An Empirical Study of Agile Software Development with SCRUM Model

Lohit K S, Rupali Wagh

Abstract— **Software is built to perform complex tasks with ease, efficiency and a lot of flexibility. Hence software building process requires lot of attention to detail and a general guiding algorithm which aims at attainment of objectives. Huge amount of time, effort and money are spent on building software. There exist many traditional software development models that have been in the market from years. But hot cake in market is called agile methodology which has paved way for revolutionary changes in software development models. Every organization is pushing its projects towards agile approach. The main purpose of this paper is to discuss agile software development framework scrum and to bring key advantages of agile that makes it better than waterfall.**

Index Terms—**Agile Methodology, Scrum, Software Development.**

I. INTRODUCTION

Modern software development is undergoing evolution. Though quality product development is the sole objective of any software building organization, this multifaceted process brings out challenges in every step. While traditional as well as agile development methodology are aimed at catering to the requirements of client to build a software system that provides all necessary functionalities, there are fundamental differences in these two major approaches.

A. Traditional software development – Waterfall Model

Traditional software development methodologies are broken into series of sequential steps which follows one after the other. The series of steps are requirement definition, planning, building, testing and deployment. As the very first step, client requirement is documented to a maximum extent. In The next step, general architecture is visualized, then the coding, followed by testing and last step would be deployment. The principle followed here is visualizing a finished project even before it is started, and work towards building it the exact same way how it was visualized.

Waterfall is a sequential development process where in one step is followed up by another. First phase has to be completed to in order to begin the second phase. The waterfall model was first presented by Winston W Royce in 1970. Since then waterfall model is used as a traditional development model across all software development projects. Generic steps of waterfall model are - System and software requirements, Analysis, Design, Coding, Testing and Operations.

System and software requirements phase is where the Business requirements are viewed or converted to proper system and software requirements. In analysis feasibility study id performed to decide upon which model, business rules should be applied to make the software work. In Design phase detailed architecture of the software to be built is laid out. Based on the architecture coding phase begins for each module. Once development is finished testing begins. With the end of testing phase the application is installed in production and later maintained.

B. Paradigm Shift - Agile Software Development Models

As the name suggests agile is less rigorous compared to the waterfall model. It follows an incremental and iterative development method where each phase is revisited again. Agile developers do not think of software as a huge structure but they view it as a piece of entity with complex moving parts interacting with each other. Adaptability and compatibility is more emphasized upon in agile software development. Most commonly used agile development model are:

- Scrum - Scrum is an iterative and incremental agile software development methodology. The key feature in scrum is customer feedback during development. In scrum each cycle of development is divided into sprints. At the end each sprint a demo is given to the customer. So if the customer wishes to make any changes it can be incorporated immediately.

- Extreme programming - This is a type of agile methodology which aims at improving quality and will be able to accommodate frequent changes in customer requirements. This is done with short and frequent releases.

- Lean development model - This model follows below seven principles: Eliminate waste, Build quality, Create knowledge, Defer commitment, Deliver fast, Respect people and Optimize the whole.

- Kanban - It is a framework which is implemented in agile. Kanban mainly focuses on planning. Here once a requirement is finished, then immediate requirement is picked from the backlog. The product owner takes care of this activity. He has flexibility to change what should go to backlog and which should not go.

Lohit KS Lohit KS completed his bachelor of science in mathematics and computer science, currently pursuing masters in Christ University, working as Consultant at CSG International.

Rupali Wagh is an Associate Professor of Computer Science with Christ University. She has over a decade of experience in academics. Her research interests include data mining, text analytics and natural language processing and mobile technology.

This paper focuses on scrum model of agile development, its basic concepts and processes and highlights its advantages.

II. RELATED WORK

Software development is a complex, multi-phase and continuous process. Though clearly defined activities for every phase of development act as base guideline, regular communication and feedback among all the stakeholders of the product to be developed is one of the key feature of success of software development. This communication supports the "evolving" aspect of the process and is seen as major factor contributing to the emergence of agile methodology. Regular and effective communication is the one of the focus point and is the driving force for the dominance of agile methodology since its arrival. Impact of agile methodology of software development has been studied widely. [8] gives a comprehensive discussion on the positive influence of agile methodology like SCRUM and XP on the overall software development process. Effectiveness of large scale off shore development process with agile – scum model has been discussed in [1]. Since 2010 agile software model has widely been considered as main stream development approach as per the survey conducted in [2] which also highlights benefits of agile along with the challenges. A detailed discussion on the effects of moving to incremental agile model has been discussed in [6]. Due to the iterative and incremental nature of agile processes, knowledge management throughout development requires a careful attention. Various knowledge management practices are discussed in [3]. Adaptation in Software testing practices file agile development has been discusses in [10]. Software deployment in agile methodology is reviewed in [7] with respect to challenges and strengths that contribute to software process improvement. Though there had been dominance of agile methodology in last two decades, there also has been criticism mainly due to the violation of traditional development theories and principles. [5] gives a comparative evaluation of globally distributed software development of structured and agile development approaches. Researchers have tried for a mid-way methodology [4] to propose approaches that give best of the two approaches.

III. SCRUM – TODAY'S PREDOMINANT AGILE FRAMEWORK

Scrum can be seen as extended spiral model of software development with enhanced communication and feedback at every iteration [9]. Customer feedback is an integral component of every activity in scrum development framework. It is very simple and its incremental and iterative approach provides ample scope addressing risks during the entire product development process.

A. Scrum Basics

Product development under scrum is based on following major concepts.

Sprint - Every development period is called sprint. It is usually two weeks where the developer picks his user story and develops the code for it. At the end of a sprint or two sprints hardening is done where QA does testing for the all user stories as a feature as whole.

Backlog - It is what yet is not taken up for development. This backlog will always help the team in picking up user storied in priority.

Feature and User Story - Feature can be considered as one entire module as whole like customer creation. A feature is broken down into many user stories. Like create business customer etc.

B. Roles in Scrum

Following well defined roles carry out the assigned responsibilities

Product owner: He/she represents the stakeholders and is the voice of the customer and its product owner's responsibility to ensure that team delivers what the business expects. The product owner writes user stories and prioritizes the tasks for the team.

Scrum master: He or she is responsible for delivery and acts as a bridge between cross region working teams to ensure delivery is not affected. He takes care of the release and delivery is on time.

Development team: The development team is responsible for delivering potentially shippable increments (PSIs) of product at the end of each sprint (the sprint goal). A team is made up of 3–9 individuals who do the actual work (analyses, design, develop, test, technical communication, document, etc.). Development teams are cross-functional, with all of the skills as a team necessary to create a product increment. The development team in scrum is self-organizing, even though there may be some level of interface with project management offices (PMOs).

C. Phases in Scrum

Following are major phases followed in scrum framework planning:

- Plan the work for the sprint
- Decide product backlog
- Set a four hour planning for two week sprint

D. Daily Scrum

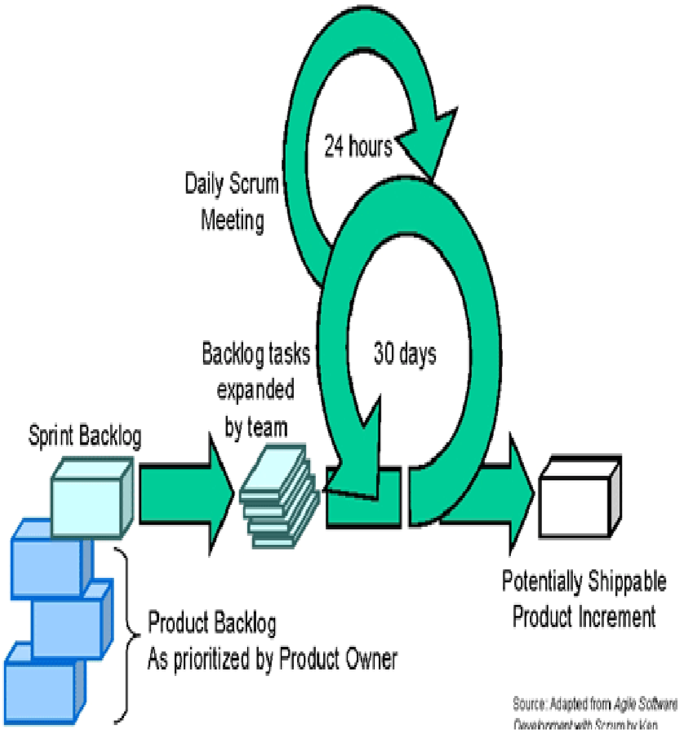
- Like a huddle where all team members are present
- Makes sure everybody are on time
- Plan of action for the day
- What we achieved yesterday

Review and Retrospective:

- Review the work completed and plan for the work which is not completed
- Handover to the stake holders
- What well in sprint?
- What can be continued?
- What process need to changed

- Any suggestions or idea

Workflow of scrum can be explained with the help of figure 1 given below


24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
30 days
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Potentially Shippable Product Increment

Source: Adapted from *Agile Software Development with Scrum* by Ken

Figure 1 Scrum Workflow [11]

SCRUM VS WATERFALL – DEVELOPERS PRESPECTIVE

Scrum methodology is based on client collaboration, individual value and adaptation to change [11]. Than focusing only on completely defined rigid process models, it emphasizes on periodic interaction among stakeholders for success of process. Main advantages of scrum as compared with traditional process centric waterfall model are highlighted below.

Customer Interaction

In traditional development model the client has to give a detail of the software which has to be built. Whereas in agile the customer is involved in all phases of product development. In traditional model if something has to be corrected then a portion of software has to dismantled and then corrected since customer would not see the product until user acceptance test. In agile due to its incremental model customer feedback is valued at every phase. This not only reduces dependency but also increases customer satisfaction.

E. Cost of building

In traditional model if a critical defect is found at the end of development lifecycle there working of it takes lot of time and effort which is nothing but money. Whereas in agile a potential problem is detected very early and dealt accordingly. This reduces the cost by a great deal.

F. Flexibility

In traditional model the role of a business analyst and architect is clearly defined. The requirements are documented by the analyst; the development team enters only during coding. The analyst looks after entire development. Whereas in agile since the customer interaction is direct with the developers it is easy for the developer to understand what exactly the customer is looking at. This reduces intervention and time wasted on meetings, redesigning etc.

G. Size of the System

Agile is best suited to build small and medium sized products since they offer lot of compatibility. When it is a much bigger system once can follow the traditional software development which is much more structured. If agile is the choice then bigger system has to be divided into smaller component, and start from there.

Documentation is the only segment where traditional model wins over agile. In traditional model everything is documented in a sequence right from requirement analysis to release management. At every phase the documents are handed over to the next phase. Whereas in agile it is mainly developers who do this, so with frequent changes comments are updated frequently which at times is confusing to understand.

Major Advantages of scrum are summarized in figure 2.

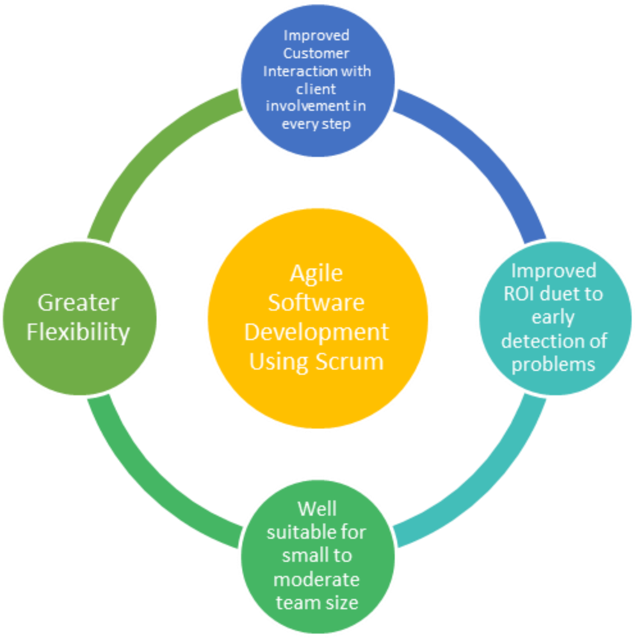

Improved Customer Interaction with client involvement in every step
Greater Flexibility
Agile Software Development Using Scrum
Improved ROI duet to early detection of problems
Well suitable for small to moderate team size

Figure 2 Advantages of Scrum

IV. Conclusion

Scrum is very effective in terms of resource utilization, cost and product feedback and development. It is the most effective model if the organization has an existing code pack which covers all modules of a product. This is most widely implemented in product companies, who have a base form of a solution working and they merely customize based on customer requirement. Scrum Works well even if requirements are not clear but its dynamic and rigorous processes tend to be complex and need highly expertise personnel to gets its true benefits. It sometimes highlights organizational issues which require careful dealing.

REFERENCES

[1] Bass Julian M, How product owner teams scale agile methods to large distributed enterprises, Empir Software Eng (2015) 20:1525–1557 , DOI 10.1007/s10664-014-9322-z.

[2] Bustard David, Wilkie George,Greer Des, Towards optimal software engineering:learning from agile practice, Innovations SystSoftwEng (2013), DOI 10.1007/s11334-013-0216-9

[3] Viviane Santos. Alfredo Goldman, Fostering effective inter-team knowledge sharing in agile software development, Empir Software Eng(2015) DOI 10.1007/s10664-014-9307-y

[4] ArdhenduMandal • S. C. Pal, Achieving mobility through BRIDGE process model : an approach to integrate the agile and disciplined software development, Empir Software Eng (2015) 20:1525–1557 , DOI 10.1007/s10664-014- 0239-x

[5] EstlerHans-Christian, NordioMartin, Furia Carlo A., Meyer Bertrand,Schneider Johannes, Agile vs. structured distributed software development:A case study, Empir Software Eng (2014) 19:1197–1224, DOI 10.1007/s10664-013-9271-y

[6] Petersen Kai , WohlinClae, The effect of moving from a plan-driven The effect of moving from a plan-driven approach with agile practices An industrial case study, Empir Software Eng (2010) 15:654–693, DOI 10.1007/s10664-010-9136-6

[7] PikkarainenMinna,SaloOuti, KuuselaRaija, AbrahamssonPekka, Strengths and barriers behind the successful agile deployment—insights from the three software intensive companies in Finland, Empir Software Eng (2012) 17:675–702, DOI 10.1007/s10664-011-9185-5

[8] M. Pikkarainen&J. Haikara&O. Salo, P. Abrahamsson&J. Still, The impact of agile practices on communication in software development, Empir Software Eng (2008) 13:303–337, DOI 10.1007/s10664-008-9065-9

[9] Rising Linda and Janoff Norman S., The scrum software development process for small Teams, 0740-7459/00/$10.00 © 2000 IEEE

[10] Malhotra Chhavi, Chug Anuradha, Agile Testing with Scrum-A Survey, IJARCSSE, Volume 3, Issue 3, March 2013 pp. 452-459

[11] Igor Ribeiro Lima, Tiago de Castro Freire, Heitor Augustus Xavier Costa, Adapting and Using Scrum in a Software Research and Development Laboratory, Revista de Sistemas de Informação da FSMA, n. 9 (2012) pp. 16-23

Lohit KS Lohit KS completed his bachelor of science in mathematics and computer science, currently pursuing masters in Christ University, working as Consultant at CSG International.

Rupali Wagh is an Associate Professor of Computer Science with Christ University. She has over a decade of experience in academics. Her research interests include data mining, text analytics and natural language processing and mobile technology.