

Individual Report:

Agile Approach as a Way to Promote Innovativeness in Software Development: A Literature Review

Class: ETM 649 Management of Technology Innovation Term: Spring-2017

Instructor: Dr. Charles Weber

Student: Husam Barham

ETM OFFICE USE ONLY Report No.: Type: Note:

Student Project

Class: ETM 649 Term: Spring-2017 Husam Barham: Individual Report. Agile Methodology as an approach to Promote Innovativeness in Software Development: A Literature Review

Outline

I. Introduction	
A. Research Question	4
II. Methodology	4
III. Literature Review	4
A. Agile Software Development Approach	4
Agile Manifesto	5
B. Agile Methodologies	6
Scrum Framework	7
C. Agile Benefits	8
D. Agile Criticisms	9
E. Case Studies	
Netflix	
Guardian	
VI. Discussion	11
V. Conclusions	
A. Recommendations	
References	

List of Figures

Figure 1: Evolutionary map of agile methods [10]	5
Figure 2: Scrum Process Framework	8

List of Tables

None

Abstract: This paper is the individual research paper for ETM 549/649 class for the spring-2017 quarter. The topic I have chosen is: "An innovation concept that has not been discussed in class", specifically, Agile approach in software development.

Agile approach and methodologies spread quickly across the software development industry in the last two decades, and in fact, it is starting to gain contraction in other industries as well. As, it held the promises of flexibility, user value focus, quality of products, faster time to market, and increased innovativeness in building software systems.

In this report, a literature review of agile was conducted; to find out how it is different from the classical waterfall approach and what shortcomings in that model agile addressed successfully that makes it so popular. The report also conducted a discussion of agile under a famous theory about how innovation is being created in organizations. Recommendations were offered at the end of the report.

I. Introduction

Software development projects are frequently challenged with scope and plan changes while the project being executed. Two of the main reasons causing this phenomena are: lack of clear details at the beginning of the project, and the rapid changes in software technologies that causes some initial technology choices to became obsolete by the time the project is done [1] [2].

Traditional Waterfall approach to planning software development projects, which depends on heavy upfront planning and rigid attitude towards changes, causes software systems to be developed with the goal of conforming to the initial plans, rather than being developed to add value to clients, causing many projects to fail [2][3].

In response to this challenge, agile software approach emerged, as a new approach to software development, with focus on software being developed to generate value to customers rather than to satisfy plans. As well as, to embrace interactions between competent individuals as a way to develop innovative software systems instead of following processes and tools [4] [5].

-3-

A. Research Question

Many researchers and practitioners view agile as a revolutionary innovative way to build software systems that offer an environment in which software engineers can be creative while having the flexibility to accurately address client needs [1] [2] [4] [5] [6].

But, does agile really add innovations to the way software engineers are doing their work, and what are the benefits and downsides of this methodology?

II. Methodology

To answer the research question, I'll conduct literature review in which, I'll go through agile methodology definition, various implementations, its benefits, and downsides. Leading to conduct a discussion about agile approach as a way to promote innovativeness, in comparison with Brown and Duguid argument about the tension between creativity and structure [7] [8] [9]. Brown and Duguid researched the work environment mechanisms to promote innovativeness and discuss the differences between process vs. practice and routine vs. innovativeness and its role in generating knowledge and innovations vs. commercialization of those innovations.

III. Literature Review

A. Agile Software Development Approach

As pointed out in the introduction, agile approach evolved as a response to the shortcomings of the Waterfall methodology [2][3]. Figure 1 shows the evolutionary path that led to the rise of agile software approach.

The aim of agile development approach is to have faster and nimbler software development processes. Under agile, software is being developed in an iterative, and incremental way, based on a team cooperative and straightforward efforts that are adaptive to generating value to customers. Class: ETM 649 Term: Spring-2017 Husam Barham: Individual Report. Agile Methodology as an approach to Promote Innovativeness in Software Development: A Literature Review

Within this approach, iterations represent small software releases that are time-boxed (every event has a maximum duration) that, incrementally, result in the final software system. The iterations approach allowed for immediate feedback and flexibility in responding to client's needs [10][11] [12][13].



Figure 1: Evolutionary map of agile methods [10]

Agile Manifesto

Agile practices emerged in the late 1990s and early 2000s [11]. And in February 2001, a group of, self-called, independent software development thinkers met in Utah and discussed the key principles that should govern Agile. They agreed that the principles should steer away from "fixed" process mindset and move toward more focus on value and culture; to revolutionize software development and promote creativity and achievement [14]. The meeting resulted in a symbolic manifesto called the Agile Manifesto. It included the following statements [5]:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: Individuals and interactions over processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation. Responding to change over following a plan. That is, while there is value in the items on the right, we value the items on the left more."

The manifesto was elaborated further by twelve principles that focused on generating value to customer, being flexible in accepting changes, delivering software frequently and on regular bases, and creating an environment of trust by building projects around motivated individual while offering them support and trust. It also, promoted building simple solutions while offering technical excellence and good design. And stressed on direct interactions between developers and business people and having self-organized teams with internal mechanisms for tuning and adjustment in team behavior and performance [15] [12][6].

B. Agile Methodologies

Agile is a theoretical framework that doesn't specify certain processes or practices to follow. Those are defined as part of agile methodologies, which are specific frameworks that extend and explain what processes and practices to follow to be agile [11][13][16]. Following are some of the most famous agile methodologies:

- Scrum [17]
- Dynamic System Development Methodology (DSDM) [18]
- Extreme Programming (XP) [19][20]
- Test-driven Development (TDD) [21]
- Lean [22]
- Kanban [23]

Scrum is one of the most famous and used agile methodologies [11]. In fact, one study suggested that at least 80% of the firms, that are agile, use scrum [24]. Following is a review of this methodology, to show how agile is being implemented and how does it add value.

Scrum Framework

Scrum is an agile software development methodology that handle new software system development with high-performance small teams working in an interdependence and intensive manner. It requires well-trained team members that are capable of selfmanagement, communication, and making decisions in real time based on the current situation [17] [11]. The word scrum comes from the scrum formation in rugby, which is used to restart the game after an even that causes a play to stop [11].

Following is a summary of how software is being developed in scrum [12][25][26]: First, workload is broken-down into user stories that explain the product features needed in the system. Each story is written in the form of a user narrative of how they will use this feature. The story contains user acceptance criteria and non-functional requirements.

Stories will be implemented by small teams, each team consists of a ScrumMaster, a person that organize, priories, coordinate, and facilitate the team work. A Product Owner, a person that represents end users, and make sure the work being done is delivering the required value to end users. And the team, a group of self-organized and cross-functional professionals capable of doing the required work without external help.

The work is being delivered iteratively using time-boxed events called sprints that have a pre-assigned unified timeframe. In each sprint, an incremental part of the work, based on user stories, will be carried out. Within the cycle of the sprint several activities are being executed. Including an assessment of what is being done so far, what are the issues facing the team, and what is the end users feedback on previous sprints deliverables. So, a continuous

improvement and in real-time feedback is being done as part of this methodology.

See Figure 2 for more details.



Figure 2: Scrum Process Framework [12]

C. Agile Benefits

Agile approach offers several advantages against the traditional waterfall approach [10] [13][11][12]:

First, faster product delivery to market: work is done in iterations, or sprints. Each iteration should result in a working part of the system being delivered to the customer.

Also, by doing incremental iterations and getting customer feedback immediately, it reduces uncertainty and risk; as customers can see and use the work done so far immediately and hence assess how properly the work is aligned with their expectations and needs, and ask for immediate adjustments and tweaks. Moreover, agile increases customer satisfaction by focusing on customer value instead of a pre-defined scope. Also, customer satisfaction is gained by being flexible in accepting customer changes as soon as the customer asked for them, by adding them to current iteration or to the next one.

Furthermore, agile increases efficiency as having a cross-functional team members working together and with customers reduces reworks and miscommunications. Also, increase team engagement, as they can see their work being used immediately.

Finally, agile increases the quality of software systems; since there are frequent deliverables, which allows for better testing and reviewing in each sprint, and finding defects as soon as possible and before they can cause much damage.

D. Agile Criticisms

time.

Agile approach faced several criticisms [10] [13][11][12]:

First, agile depends on small self-governing teams of professionals working closely with each other. So, for effective agile implementation, the majority of team members should be highly experienced and trained, and that is costly and can't always be achievable. Also, to be able to effectively deliver workable components on timely manner, the team must be fully dedicated, and in many firms, programmers do work on several projects in the same

Furthermore, agile manifesto favors working software over documentation, and that led to a misconception among many agile practitioners that there is no need for documentation at all! Moreover, agile approach offers scope flexibility, so; under agile approach, teams start with the core features of the system and incrementally add features based on feedback from clients, and that could lead to a final system that is totally different than what was originally expected.

-9-

Finally, since the scope is flexible and the system is being developed in an incremental manner, it is difficult to plan for time and cost properly in advance.

E. Case Studies

Netflix

Netflix has been using agile in developing its services ever since it started in the early 2000s [27]. The company releases new updates to its services every two weeks. Allowing them to be adaptive and flexible as they are building for the present all the time. The agile fast incremental iterations approach enabled Netflix to be successful by allowing them to fail fast, have the chance to do more experimentation, and learn quickly. Which enabled Netflix to offer more new creative features and get feedback immediately to advance or discard those ideas, effectively creating a safe and risk reduced environment for creativity while maintaining stability and quality of service. Netflix mantra is "Release early, release often" [27].

Guardian

Guardian.co.uk is one of the most popular newspaper websites in the UK with more than 700 pages view per second [28]. They tried to upgrade their massive legacy content management system using traditional Waterfall approach unsuccessfully. Then they applied agile approach, and start to upgrade their content management system and related online services section by section. The whole system was upgraded after 22 releases, with each release containing enhancements based on feedback and evaluation of previous releases. At 70% of the way, the project team started to add new features that adds value to the website, they were absorbing growth in scope while maintaining their time schedule. The project was done on time with far more features and enhancements than what was originally expected [21]. Class: ETM 649 Term: Spring-2017 Husam Barham: Individual Report. Agile Methodology as an approach to Promote Innovativeness in Software Development: A Literature Review

In summary, the common factor between those cases is how agile allowed them to be more innovative, by allowing them to have scope flexibility with focus on generating value to customer instead of satisfying initial requirements. The time-boxed iterations approach, allowed them to do so, while significantly reducing time-to-market and reduced risk of defects due to continuous improvement and in real-time feedback allowed by the short iterations approach.

VI. Discussion

In the section, agile approach as a source to promote innovativeness in software development is discussed. It is juxtaposed with Brown and Duguid argument about the tension between creativity and structure [7].

Broun and Duguid argued that knowledge creation and wealth creation don't necessarily move hand in hand. They indicated that small groups of practice, which are not governed in rigid structure, could be creative and spark knowledge creation and innovations. While wealth comes from growth that needs a process, specialized subgroups that are coordinated by an "orchestra" conductor, and the best firms are the ones who can manage both forces instead of being in favor of on of them, to maintain creativity while making money.

While this argument is sound and logical, it seems that applying agile in the software development industry does not quite fit under it; agile promotes an incremental time-boxed approach that result in the ability to deliver an initial working software and keep adding/ testing new features to the system while it is being used by the customers. So, innovativeness and knowledge creation are being created in the same time the software system is being delivered to customer and the wealth is being generated. Requiring the small, cross-functional teams to keep working together and the process rigid structure to never be used. So, it seems agile is solving the creativity vs. structure tension dilemma, by simply

-11-

eliminating the need for structure to generate wealth and instead, offering a sustainable creativity model that can generate the wealth in the same time and as much as it can generate the knowledge and innovations.

V. Conclusions

Agile approach allows for increased creativeness and flexibility in software development. Its incremental time-boxed approach, that is being carried out be self-governing cross-functional teams, gives those teams the ability to be creative by allowing them to decide directly, in coordination with end users, on what features to offer next, based on previous release feedback and evaluation, instead of being restricted by a pre-determined rigid scope planning. The cyclic short period releases and the incremental manner, reduces time to market while maintaining quality of work, and in the same time, it allows to test new ideas in real production conditions and decide to withdraw or expand on them based on the feedback and evaluation.

In this reports, a review of literature was conducted to find out how agile approach has changed the software development industry. It covered what is agile, what is agile manifesto, and how does agile, based on it manifesto, did innovate the process of software development and overcame the shortcomings of the traditional Waterfall approach. Moreover, the literature review covered agile methodologies, with detailed focus on scrum, one of the most popular agile methodologies. Then the review went on to explore the advantages and criticisms of agile. And, it concluded with describing two high profile case studies of applying agile successfully.

Finally, in this report, agile approach was put under the lens, by juxtaposed it against Brown and Duguid theory about the tension between creativity and structure.

A. Recommendations

Based on this paper's literature review, following recommendations are offered:

-12-

- Agile is a viable approach that can be applied not just in software development sector, but also in other sectors, especially when the final product requirements are not clear and when there is a need for continues releasing of new features in short time cycles.
- Creating value to the customer is the ultimate reason of producing products or services, and it should always take precedence over rigid pre-decided scope.
 However, to achieve that, there should be a flexible but smart approach that will not result in scope creep or loss of focus.
- Agile approach proved that firms could be creative and profitable in the same time without the need to have separated or conflicting ways of doing the business.
- While agile approach solved many problems related to software development in compare with the waterfall approach, it is not always the right way to develop software; if the end requirements are mature and predictable, e.g. something we did many times before, and a predefined budget and time is very important, then in this case the traditional waterfall approach would be the better choice.

References

- [1] J. Highsmith and A. Cockburn, "Agile software development: the business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, Sep. 2001.
- [2] T. Dyba and T. Dingsoyr, "What Do We Know about Agile Software Development?," *IEEE Softw.*, vol. 26, no. 5, pp. 6–9, Sep. 2009.
- [3] B. Boehm, "Get ready for agile methods, with care," Computer, vol. 35, no. 1, pp. 64–69, Jan. 2002.
- [4] S. Nerur and V. Balijepally, "Theoretical reflections on agile development methodologies," *Commun. ACM*, vol. 50, no. 3, pp. 79–83, Mar. 2007.
- [5] K. Beck *et al.*, "Manifesto for agile software development," 2001. [Online]. Available: http://agilemanifesto.org/.
- [6] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131–133, Nov. 2001.
- [7] J. S. Brown and P. Duguid, "Creativity Versus Structure: A Useful Tension," in *Innovation: Driving Product, Process and Market Change (Edward B. Roberts, editor)*, Josey-Bass, 2002, pp. 41–45.
- [8] R. I. Sutton, "Weird Ideas that Spark Innovation," in *Innovation: Driving Product, Process and Market Change (Edward B. Roberts, editor)*, Josey-Bass, 2002, pp. 311–322.
- [9] A. H. Van de Ven, "Central Problems in the Management of Innovation," *Manag. Sci.*, vol. 32, no. 5, pp. 590–607, 1986.
- [10] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New directions on agile methods: a comparative analysis," 2003, pp. 244–254.
- [11] C. G. Cobb, *Making sense of agile project management: balancing control and agility*. Hoboken, N.J: Wiley, 2011.
- [12] tutorialspoint.com, "Scrum Overview," *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/scrum/scrum_overview.htm. [Accessed: 01-Jun-2017].

- [13] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," *jeffsutherland.org*, 2002.
- [14] The Agile Alliance, "History: The Agile Manifesto." [Online]. Available: http://agilemanifesto.org/history.html. [Accessed: 03-Jun-2017].
- [15] K. Beck *et al.*, "Principles behind the Agile Manifesto," 2001. [Online]. Available: http://agilemanifesto.org/principles.html.
- [16] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," 2003, pp. 308–313.
- [17] K. Schwaber, "SCRUM Development Process," in *Business Object Design and Implementation*, J. Sutherland, C. Casanave, J. Miller, P. Patel, and G. Hollowell, Eds. London: Springer London, 1997, pp. 117–134.
- [18] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer*, vol. 36, no. 6, pp. 47–56, Jun. 2003.
- [19] W. Tracz, "Extreme Programming (XP)," in *Encyclopedia of Software Engineering*, J. J. Marciniak, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2002.
- [20] G. Sambasivam, "EXTREME PROGRAMMING(XP)." [Online]. Available: http://www.unf.edu/~broggio/cen6016/ExtremeProgramming(XP)Article.pdf. [Accessed: 01-Jun-2017].
- [21] S. Fraser, K. Beck, B. Caputo, T. Mackinnon, J. Newkirk, and C. Poole, "Test Driven Development (TDD)," in *Extreme Programming and Agile Processes in Software Engineering*, vol. 2675, M. Marchesi and G. Succi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 459–462.
- [22] O. Cawley, X. Wang, and I. Richardson, "Lean/Agile Software Development Methodologies in Regulated Environments – State of the Art," in *Lean Enterprise Software and Systems*, vol. 65, P. Abrahamsson and N. Oza, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 31–36.
- [23] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," 2013, pp. 9–16.
- [24] Scrum Alliance, "The 2015 State of Scrum Report," Jul-2015. [Online]. Available: https://www.scrumalliance.org/scrum/media/scrumalliancemedia/files%20and%20pdfs/state%20of%20scru m/scrum-alliance-state-of-scrum-2015.pdf. [Accessed: 06-Jun-2017].
- [25] F. J. Pino, O. Pedreira, F. García, M. R. Luaces, and M. Piattini, "Using Scrum to guide the execution of software process improvement in small organizations," J. Syst. Softw., vol. 83, no. 10, pp. 1662–1677, Oct. 2010.
- [26] P. Deemer, G. Benefield, C. Larman, and B. Vodde, "Scrum Primer v2.0," 2012. [Online]. Available: http://www.goodagile.com/scrumprimer/scrumprimer20.pdf. [Accessed: 01-Jun-2017].
- [27] J. Porter, "The Freedom of Fast Iterations: How Netflix Designs a Winning Web Site," UX Articles by UIE, 14-Nov-2006.
- [28] Jez Humble, "Agile Case Studies and Lessons Learned," 13:07:43 UTC.
- [29] D. Greer and Y. Hamon, "Agile Software Development," Software: Practice and Experience, vol. 41, no. 9, pp. 943–944, Aug. 2011.
- [30] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," Information and Software Technology, vol. 50, no. 9–10, pp. 833–859, Aug. 2008.
- [31] M. Lindvall et al., "Agile software development in large organizations," Computer, vol. 37, no. 12, pp. 26– 34, Dec. 2004.
- [32] A. Begel and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study," 2007, pp. 255–264.
- [33] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," Journal of Systems and Software, vol. 82, no. 11, pp. 1869–1890, Nov. 2009.
- [34] P. Meso and R. Jain, "Agile Software Development: Adaptive Systems Principles and Best Practices," Information Systems Management, vol. 23, no. 3, pp. 19–30, Jun. 2006.