



**ORIGINAL RESEARCH PAPER**

**Computer Science**

**AGILE TECHNOLOGY IN SOFTWARE DEVELOPMENT AND QUALITY ASSURANCE**

**KEY WORDS:** Agile, Scrum

**Sat Kumar**

Extension Lecturer of Computer Science Government College, Hansi (Hisar) Haryana, India

**ABSTRACT**

In this paper I have studied about modern software engineering concept and their strategy which is basically used for software development and constructs a model which measure the software quality of the development process. In Continuous improvement is a key factor for survival in today's turbulent business environment. This counts for civil engineering and even more so in the fast-paced world of information technology and software engineering. The agile methodologies, like scrum, have a dedicated step in the process which targets the improvement of the development process and software products. Crucial for process improvement is to gain information which enables you to assess the state of the process and its products. From the status information you can plan actions for improvement and also assess the success of those actions.

**1. INTRODUCTION**

Software engineering is defined as the basic the discipline of developing and software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. Today, software engineering is faced with rapid change in user requirements, technology. To cope with the rapid change in the business environment, the software development process must be assessed and adapted frequently. According to Fowler, the traditional methodologies or engineering methodologies, like the waterfall methodology, are not suitable for environments with rapid change, Engineering methods tend to try to plan out a large part of the software process in great detail for a long span of time, this works well until things change. So their nature is to resist change.

In Agile methodologies evolved to promote continuous improvement. Scrum belongs to the agile methodologies and defines frequent feedback loops for improvement. Although continuous improvement is a pivotal element of Scrum and other agile methodologies we are faced with the problem to identify potential for improvement. The Software development and metrics can be used to develop agile application technology. A variety of software metrics exist that provide information about resources, processes and products involved in software development. Software metrics provide factual and quantitative information.

**2. METHODOLOGY**

In this paper the methodology used for measure the agile development process and their parameters are theoretical basis for the definition of a measurement model. For this purpose, we study the literature of such papers and thesis databases. During the literature study we will also investigate how to choose and combine the metrics appropriately. The implications from agile development research questions help to ensure that measurement models deliver significant data and that the data can effectively be used by the stakeholders involved in the software measurement.

In this regarding the measurement goals in an agile software development process we will partly explore by means of the literature. The exploration tries to find out what industry experts define as the main goals in an agile software development process and their parameters which is used in research. This paper will investigate what are the specific needs in practice. The analysis of the business context will be used to find out the answers of agile manufacturing. We will analyze how exactly the development process is executed.

**3. AGILE METHODOLOGIES**

In Traditional and modern software development methodologies were deduced from engineering disciplines such as civil or mechanical engineering. But the adopted

concepts are not suitable for every software engineering project, because of the lack of edibility.

The agile engineering disciplines have a clear separation between design and construction. The separation can be made in construction because design is mainly an intellectual activity whereas the construction is foremost a physical activity. A comparable segregation is not possible in software engineering because a continuum exists between design and construction where both are thoroughly intellectual activities. In software engineering and civil both phases are executed sequentially which means that once the design phase is finished it is not reentered. Projects which are strictly executed according to such a sequential process are predictable. The traditional software development methodologies try to plan out a large part of the software process in great detail for a long span of time. The requirements are collected in the beginning of a project and then these requirement pass in one big bulk through all the steps in the development process but the same level of predictability.

The traditional software development methodology is the waterfall model. In this model UML diagrams are sometimes created as design documents. Although you can use peer review to check the design, errors in the design are often times only uncovered during coding and testing. Furthermore, requirements often change even late in a project. These circumstances show that a high risk exists of the design becoming obsolete. Once the design must re-engineered a lot of the invested time becomes wasted e ort. The high risk of wasted e ort constitutes one reason why a detailed planning is very inefficient.

**4. AGILE PARADIGM AND LIFE CYCLE**

In the agile paradigm, I have studied about this methodology and their development process but there is another reason for the inefficiency of a detailed planning. Jim Highsmith states that Projects may have a relatively clear mission, but the specific requirements can be volatile and evolving as customers and development teams alike explore the unknown. This fact adds to the risk of producing waste. Either time is wasted planning requirements or time is wasted on implementing requirements which the customer no longer needs. From the desire to cope with the unpredictability the agile software development methodologies emerged.

Agile methods and technology are adaptive rather than predictive as well as modern technology to develop software as quality. Agile methods are not only able to adapt to changes in the customer requirements but are also able to react to changes in the environment by adapting the software development process. Through these properties they are very effective in changing environments.

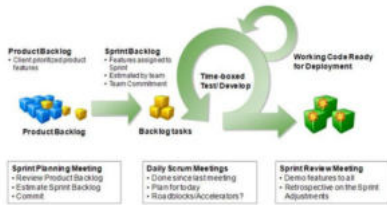


Figure-1: The Agile Scrum Process

**5. AGILE PRINCIPLES**

Agile practitioners keep in mind different ways or basic set of rules during any of agile methodology. According to following are the principles of agile development:

- Satisfy the customer through the early and quick delivery.
- Welcome change in requirements even in the late in the project.
- Keep delivery cycle short (weeks).
- Business and development people should work together.
- Build project around some motivated people.
- Place emphasis on face to face communication.
- Working software is primary measure of progress.
- Promote substantial development pace.
- Continuous attention to the good design and technical excellence

**6. SOFTWARE DEVELOPMENT METHODOLOGIES IN AGILE**

The different methodologies of agile development are given below:

- Scrum
- Extreme programming (Xp)
- Feature Driven Development (FDD)
- Crystal Clear Methodology (CC)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development

**7. THE SCRUM**

Scrum is an agile software development framework which also has the four principles of the agile manifesto at its core. In a scrum environment a team develops software in sprints. The length of a sprint is consistent but different teams may use different sprint lengths. Usually sprint lengths of 2 to 4 weeks are used.

A sprint is an iteration which consists of several steps as illustrated in Figure 2.1. At the beginning of each sprint stands the sprint planning. In the sprint planning meeting the development team meets with the product owner who is responsible for the business value of a project. Together with the product owner the development team chooses a set of work items from the product backlog to implement during the sprint. The product backlog contains all the work items for a specific software product ordered by their priority.

**8. PROBLEMS IN TRACKING SYSTEMS**

An Issue Tracking System (ITS) is a software tool which is used in the change management of software engineering projects. Issue tracking capabilities enable a project team to record and track the status of all outstanding issues. In the context of this master thesis an issue is either of the type enhancement, bug or task. In addition to issue recording and status tracking of issues, ITSs are used for project planning.

A user story describes a software enhancement from the user perspective. Usually, it contains the user requirements and the acceptance criteria which are used to evaluate whether an enhancement has been implemented according to the requirements. An issue of type bug is created if a deviation

**9. CHALLENGES IN SOFTWARE TESTING PROCESS IN AGILE**

Agile manifesto is the set of rules or principles for agile software development. These principles consist of the ideas that are basic guidelines and are common for all agile development methods. If we take a look at the software testing process in Agile methods.

First of all, the ultimate priority of agile development is to deliver a working piece of software to customers early and continuously with a rapid releasing cycle. For testing process it is a challenge because, if the release cycles are rapid then it will put fixed deadlines for testing activities and that does not allow maximizing the testing time if more defects are found than estimated. They cannot send them back to fix, because the testing time is pre planned and they need to deliver a product on a set date.

Secondly, Agile demands that changing requirements should be welcomed even in later stages of the development. Testing activities are traditionally been based on specifications that are completed in a phase of development and then they can be used as a basis for test design and other quality assurance activities. And if they will change the requirements and which will eventually change these documents then it will challenge the traditional way of doing testing.

**10. CONCLUSIONS & FUTURE SCOPES**

In this paper I, have to identify a structured approach which can be used to establish a software measurement program. The GQM approach proved to be suitable to design a software measurement program in this study. The pivotal factor is that the GQM approach follows a top-down fashion for the definition of a measurement model. The construction of a measurement model starts with the identification of the business goals. The business goals are then linked to measurement goals. From the measurement goals we derive questions which need to be answered to assess the development performance against the measurement goals.

**REFERENCES:**

1. Ahsan Nawaz and Kashif Masood Malik, Software testing in Agile Development, A Thesis in Computer Science, MCA-2008-25 (2008)
2. Falk Martin Software Quality Assessment in an Agile Environment, MCS (2012)
3. ACM. Computing degrees & careers. [http://computingcareers.acm.org/?page\\_id=12](http://computingcareers.acm.org/?page_id=12), 2006. Retrieved: 16 March 2013.
4. Agile Alliance. Guide to agile practices. <http://guide.agilealliance.org/>.
5. Scrum Alliance. Scrum's three roles. [http://www.scrumalliance.org/pages/scrums\\_roles](http://www.scrumalliance.org/pages/scrums_roles). Retrieved: 11 October 2012.
6. T L Alves, C Ypma, and J Visser. Deriving metric thresholds from benchmark data, 2010.
7. D. J. Anderson. Agile management for software engineering: applying the theory of constraints for business results. Prentice Hall, 2004.