Agile Software Development – Assignment One (Report)

The aim of this report is to recommend an agile methodology to the client, CGHM, an e-commerce start-up which wishes to rival Amazon. The client has conducted research and is seeking a recommendation between either the Extreme Programming or Scrum development methodology. In order to conduct this recommendation, this report will first discuss the goals of Agile software development, then the core values and goals for Extreme Programming and Scrum. The structures of both staff and the project will then be analysed. This report will then conclude with a recommendation for CGHM.

The Agile software development philosophy was developed to create a new software development paradigm. Beginning with the code-and-fix method, traditional software development methodologies were monolithic in nature, focusing on a series of concrete development stages that were often difficult or impossible to adapt to customer, business and market changes [1]. This led to expensive, delayed or even cancelled software projects [1]. The Agile philosophy was born in a revolutionary way through the publication of a manifesto in 2001 in order to overcome the aforementioned difficulties [2]. Through four core values, the manifesto prioritised collaborative and face to face interactions over official processes, producing functional software over the publication of documentation, customer interaction over business practices and finally adaptability over planning [3]. These were accompanied by an additional 12 principles which further distanced itself from the accepted yet criticised models, notably the heavily planned and document heavy Waterfall method [1], [4]. Thus, different models in line with the Agile philosophy, such as Scrum and Extreme Programming (XP) must be considered.

In order to evaluate the best suited methodology, the core values of each must be analysed. Although published prior to the Agile Manifesto, these methodologies are in line with its core values, namely promoting communication, simplicity, feedback and courage for XP and transparency, inspection and adaptation for Scrum [3], [5]. XP is further developed with 12 distinct practices [3]. One could argue a large proportion of the philosophy of XP can be seen in the practice of Pair Programming. In theory this would promote not only good quality code, the sharing of ideas and implementations but also allow enable the structure of the project to be understood throughout the team [3]. However, this process can be monetarily and time consuming, strongly negative risks for a start-up like CGHM.

On the other hand, Scrum intends to strive in complex situations [6]. This is primarily done through a set of rigid rules and time frames that will be further discussed later in this report. CGHM could likely have a sense of urgency to get the project completed in order to gain revenue and continue developing the business. Additionally, the team is said to be talented, thus investments in time for training and to create quality code may not be necessary at this stage. Furthermore, the differences between XP and Scrum can be seen in practice.

The individuality of each methodology is visible in their implementation. Both methodologies revolve around a similar process, that of dividing the work into items described in plain English known as stories [7]. These stories are then assigned to a development cycle called an iteration in XP which last one to two weeks and are part of a larger release every three months [7]. A release is a fully functional deliverable that is ready to be integrated into the system should the client choose to do so [7]. Scrum works in 30 day Sprints, working through a product backlog in terms of most important to the client [6]. Once more, XP demonstrates its focus on quality of output whereas Scrum focuses on delivering results within a time frame, a difference that will remain throughout this report. The differences in philosophy continue in the staff structure.

Each methodology represents the client and the development teams. In both methodologies, there should be a client on-site or available to discuss any issues that arise [5], [7]. In XP, the Client Team can be made up of a single person or group of people who are the business and client point of contact for the developers and select tests to accept the team’s release [3]. In contrast, Scrum represents the Client on a day to day basis through the Product Owner who is the business point of contact. They are in charge of information gathering, working with the client, obtaining investment as well as assigning the stories for the upcoming Sprint [6]. A Sprint is concluded with a review of the code and the planning of the next Sprint with the client [6]. The customer is heavily involved in both cases and is often encouraged to provide feedback and be part of the development process as much as possible.

On the other hand, the development teams have one key difference. For both Scrum and XP, the development teams are self-managing, in charge of their own workload and producing the iteration [3], [5]. However, a key difference is the presence of a ScrumMaster in Scrum. The ScrumMaster acts as a team manager, tasked to ensure that the project is on track, identify issues and uphold the three pillars of Scrum [6]. This brings focus and drive to the development team as they are held accountable not only to each other but also to the ScrumMaster [5]. As previously mentioned, CGHM requires the delivery of a product in order to begin their operation. Although XP does promote more direct interactions with the client, the role of the ScrumMaster is to keep the team focused and to bring deliverables to the client. Furthermore, as the client is also the team's employer, this will limit difficulties in communication between the development team, the Project Owner and the client. The impact of these roles becomes apparent in the iteration process.

The process for each iteration demonstrates some of the key differences between XP and Scrum. Scrum takes its name from Rugby as it can be likened to the scrum manoeuvre, a team of developers working closely together with a common goal in a high pressure environment [8]. Both methodologies promote daily activity meetings, although XP only requires regular structured meetings [3]. This flexibility spreads throughout XP, expecting changes and adaptations to be made due to environment or work constraints [3]. In contrast, Scrum requires daily meetings of a maximum of fifteen minutes during which each member answers three questions: What have you done since the last meeting? What do you plan on doing today? What difficulties are you experiencing? [6]. This rigidity can be seen throughout the Scrum process with time limits on every meeting and process [6]. However, during the maximum three hour time Sprint retrospective, the team discusses their experiences and improvements for the upcoming sprint [6]. Additionally, the contents of the Sprint cannot be altered by forces outside of the team but will allow developers to reduce the number of items in the sprint in order to reach the deadline [9]. This illustrates the ability for some flexibility within Scrum. This rigidity can be difficult, especially if the development team have not yet experienced Scrum.

XP focuses on the quality of the output. Many of the practices promoted by XP such as testing, refactoring and coding standards promote a focus on ensuring high quality code [9], exemplified with the practice of creating the unit test prior to starting development on the story. This ensures that the developer's work is in fact correct and aligns with the customer's expectations [9]. This focus on testing permeates throughout XP. Every time work is completed and added to the iteration or release, every test is run and must pass [9]. In contrast, Scrum emphasises testing during the Sprint Review, but is also done iteratively throughout the process [6]. The flexibility of XP is advantageous and the development team to create their own best practices. Nonetheless, one could argue the amount of time to adapt XP to the development team can take as long if not longer than for the team to adapt to Scrum.

In this case, CGHM is an e-commerce start-up, looking to rival the likes of Amazon. With a talented team of developers, it is unlikely that they will require much training and will also be likely to produce good quality code from the outset. As previously stated, this start-up will likely prioritise generating revenue in order to commence and continue their operations. Furthermore, CGHM is the developer's employer, thus it can be assumed that communication between developers and business may not be an issue. Thus, this report will recommend the Scrum methodology to CGHM due to its fast-paced, efficient and rigid methodology. Although XP ensures better quality code as well as more flexibility in its implementation, it is also more expensive and time consuming than Scrum. This report would recommend an eventual switch to Scrum when the development team grows and is able to accommodate developers who require training. Of course, the company must also be in a financial position to invest in adapting a new developing methodology.

References:

[1] S. Misra, V. Kumar, U. Kumar, K. Fantazy, M. Akhter, “Agile Software Development Practices: Evolution, Principles and Criticisms,” *International Journal of Quality & Reliability Management*, vol. 29, no. 9, pp. 972-980, 2012.

[2] T. Dingsøyr, S. Nerur, V. Balijepally, N. B. Moe, “A Decade of Agile Methodologies: Towards Explaining Agile Software Development,” *The Journal of Systems and Software*, vol. 85, pp. 1213-1221, Feb. 2012.

[3] L. Lindstrom, R. Jefferies, “Extreme Programming and Agile Software Development Methodologies,” *Information Systems Management*, vol. 21, no. 3, pp. 41-52, Summer 2004.

[4] W. W. Royce, “Making the Development of Large Software Systems,” *Proceedings of IEEE WESCON*, vol. 26, pp. 1-9, Aug. 1970.

[5] J. Cho, “Issues and Challenges of Agile Software Development with Scrum,” *Issues in Information Systems*, vol. 9, no. 2, pp. 199-195, 2008.

[6] K. Schwaber, *Agile Project Management With Scrum*, Redmond: Microsoft Press, 2004.

[7] L. Layman, L. Williams, D. Damian, H. Bures, “Essential Communication Practices for Extreme Programming in a Global Software development team,” *Information and Software Technology*, vol. 48, pp. 781-794, Mar. 2006.

[8] L. Rising, N.S. Janoff, “The Scrum Software Development Process for Small Teams,” *IEEE Software*, vol. 17, no. 4, pp. 26-32, Jul. 2000.

[9] K. Beck, “Embracing Change with Extreme Programming,” *Computer*, vol. 32, no. 10, pp. 70-77, Oct. 1999.