Agile Requirements Engineering Practices: An Empirical Study

Bibliographic data

The paper was published in 2008 by the IEEE Computer Society and it was written by Lan Cao and Balasubramanian Ramesh. It goes by the name "Agile Requirements Engineering Practices: An Empirical Study".

Theme of the paper

The paper approaches the theme of **Requirements Engineering** and **Agile Methods** application in requirements engineering.

The main focus is to reach a conclusion on which requirements engineering practices agile developers actually follow and the benefits and challenges presented by those practices.

Synthesis of the paper

A rapidly changing business environment contributes to **ever-changing and evolving requirements**, which challenges traditional requirements engineering approaches. As of now, pre-specified requirements are inappropriate.

In this context, arises the proposal of the application of agile methods to requirements engineering which resides in the development of code without waiting for formal requirements analysis and design phases as requirements emerge throughout the development process based on constant feedback from stakeholders.

There have been identified seven agile Requirements Engineering practices: **Face-to-face communication over written specifications**, which focuses on face-to-face communication as the primary source of requirements over overly formal written specifications; **Iterative requirements engineering**, where requirements are not predefined. High-level requirements are understood in the beginning and every new cycle the customer gathers with the development team to provide additional detailed information on features to be implemented; **Requirement prioritization goes extreme**, in which requirements are implemented according to prioritization agreed with the customer. Prioritization is based on what will deliver most business value to the customer and requirements are prioritized each planning meeting; **Managing requirements change through constant planning** allows for accommodating requirements changes during development to better satisfy customer needs. Changes become easier and less costly to implement; In **Prototyping**, there is the development of an operational prototype from an initial set of prioritized features to be optimized through several iterations considering feedback from the users who test it; **Test-driven development** is an evolutionary approach where tests are created before writing new functional code; At last, **Use review meetings and acceptance tests** where there are frequent review meetings for requirements validation where developers demonstrate the delivered features to receive feedback and acceptance tests used for requirements verification and validation.

Amongst all of these practices, a few of the benefits appear to be in line with it being easier to **deliver business value to please the customer**, reduced time-consuming documentation, clearer requirements, **minimized need for changes and the cost of those that are needed** and there is quicker customer feedback. On the other hand, some challenges are the fact that effectiveness of communication is dependent on many factors, negotiation and reaching an agreement between customer groups is difficult, projects' scopes are often subject to many changes, there can be communication breakdowns, neglect of non-functional requirements and instability.

In the conducted study, most organizations practiced most of the aforementioned practices but not all of them struggled with the mentioned challenges of agile RE practices. The most common challenge is the inability to gain access to the customer and achieve an agreement between various customer groups. Test-driven development was the least used practice as it requires more discipline than most developers are used to and, surprisingly, almost one-third of the organizations do not practice prototyping. In contrast with literature findings on unpaid attention to reviews and tests in traditional development, these organizations use them extensively.

In conclusion, the main difference between traditional and agile RE resides in the fact that agile takes an **iterative approach** since it occurs in an environment where developing unambiguous and complete requirement specifications is too difficult or does not even make sense. The core practice in agile is **intensive communication** between developers and customers and it is important to note that **processes** are not centralized in one phase before development but **evenly spread all throughout development**. Although it allows for many benefits, it proves to pose some challenges as well, which is why the study suggests that agile RE practices are "neither panacea nor poison" to the challenges intrinsic to RE, i.e., agile RE practices are not the magical solution that will fix all problems in requirements engineering but a method that should be considered after weighing its benefits and costs for a particular project environment.

Questions and reflection

Q1: It is stated that the choice to adopt agile RE practices should come after careful consideration given the costs and benefits for a particular project environment. Is it feasible to have a company/team choose different methods every time they take on a new project? How can we train workers to adapt to all of these different methodologies from project to project? Will there not be more time invested in learning and adapting to a new methodology than on actual development?

Q2: How can a team truly choose and commit to a methodology in the beginning of a project? Do they have enough information from the start of the project to compare the benefits and costs, particularly when we are dealing with customers who are not used to those methods? For example, a customer who promises to be around and available for constant feedback but, because they are not familiar with the method, actually does not truly understand how much availability would actually be required of him.

After reading this paper, I believe I have a more consolidated view on agile RE practices. I already knew that there were many benefits to it but was not as aware of the deriving challenges. That being said, I feel like most challenges were identified in theory but not a significant issue for the companies implementing the methods.

From personal experience, I have always found that the biggest issue for me was maintaining formal specifications throughout the iterations as the requirements were always changing, but now realise that the level of formal specification that was required in our previous classes in college is not in complete line with the actual methods as they were intended or as they are actually applied in the real world context. I feel like there is a disparity between what is asked of us at school and what we should actually be doing in order to fully take advantage of the benefits of this practices.

As for the actual challenges presented in the paper, I think that the main one, since it is the hardest to control as it is not solely dependent on train and discipline, is the difficulty in communication, which includes communication between the customer and the developing team and the process of negotiation to reach an agreement between different customer groups. Since these are dependent on the human factor, they are more difficult to control or predict.

As for my own questions, I do not have an answer to them. I think the method should always be in line with the project but I also do not think it is always feasible. Maybe sometimes it makes more sense to "compromise" and use a method that is not as adequate but in which the team is better trained if that one project is the exception in the line of all the other projects they work in. Ideally, all professionals should be trained in college or receive seminars in their workplace considering the existing methods so it is easier to adapt once they do have to adopt them.

Maria Eduarda Santos Cunha
Up201506524
20.03.2019