Requirements engineering in traditional and agile contexts

The continuous evolution of the business environment in most of the organization has constantly questioned the traditional approaches of requirement engineering. The software development processes followed in various organization has to deal with rapidly changing requirement during a project to coupe up with the customer preferences, developing technology and competitors in the market (Cao.L and Ramesh.B, 2008).

Before dwelling into the difference between the requirement engineering approaches, let us briefly understand what requirement engineering is and why it is necessary. Requirements engineering is about understanding why a software system/solutions is being developed. It mainly involves understanding the **current situation**, **customer/user needs** and transforming the needs into requirements (Marjo Kauppinen, 2014).

In the traditional approach, the requirement engineering was the initial phase of the project and used to start much before the actual project started. The process used to begin with developing a description of the problem or sub problem that needed to be solved. Resist all efforts to expand or modify the problem or in the software domain more commonly known as '**freeze the requirement**' (Jeff Conklin, 2010).Once the requirements are agreed upon by both the stakeholders, a document is prepared based on the requirements and signed by either parties. After the agreement there is minimal scope to change the requirements and will lead to '**analysis paralysis**' (Jeff Conklin, 2010). The requirement engineering phase used to eat huge time out of the project as it involved **detailed documentation** and coming to a mutual agreement on the system specifications and function among the stake holder. This will result in lead time of the project (Sommerville, 2011).

The Agile method challenges the creation of requirement document is a waste of time as the requirements are rapidly evolving and the customer is unsure of the needs until he sees the system in action.**Extream programming** uses incremental requirements engineering and also expresses requirements as **user stories**. System developed using this method will have less detail in the requirement document (Sommerville, 2011).The requirement change in Agile can be handled better than the traditional waterfall approach. As there is continuous observation and analysis of system development process by the customers, the can suggest changes in the requirements after each iteration thus increasing the customer satisfaction about the product (Sommerville, 2011). The sprints and increments of requirement engineering phase were much shorter compared to waterfall model (Marjo Kauppinen, 2014)

Marjo Kauppinen believes that there is no much difference in the requirements engineering processes in the traditional and modern approaches. She believes that requirement engineering should be an ongoing process throughout the project to improve the system and deliver the required solutions. **Prototyping** is one of the key feature of agile method which will help analyse the critical and hidden requirements of the users (Marjo Kauppinen, 2014). In agile performing requirement change management **cost-effectively** is a challenge. Product backlog lists all the requirement of individual feature but it does not give an overview of methods and modelling of the requirement .This is a drawback of agile approach to requirement engineering. Scrum hides the complexity of the requirement engineering process but it's the backbone of the process (Marjo Kauppinen, 2014).

In conclusion to deliver a stable system or software there is a need for systematic and manageable approach to Requirement engineering and this form a key area of the project. Good requirement engineering principles will result in great systems and solutions.

References

Sommerville. I. (2011) Software Engineering, 9th edition, U.S.A: Pearson Education, Inc.

Podcast interview between Prof. Marjo Kauppinen and Prof. Casper Lassenius, 2014

Cao, L. and Ramesh, B. (2008). 'Agile Requirements Engineering Practices: An Empirical Study', IEEE Software, 25 (1), pp. 60-67.

Conklin, J. 2010, 'Wicked Problems and Social Complexity'.