

REQUIREMENTS ENGINEERING

Requirements for a software system set out what the system should do and define constraints on its operation and implementation (Sommerville, 2016). The requirements engineering process mainly includes four steps: **requirements elicitation**, **requirements specifications**, **requirements validation** and **requirements management** (Sommerville, 2016). RE is difficult due to ever changing requirements, different viewpoint of stakeholders, no standard process etc. The evolving requirements make traditional (like waterfall) requirements engineering process different from that for agile development.

Requirements engineering process for traditional development:

In traditional development, requirements engineering is done only at the beginning of the project. Only after all functional and quality requirements are validated, can we proceed to the next step. This method requires requirement specification to be very detailed because there is a very limited scope of changing requirements at a later stage.

It is efficient when requirements are completely known at the beginning and are not going to change after the development process has started. For example, in hardware development where changing the requirements at a later stage can be very expensive.

Requirements engineering process for agile development:

In agile development, requirements engineering is an ongoing activity. That is, the requirements keep evolving with every step (say every sprint) until the project is finished and/or stakeholders are satisfied (Kauppinen, 2013). Requirements are prioritized before each iteration and customer feedback is taken after the iteration to see if that is what they wanted. According to Cao & Ramesh (2008), informal communication in agile development discards the need for extensive documentation and hierarchical approval procedure. The most common method to represent requirements is user-stories and use-cases. Main focus is on identifying critical requirements using prototypes, which is a key activity of agile methods, and customer collaboration (Kauppinen, 2013).

This method is efficient when requirements are not precisely known at the beginning and are likely to change with time.

Other differences:

In traditional development, many factors affect requirements prioritization like business value, risks, cost, implementation dependency, etc. While in agile development, requirements prioritization depends only upon business value defined by the customer. (Cao & Ramesh, 2008)

While reprioritization can be difficult and expensive in traditional development, there are many opportunities for reprioritization in agile development. Hence, cost of incorporating a change in requirements is far lower in agile as compared to traditional development.

Since customer is more involved in agile methods, requirement validation can be very accurate. While in traditional development, there is a limited scope of customer involvement after the development has started. So requirement validation might not be satisfactory because the customer may not know what s/he precisely wants in the beginning.

References

SOMMERVILLE, I., 2016. Software Engineering. 10th ed. Boston: Pearson.

Kauppinen, M., 2013. Requirements Engineering [podcast]. 6 February 2013. [viewed 15 February 2020]. Available from:
https://mycourses.aalto.fi/pluginfile.php/1177198/mod_resource/content/11/Kauppinen.mp3

Cao, L. & Ramesh, B., 2008. Agile Requirements Engineering Practices: An Empirical Study [online].[viewed 15 February 2020]. Available from:
https://mycourses.aalto.fi/pluginfile.php/1177197/mod_resource/content/11/AgileRE.pdf