same prototype in other project in future is difficult.
It is used when experimenting on technology trying out new skills and when is not able to offer requirements in clear terms	It is not suitable for low risks projects.
Here project monitoring is very effective and easy. Each phase as well as each loop requires a review from concerned people. This makes model more transparent.	Meeting, budgetary and scheduling, requirements is tough if this development process is followed
Control towards all phases of development.	Amount of documentation is required in intermediate stages makes management of project very complex affair.

A. Agile Software Development

Agile-denoting “the quality of being Agile, readiness for motion, nobleness, activities, dexterity in motion” as mentioned in oxford dictionary [3]. Agile software development is a group of software development methodologies which is based on incremental and iterative development where requirements a solutions evolve through flexibility and collaboration of independent teams. It is alternative to traditional project management typically used in software development. There are number of methodologies used to develop and improve the system named as Adaptive Software Development, Agile Modeling, Crystal Method, Dynamic System Development, Lean Development and Scrum. All these methodologies acknowledge that high quality software and more importantly customer satisfaction could only be achieved by bringing lightness to their processes. However this section will focus on: Scrum Methodology.

Scrum Model:

In 1995 Ken Schwaber and Jeff Sutherland presents a study based on software development method and called it “SCRUM”. Basically Scrum is derived from rugby where scrum restarts the gone after the ball has gone out for play. Applying this interpretation in technical terms a software scrum gets the team back together and everyone is moving in appropriate direction. Scrum is process framework that is used to manage complex product development [8].

1. Sprint Planning: The work to be performed in sprint is created by collaborative work of entire team.
2. Daily Scrum: The development team uses daily Scrum to inspect progress towards the sprint goal and to observe the progress is trending towards completing the work to meet the sprint goal.
3. Sprint Review: Sprint review is held at end of the sprint to inspect the increment and adopt the product backlog if needed.
4. Sprint Retrospective: It is an opportunity for the Scrum team to direct itself and create a plan for improvements to be enacted during next sprint planning.

SCRUM PROCESS

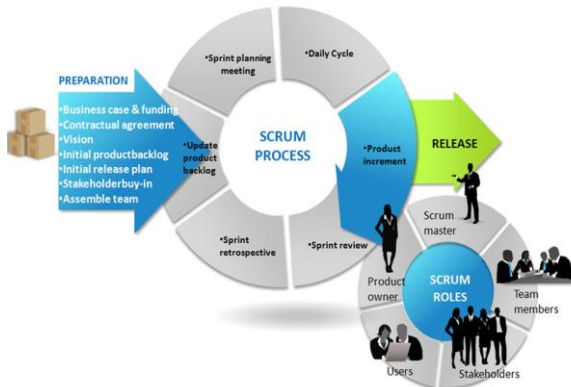


Figure 2: Scrum Model

TABLE 2: FEATURES OF SRUM MODEL

Advantages	Disadvantages
Scrum methodology enables projects where the business requirement documentation is hard to quantify to be successfully developed.	Adapting scrum framework in larger team is challenging.
It is lightly controlled method which insists on frequently updating the progress in work through regular meeting. Thus there is clear visibility of project development.	Daily meeting at times can go too strict and frustrating for team members.
Like any other agile methodology this is also iterative in nature. It requires a continuous feedback from user.	If any team member leaves the project in between, it affects the project badly.
Due to short sprints and constant feedback it becomes easier to cope with changes.	Quality is hard to implement till the team goes through aggressive testing progress.
Agile scrum can work with any technology/programming language but is particularly useful for fast moving web2.0 or new media projects.	This methodology needs experienced team members only. If the team consists of people who are novices, the project cannot be completed in time.
The overhead costs in the terms of process and management is minimized, thus produces quicker and cheaper result.	If the task is not well defined estimating project cost and time will not be accurate.
It emphasis on face to face communication, cross functional, self organizing team.	Usually established organizations have a hard time accepting scrum.
It is easier to deliver a quality product in schedule time.	The peer pressure within the team can be immense.
Works well for fast moving developments project.	Easy to slack and let discipline go.
Developments are coded and tested during sprint review.	Team needs to embrace the methodology, for this needs a very open and communicative mindset.

III. OVERVIEW

Hu Zhigen [4] discussed an example based on agile development methodology. This introduced how scrum (an agile method) being useful in software development and how it helps us to manage project efficiently as being adaptive process in many of the countries. It concluded that

Sonali Pathak [7] has taken a real time small case study on “COLLEGE MANAGEMENT SYSTEM” which shows scenario of development technology that applies human centric approach to software development. On the

due to self organizing and self managing structure the entire development process becomes more active.

Aitken [9] introduced differences between traditional and agile methodologies and technique which shows the approaches for modeling the model system. Accordingly agile software development helps and motivates emergent and evolving architecture whereas traditional software development encourages architecture with justification and evaluation of options. It concludes that there is nothing really incompatible with applying all the principals and values of agile software development along with most of the practices, to traditional software development.

Kumar G [13] The goal of software engineering is to develop a quality software product meeting deadline, maintaining budget and requirements of customers. The paper examines different life cycles of process models which aims showing advantages and disadvantages of both traditional and agile methodology with detailed description of adopting and knowing good and bad practices. It concludes that a highly critical projects having large number of developers with limited requirements and limited features can be developed by using traditional methodologies whereas projects having iterative and incremental deployment, enhancement in quality, easy fault detection capabilities developed by using modern methodologies.

Pozgaj Z [12] developed and described model based on scrum which is most popular agile methodology. Scrum is basically a framework which manages software project and application development which was compared with teaching process model based on 3 units: 1) theoretical knowledge 2) practical knowledge and student’s projects 3) exam and admission of grades. Each unit represents a scrum that’s results with particular product. It concludes that this teaching model will be useful in education system.

Agrawal A [14] introduced with agile methods and their growth from 2001 to 2014 in field of software development. They notified the journey of agile methods and vigorous demand in industries, organizations, people, jobs, professionals and other purposes. This comparison is helpful survey in acknowledging various social aspects and current scenario in large systems.

Akbar [15] discussed the trends of global software development and emphasis on need of shifting from traditional to agile methodologies. Therefore GSD (Global Software development) in modern technologies is challenging to meeting the requirements which are not fulfilled by conventional systems. In this regard software process tailoring provides flexible solution to modification and adjusting existing process and practices. Such as code quality, disruption in team work, mature/immature scrum, release process, backlog management, communication, scrum ceremonies along with solutions are identified for these above mentioned issues.

basis of this survey it was found that the most of the academic projects are facing a common problem of controlling, monitoring and management of small projects and there is a need to improve the project development in

this area. Also the study focuses on adaption of newer technologies in academic environment are a better method of development such as implementation of agile methods in the academic projects. This research paper concluded that the scrum methodology has proven the best solution for the issue of handling academic software projects.

Mahalakshami [10] has done a comparative study on Traditional SDLC v/s Scrum Methodology. The paper concluded that requirements of customer are changing frequently, so the software development is expanding and becoming more complex making the project development difficult. According to this research paper traditional SDLC is not able to meet the requirement of market so scrum methodology was introduced to the new requirements of software companies. Hence it states that it depends on not able to meet the requirement of market so scrum methodology was introduced to the new requirements of software companies. Hence it states that it depends on organization which model to use.

Luigi [5] presented the paper which was based on the report of applying Scrum in open source software java tool. The aim of the paper was to rectify these issues 1) Is it possible to switch successfully to the scrum methodology in an ongoing open source software development process? 2) Is it possible to apply Scrum when the developers are geographically distributed? 3) Does Scrum help improve the quality of the product and productivity of the process? The paper concluded by identifying the set of measures and by comparing the data before and after the instigation of Scrum. The result shows that the scrum can be introduced and used in an ongoing geographically distributed open source software process and it helps control the development process better.

Danh [11] presented the review of current literature on effort estimation in Agile, iterative, incremental software projects. It shows the affirmations about common trends, gaps and recommendations for future work. The research paper concluded the need for future research on estimating effort, analyzing the impact of historical and current data and empirical validation of estimation model.

IV. COMPARATIVE ANALYSIS AND DISCUSSION

On the basis of above literature survey, tabulated comparison between traditional, spiral, agile and scrum methodologies is being formulated to gather information on different parameters showing strengths and weakness of spiral and scrum process model useful for selecting which software model is appropriate according to the user requirement and definition.

TABLE 3: FEATURES OF TRADITIONAL, SPIRAL, AGILE AND SCRUM .

Features	Traditional Methodology	Spiral Model	Agile Methodology	Scrum Model
History	According to Elliott, Strachan & Radford traditional software development was originated in 1960.	This model was first described by Barry Boehm in its 1986 paper "A spiral model of software development".	Agile software development model was evolved in the mid of 1990's and in 2001, seventeen software developers published the manifesto for agile software development.	Scrum was first introduced in 1995 by Jeff Sutherland & Ken Schwaber. It was first tried and redefined at Fidelity Investment and IDX.
Requirement	All the requirements are stable	Requirements are compatible with all the key system, stake holders, and expectations including users, customers, developers, maintainers and investor. The nature of requirement will not change very much during development and evolution.	Eventual changes in user requirement according to demand in industry. Requirements are gathered in inception phase.	Requirements are based on meetings between product owner and developer. The product owner is only responsible for management of requirements. Requirements are therefore flexible.
Documentation	Every activity is measured by intensive documentation. Hence heavy documentation	Documentation is done in iterative manner at each phase.	Light documentation	Minimal document rather than no documentation. Level of documentation remains upto the team to decide on case-by-case basis.
Budget	Depends on process model and determined during planning.	Partially variable.	Depends on process model	Comparatively low and set during each sprint.
Conceptual difference	It is characterized by a sequential series of steps like requirement definition, planning, building, testing and deployment.	It is traditional software development model which combines elements of both incremental and prototyping in stages.	Agile is a group of software development methodologies based on incremental and iterative approach.	Scrum is simple set of roles, responsibilities and meeting that never change.

Simplicity	Simple and ease of working depends on various models.	High amount of risk analysis.	Simple and easy models.	Adaptive simplified approach.
Time frame	Long term.	Time is defined in terms of years and months.	Short term.	Project development time vary from days to weeks.
Efficiency	Efficiency is variable on the basis of requirements of software models.	Due to high risk analysis efficiency is low for large and complex software and high for middleware software.	High efficiency.	Efficiency may vary from case-by-case basis as it depends on daily scrum meetings. However scrum is more efficient.
User /customer involvement	Involvement is considered at the beginning.	From beginning to evaluation phase.	Continuous customer interaction is very essential to get proper product requirement.	Development of software product using scrum totally relies on product owner so it has high user/customer involvement.
Suitability	Suitable for the software products having clear list of requirements which are not likely to be modified.	Suitable for large and mission critical projects where risk analysis contains high priority.	Best suited for web based application where its iterative nature helps in incorporating, correcting various bugs arises over time.	Easy to change directions as the requirement changes.
Flexibility and Functionality	Limited cookbook approach.	Limited cookbook approach.	Limited cookbook approach.	Unlimited during iterations.
Testing	Performed after coding.	Testing is done in engineering phase.	It is performed at each iteration.	Done at each sprint.
Software development	Process oriented.	Process Oriented.	People oriented.	People oriented.
Team role assignments	Based on skill level of individual team members.	Highly skilled and experienced team members.	Self organizing teams.	

V. CONCLUSION

Software development methodologies came into existence since 1970's. The aim of this survey paper is to try to assure that if spiral process methodology and scrum methodology are compatible in terms of process, tools, interaction, implementation, functionalities, efficiency etc in software development or not. This paper surveyed that traditional methodology and agile methodology seem to be growing rapidly. Modern technologies demand for scrum methodology and it is up to organizations requirement which model to use. Therefore it is important for development team to select software methodology best suited for software projects.

REFERENCES

- [1] B.Boehm, "A Spiral Model of Software Development and Enhancement", IEEE computer, May1998.
 - [2] R.S Pressmen, "Software Engineering A practioner Approach", 5th edition, Newyork Mc-grawhill, 2001.
 - [3] [http:// WWW.dictionary.oed.com](http://WWW.dictionary.oed.com), Accessed 20/4/2005.
 - [4] Hu Zhigen, YUAN Quan, ZHANG Xi, "Research on Agile Project Management with Scrum Method",IEEE 2009.
 - [5] Luigi Lavazza, Sandaro Morasca, Davide Taibi and Davide Tosi, "Apply Scrum in An OSS Development Process: An Empirical Evaluation", Springer, 2010.
 - [6] Akif, R. H. Majeed, "Issues and Challenges in Scrum Implementation", IJSER, 2012.
 - [7] Sonali pathak, Pushpendra Pateriya, Preet Pal, "A Case Study on Software Development Projects in Academic Knowledge Centers using Scrum", IJCA 2012.
 - [8] K.Schwaber, J. Sutherland, "The Definitive Guide to Scrum: The Rules of Game", 2013.
 - [9] Ashley Aitken, Vishnu Ilango," A Comparative analysis of traditional Software Engineering and Agile Software Development, IEEE 2013.
 - [10] M. Mahalakshami, DR. M.Sundarajan,"Traditional SDLC v/s Scrum Methodology- A Comparative Study", IJETAE, 2013.
 - [11] Danh Nguryen -Cong, De Tran- Cao, " A review of effort estimation studies in agile, iterative and incremental software development", 2013 IEEE RIVF conference.
 - [12] Z . Pozgaj, N. Vlahanic and V. Bosiliz- Vuksic," Agile Management : A Teaching Model Based on Scrum", IEEE, 2014.
 - [13] Ashish Agrawal, Sadhana Singh , L.S Maurya, "A study on the Growth of Agile Methods n India till 2014", ICACEA 2015, IEEE conference.
 - [14] Rehan Akbar, Sohail Safdar, "A Short Reviewof Global Software Development (GSD) And Latest Software Development Trends", I4CT 2015 IEEE Conference.
 - [15] <http://WWW.itinfo.am/eng/software-development-methodologies/>
 - [16] <http://en.wikipedia.org/wiki/Agile-software-development>.
 - [17] <http://en.wikipedia.org/wiki/Scrum-model>.
- WWW.Scrumguides.org/history/html