Towards a Better Understanding of Simplicity in Agile Software Development Projects

Wylliams Santos*
Informatics Center (CIn), Federal University of Pernambuco (UFPE)
Recife, Pernambuco, Brazil
wbs@cin.ufpe.br

ABSTRACT

The study of simplicity is an interdisciplinary endeavor with many concepts and attributes and the realization that there are many difficulties in defining simplicity and its relationships development and use. Agile software development has proven to be an important set of methods in promoting simplicity issues, however, the concept of simplicity is, in itself, by far not a simple concept because, there are many perspectives on the perception of simplicity. To address this gap, we propose an approach towards fostering simplicity in Agile software development. Additionally, we present the problem statement, research goal, research question and research methodology, including the research decision-making structure. Besides, the literature review covering models related to simplicity in different areas of research are then drawn.

Categories and Subject Descriptors

k.6.3 [**Management of Computing and Information Systems**]: Software Management—*Software Process*; D.2.9 [**Software Engineering**]: Management—*Productivity, Programming Teams, Software Process Models*

Keywords

Simplicity, Agile Software Development, Methodologies

1. INTRODUCTION

The Standish Group points out that the recent increase rates in IT project success is a result of several factors, including looking at the entire project's environment of processes, methods, skills, costs, tools, decisions, optimization, internal and external influences, and team chemistry [29]. Agile methods provide ways to develop software which place emphasis on people and their creativity [4]. Additionally, agile software development is a set of practices, methods, values, principles expressed in the Agile Manifesto. Its values focus on people and interactions, working software, customer collaboration, responding to change, and continuous improvement [8] [2].

According to the proposed Agile Manifesto [2], the Agile Software Development demands a focus on simplicity stating that "Simplicity is essential". Simplicity has emerged as a key [10] [18] [19], giving evidence that the philosophy of simplicity is strategically important. However, the concept of simplicity requires a greater understanding in order to be applied successfully. In this context, we believe that it is important to develop an approach to support the agile team in order to foster simplicity in Agile Software Development.

2. PROBLEM STATEMENT

Meyer [20] takes a closer look at the concept of the official agile principle, which defines simplicity as "the art of maximizing the amount of work not done". Meyer affirms that who has ever obtained a first solution to a problem of any kind, found it complex, and tried to simplify it. Achieving simplicity often means adding work, sometimes lots of it.

From this point of view, achieving simplicity is not the same as minimising work. Meyer [20] discusses that both are worthy goals in software engineering, but they arise in different contexts and lead to different principles: (i) proponents of rigorous, elegant programming techniques, (ii) avoiding unneeded work which leads to such principles as "Eliminate waste" and "Decide as late as possible" in Lean [24]. These two views meet, but not necessarily in the way agile authors [2] would like.

Simplicity has been considered an important research topic in Information and Communications Technology (ICT). Agile Software Development has proven to be an important set of methods in promoting simplicity issues, but the lack of a framework for applying simplicity can be an important factor in the lack of success for software development projects in meeting the project's target and achieving customer's satisfaction.

Motivated by scenario presented previously, the research question investigated by this thesis is: *(RQ) How to manage simplicity in order to improve Agile Software Development projects?*

The goal of this thesis can be stated as: *This work proposes an approach to manage simplicity in the context of Agile Software Development in order to improve the projects. The approach is supported by the state-of-the-art in the Agile*

*supervised by Prof. Hermano Moura, Informatics Center (CIn), Federal University of Pernambuco (UFPE), hermano@cin.ufpe.br

EASE '16, June 01-03, 2016, Limerick, Ireland

DOI: http://dx.doi.org/10.1145/2915970.2915978

Software Development context concerning Simplicity's understanding.

3. METHODOLOGY

Several researchers [9] [26] [6] have addressed the challenges in selecting an appropriate research method in empirical software engineering research in the last two decades.

3.1 Research Decision-Making Structure

In this sense, this work adopts the decision-making structure containing a number of decision points, each one of them representing a specific aspect on empirical software engineering research [30].

Figure 1 shows the decision points outlining the structure of the decision process during study design. The decision points are logically ordered from left to right. Initially, the decision making-structure shows the starting point (bull's eye), the identification of the research question. The outcome of research (DP1) is classified as basic (pure research) because it applies to a problem where the emphasis is on the understanding of the problem rather than providing a solution to a problem, hence the main contribution is the knowledge generated from the research [5]. The "Inductive Research" is adopted as Research Logic (DP2).

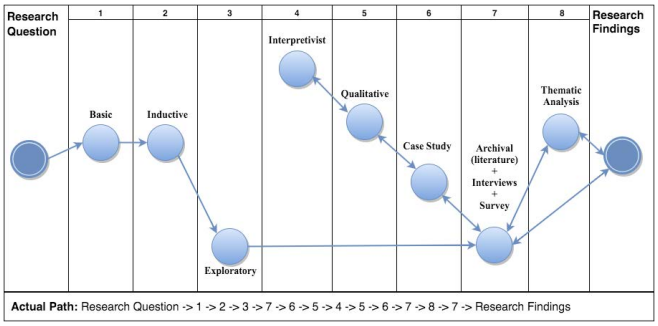

Figure 1: Research paths through the research decision-making structure

The purpose of this research (DP3) is classified as exploratory. According to Wohlin [30], exploratory research is applied when there is not much information available on the topic area the researcher aims to gather some insights about the problem. Typical data collection methods are observation, interviews, and focus group interviews. Exploratory questions attempt to understand the phenomena, and identify useful distinctions that clarify our understanding [9]. Our research question is also classified as description and classification [9].

The exploration is done from the perspective of the team in the Agile Software Development project, and hence the research is conducted (DP4) from an interpretivist (constrution-ivism) research approach.

The qualitative approach is defined as a research process (DP5). It is decided that the research should be conducted as a case study (DP6). We plan to conduct a systematic literature review as data collection methods (DP7), using thematic analysis (DP8) to identify to what extent Simplicity in agile software development is covered in academic literature. The literature review applies a qualitative approach as a research process (DP5), which involves an interpretivist research approach (DP4), and thematic analysis as a data analysis methods (DP8).

We plan to conduct a systematic literature review as data collection methods (DP7), using thematic analysis (DP8) to identify to what extent simplicity in agile software development is covered in academic literature. The literature review applies a qualitative approach as a research process (DP5), which involves an interpretivist research approach (DP4), and thematic analysis as data analysis methods (DP8).

3.2 Research Design Strategy

The research design strategy presented in Figure 2 is inspired on [15] [21] [25]. It is structured in two phases, they are "Approach Emergence" and "Approach Assessment" which encompass all the decision points covered by Figure 1. Each phase comprises steps, as a result of each step, several products are derived.

The first phase (emergence approach) is composed by the (i) theoretical methodological positioning, which establishes the methodological strategy, (ii) initial literature review, gathering knowledge, (iii) semi-structured interviews conducted in order to gather data from practitioners and researchers in several companies (Ireland, England, United States and Canada)[1] and research centers, (iv) Systematic Literature Review, and (v) Establishing and refining the approach.

The second phase of our research design strategy 2 incorporates the industrial case study in order to evaluate, refine and consolidate the proposed approach.

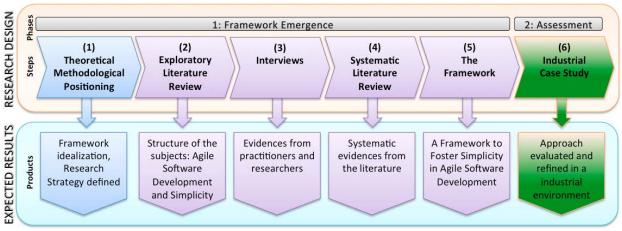

Figure 2: Research Design Strategy (Inspired on [15] [21] [25])

4. THEORETICAL BACKGROUND

Simplicity principles have been proposed in various forms by theologians, philosophers, and scientists, from ancient to modern times. There is a widespread philosophical presumption that simplicity is a theoretical virtue [1] [11].

Several works [10] [18] [19] analyse simplicity in the context of Information and Communications Technology. The findings resulted from a Systematic Literature Review and direct interaction with experts (individual interviews and focus group) in the area of ICT [10] [17] give evidence that the community of researchers and practitioners believe that the philosophy of simplicity is strategically important, yet still insufficiently understood.

According to Dingsøyr et al. [7], the articulation of the Agile Manifesto in 2001 has brought unprecedented changes to the software engineering field. The Agile Manifesto is a statement of values and principles that describe the various

[1]project approved by the Research Ethics Committee

agile processes [2], as follows: (i) Individuals and interactions over processes and tools, (ii) Working software over comprehensive documentation, (iii) Customer collaboration over contract negotiation, (iv) Responding to change over following a plan.

In order to satisfy these values, some principles[2] should be respected, among them, the principle that "Simplicity, the art of maximizing the amount of work not done – is essential" available literature, various methods propose agility in their definitions, aiming to find efficient ways for developing software of quality across an agile development process.

According to Meyer [20] simplicity arises in different contexts and leads to different principles: (i) proponents of rigorous, elegant programming techniques, (ii) avoiding uneeded work which leads to such principles as "Eliminate waste" and "Decide as late as possible" in Lean.

5. PROPOSED APPROACH AND FUTURE AGENDA

Our approach proposes five simplicity perspectives in the context of Agile Software Development, such as agile team, product, customer, user and process. Conforming to Margaria [17] there are many perspectives on the concept of simplicity for example, simplicity can be related to size as in the number of components a system processes. It can also reflect the amount of effort a user of the system has to expend to use the system or the level of effort and amount of knowledge to understand the system.

In the context of the present study, we are particularly interested in proposing a coherent approach of interrelated fundamental directions fostering simplicity in the context of Agile Software Development. Initially, our framework is composed by several dimensions that leads to simplicity in the context of Agile Software Development, such as: reduction, orthogonality, time, structure, communication, organization, decomposition and so on.

Figure 3 provides an overview of the conceptual framework illustrating the nature and relationships between the different components of the pyramid. The pyramid is revolving around Agile Manifesto, Dimensions of Simplicity and Critical Success Factors (CSF), as its cornerstones. The philosophical concept of simplicity is considered as the centre or as the core part of the framework. It is set up and inspired by Egyptian pyramids, certainly one of the most perfect and extraordinary shapes created by humans [27]. Humbert and Price [12] argued that the regular tetrahedron, comprising only four equilateral triangles, has at least as forte a claim to simplicity and symmetry. Analogously, these are the main characteristics of our pyramid.

The basis of the pyramid lists the Agile Manifesto and Dimensions of Simplicity. The Agile Manifesto unifies and establishes a common set of values and principles for all of the agile methodologies, including the tenth principle that addresses simplicity as essential "the art of maximizing the amount of work not done". The Dimensions of Simplicity's triangle identify the structures and aspects that lead to simplicity.

On the top of the pyramid, Critical Success Factors - are the factors that must be present for the agile project to be successful. In this sense, we believe that the simplicity dimension affects and is related to some Critical Success Factors [3] [28]. The core (centre) of the pyramid, extracts the essence of simplicity according to the philosophical concept.

[2]http://agilemanifesto.org/principles.html

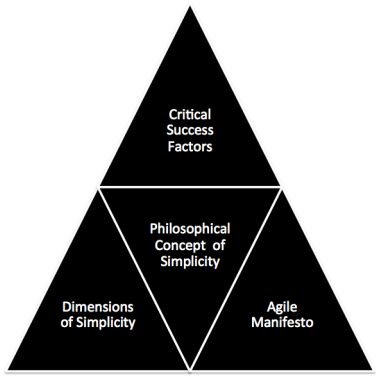

Figure 3: Conceptual Framework of Simplicity in Agile Software Development

Additionally, relationships between the components of the pyramid are not stated in a linear/sequential way. Essentially, they must be present in order to keep the spirit of simplicity.

Thus, if well analysed, the simplicity dimension could lead to increasing the success rates of the Agile Project. This new way of dealing with simplicity in the context of Agile Software Development requires the capacity to rethink the underlying competences under a different image of project, demanding a new team's mindset in order to further boost the success of the projects with focus on simplicity.

As a partial result of the work presented in this thesis, the following main contributions can be highlighted: (i) an overview of the key concepts in the field of simplicity, with emphasis on the Agile Software Development aspects; (ii) a survey [14] performed with practitioners from the European and American companies, besides researchers in order to understand the aspects of Simplicity in Agile Software Development; (iii) an initial conceptual framework in underlying simplicity definition from the Agile Team's perspective; (iv) a focus group performed with Brazilian practitioners and researchers in order to refine and evaluate the conceptual framework underlying simplicity definition from the agile team's perspective;

As the next steps and statements of future contributions, our schedule proposes: (i) a Systematic Literature Review (SLR) [13] in order to understand the challenges and opportunities of Simplicity management, as well as to identify research gaps and the road ahead; (ii) the establishment of an approach to simplicity management in Agile Software Development; and, (iii) the evaluation of the proposed approach.

6. CONCLUDING REMARKS

Simplicity is a mindset, a deeply rooted cultural paradigm [18]. In this sense, simplicity has been increasingly recognized as a driving paradigm in ICT development, maintenance, use and management [10] [16] [19] [22] [23].

Agile Software Development emerged as a popular paradigm and was broadly adopted by the software industry, yet some theoretical concepts are still insufficiently understood by practitioners and researchers. In this context, we argue

that an exhaustive comprehension of simplicity could support academics and practitioners in the direction to improve agile software development.

Acknowledgements

The authors would like to acknowledge the Brazilian National Research Council (CNPq), Brazil′s Science without Borders Program, University of Pernambuco (UPE), Lero – the Irish Software Research Centre, Science Foundation Ireland (SFI), University of Limerick (UL) and Professor Tiziana Margaria for their support during this research.

7. REFERENCES

[1] A. Baker. Simplicity. The Standard Encyclopedia of Philosophy, 2013. Accessed on February 22, 2015.

[2] K. Beck, M. Fowler, and K. Schwaber. Manifesto for Agile Software Development, 2001. http://agilemanifesto.org. Accessed on February 22, 2016.

[3] T. Chow and D.-B. Cao. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6):961–971, 2008.

[4] A. Cockburn and J. Highsmith. Agile software development: The people factor. *Computer*, 34(11):131–133, 2001.

[5] J. Collis and HusseyRoger. *Business Research: A Practical Guide for Undergraduate and Postgraduate Students*. Palgrave Macmillan, London, United Kingdom, 3rd edition, 2009.

[6] J. Creswell. *Research design: qualitative, quantitative and mixed methods approach*. SAGE Publications, Inc, 4th edition, 2013.

[7] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6):1213–1221, 2012.

[8] T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10):833–859, aug 2008.

[9] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting Empirical Methods for Software Engineering Research. In *Guide to Advanced Empirical Software Engineering*, chapter Selecting Empirical Methods for Software Engineering Research, pages 285–311. Springer London, 2008.

[10] B. D. Floyd and S. Bosselmann. ITSy - Simplicity Research in Information and Communication Technology. *Computer*, 46(11):26–32, Nov. 2013.

[11] J. C. Gambrel and P. Cafaro. The Virtue of Simplicity. *Journal of Agricultural and Environmental Ethics*, 23 VN - r:85–108, 2009.

[12] J.-M. Humbert and C. Price. *Imhotep Today: Engyptianizing architecture*. UCL Press - Institute of Archaeology, London, United Kingdom, 2003.

[13] B. Kitchenham and S. Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, Keele University and Durham University Joint Report, 2007.

[14] B. Kitchenham and S. L. Pfleeger. Principles of survey research part 4: questionnaire evaluation. *ACM SIGSOFT Software Engineering Notes*, 27(3):20, 2002.

[15] A. J. H. d. O. Luna. *Agile Governance Theory*. Ph.d. thesis, Informatics Centre (CIn), Federal University of Pernambuco (UFPE), Recife, Brazil, 2015.

[16] T. Margaria, B. D. Floyd, and B. Steffen. IT Simply Works: Simplicity and Embedded Systems Design. In *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, pages 194–199. IEEE, July 2011.

[17] T. Margaria, B. D. Floyd, and B. Steffen. ITSy – Recommendation Document. Technical report, University of Postdam, Postdam, 2011. Accessed on February 21, 2016.

[18] T. Margaria and M. Hinchey. Simplicity in IT: The Power of Less. *Computer*, 46(11):23–25, Nov. 2013.

[19] T. Margaria and B. Steffen. Simplicity as a Driver for Agile Innovation. *Computer*, 43(6):90–92, June 2010.

[20] B. Meyer. *Agile! The Good, the Hype and the Ugly*. Springer, Zurich, Switzerland, 2014.

[21] M. Monasor, V. Aurora, M. Piattini, J. Noll, and S. Beecham. Towards a global software development community web: Identifying patterns and scenarios. *Proceedings - 2013 IEEE 8th International Conference on Global Software Engineering Workshops, ICGSEW 2013*, pages 41–46, 2013.

[22] H. Moura. Software Project Framework. Technical report, Federal Univerity of Pernambuco, Recife, Pernambuco, 2011. Accessed on February 21, 2015.

[23] H. Moura and M. Skibniewski. The Evolution of Management Thinking. In *International Research Network on Organizing by Project (IRNOP)*, 2011.

[24] M. Poppendieck and T. Poppendieck. *Leading Lean Software Development: Results Are Not the Point*. Addison-Wesley Professional, 1st edition, 2009.

[25] S. Reinehr. *Reuso sistematizado de software e linhas de produto de software no setor financeiro: estudos de caso no Brasil*. Phd thesis, University of São Paulo, 2008.

[26] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, dec 2008.

[27] M. G. Schadla-Hall RT. Ancient Egypt on the Small Screen - from Fact to Fiction in the UK. In *Consuming Ancient Egypt*, chapter 14, pages 195–215. UCL Press - Institute of Archaeology, London, United Kingdom, 2003.

[28] D. Shahane, P. Jamsandekar, and D. Shahane. Factors Influencing the Agile Methods in Practice - Literature Survey & Review. In *International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 556–560, 2014.

[29] The Standish Group International. CHAOS MANIFESTO 2013. Technical report, The Standish Group International, 2013.

[30] C. Wohlin and A. Aurum. Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, 20(6):1427–1455, 2014.