A STUDY ON AGILE SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

Manisha Wadhwa
India

ABSTRACT
This paper describes about the Agile software development life cycle. It also focuses on main features of an Agile software development life cycle model and various common features of Agile frameworks and methodologies.

Key words: Cross Functional Teams, Scrum, Kanban

INTRODUCTION

Agile software development describes a set of values and principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods. The term agile (sometimes written Agile) was popularized by the Manifesto for agile software development, which defines those values and principles. Agile software development frameworks continue to evolve, two of the most widely used being Scrum and Kanban. The Agile software development life cycle is based upon the iterative and incremental process models, and focuses upon adaptability to changing product requirements and enhancing customer satisfaction through rapid delivery of working product features and client participation. Agile methods primarily focus upon breaking up the entire product into smaller, easily developable, “shippable” product features developed through “incremental” cycles known as “sprints”. What is Agile software development life cycle? Detailed explanations are available which explain how the Agile cycle works. However, to understand the entire cycle in a nutshell, each Agile sprint traditionally lasted from two weeks up to one month. Agile trends now indicate they typically last from seven days up to ten “working” days. Cross-functional teams work simultaneously while developing the product features in daily sprints. The team members are generally experienced and possess varied levels of expertise in activities such as designing, coding, testing, and quality acceptance. At the end of each sprint, a working product feature(s) is developed and presented to the product owner for verification purposes. Once the PO Okays the development, it is presented to the stakeholders, and their opinions are carefully noted to improve upon the current product development cycle. The entire process is repeated through sprints until all the constituent product features are developed.
MAIN FEATURES OF AN AGILE SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

Agile frameworks and methodologies have some common features.

INDIVIDUALS AND INTERACTIONS
In the Agile software development life cycle model, self-organization and motivation takes precedence over delegation of authority and following the “seniority” hierarchy. Team members are encouraged to take an active part in the development and planning activities. They are also “empowered” to take certain decisions on their own. The Agile team has to collaborate and share ideas to develop the product “as a whole” unit i.e. each member should support a common vision.

WORKING SOFTWARE
Agile concentrates upon delivering sustained “working” product releases through product incremental cycles over documentation and working protocols. The main objective is to develop, and deliver, bug free product feature releases in a continuous and sustained manner until the entire product is developed.

CUSTOMER COLLABORATION
Since all the requirements pertaining to product development may not be available, or “acquirable”, at the project start up time owing to various factors, development should commence almost “immediately”, and presented to clients for verification purposes. Stakeholders and project owners “clear” the product features developed through the sprint cycles. A lot of time is saved through customer collaboration, and as a result, the project proceeds in a successful manner as the client always Okays the development keeping in mind the current market trends.
RESPONDING TO CHANGES
Agile focuses upon incorporating dynamic changes in the product development cycle. Changes in the product features can be easily and effortlessly carried out by developing “user stories” – product functionality or features as defined in the product backlog. Changes can be carried out at any time while the features are being developed – even late in the product development cycle.

AGILE SOFTWARE LIFE CYCLE BASICS

An Agile software life cycle is much different as compared to traditional software development frameworks like Waterfall. In Agile, more emphasis is given to sustained and quick development of product features rather than spending more time during the initial project planning, and analyzing the actual requirements. The Agile team develops the product through a series of iterative cycles known as sprints. Besides development activity, other aspects pertaining to development such as product analysis, designing the product features, developing the functionality, and “testing” the development for bugs are also carried out during the sprints. The incremental cycles should always produce a “shippable” product release that can be readily deployed. Agile processes make extensive use of events such as the daily scrum meetings, sprint review meeting, and the sprint retrospective meeting to identify and self-correct the development carried out by the team. Feedback is solicited frequently, as and when needed, to collaborate, and speed up the development process through sharing of ideas and self-management. The feedback system helps to support the self-correction features of Agile frameworks, and is very important. The roles played in the Agile process constitute of the product owner, scrum master, and the development team. The product owner “owns” the project on behalf of the stakeholders and
ensures that the entire project is developed successfully keeping in mind the stakeholders vision of the product as it should “appear” in the market. The scrum master ensures that the Agile process is followed at all times, and does his or her best to resolve any difficulties or technical issues arising during the development process. The team members participate actively in the daily sprints and make sure meaningful and useful development of product features is presented at all times.

CONCLUSION
Agile software development methods support a broad range of the software development life cycle. Some focus on the practices (e.g., XP, pragmatic programming, agile modeling), while some focus on managing the flow of work (e.g., Scrum, Kanban). Some support activities for requirements specification and development (e.g., FDD), while some seek to cover the full development life cycle (e.g., DSDM, RUP).

REFERENCES