PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
FACULTY OF INFORMATICS
COMPUTER SCIENCE GRADUATE PROGRAM

FLUENCY IN LARGE-SCALE AGILE DEVELOPMENT

GREICE DE CARLI ROMAN

Research plan submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fullfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Sabrina Marczak, PhD

Porto Alegre
2016

LIST OF FIGURES

Figure 4.1 – The stages of Agile Fluency model 15
Figure 4.2 – Model for an Adaptive Organization. Taken from: [36]. 17
Figure 6.1 – Proposed research design 25

CONTENTS

1 INTRODUCTION **6**

2 MOTIVATION **8**

3 RESEARCH GOAL AND QUESTIONS **11**

3.1 MAIN RESEARCH GOAL AND QUESTIONS 11

3.2 OBJECTIVES 11

4 THEORETICAL FOUNDATION **12**

4.1 AGILE SOFTWARE DEVELOPMENT 12

4.2 AGILE TRANSFORMATION 13

4.2.1 AGILE FLUENCY 14

4.3 LARGE-SCALE AGILE SOFTWARE DEVELOPMENT 16

5 RELATED WORK **21**

6 PROPOSED RESEARCH METHODOLOGY **25**

6.1 PHASE 1: FOUNDATION 25

6.2 PHASE 2: EXPLORATION 26

6.3 PHASE 3: SOLUTION 26

6.4 PHASE 4: EVALUATION 27

7 DETAILED RESEARCH PLAN AND CURRENT PROGRESS **28**

7.1 PLANNED ACTIVITIES AND SCHEDULE 28

7.1.1 PHASE 1: FOUNDATION 28

7.1.2 PHASE 2: EXPLORATION 28

7.1.3 PHASE 3: THEORETICAL PROPOSAL 29

7.1.4 PHASE 4: EVALUATION 29

7.1.5 ADDITIONAL ACTIVITIES AND DOCUMENTS 30

7.1.6 PLANNED SCHEDULE 30

7.2 CURRENT PROGRESS 31

REFERENCES **33**

ATTACHMENT A – Paper Accepted at the ELA-ES 2015 . . . . **39**

ATTACHMENT B – Paper Accepted at the WBMA 2015 . . . . **44**

ATTACHMENT C – Research Protocol of Systematic Review Literature **53**

FLUENCY IN LARGE-SCALE AGILE DEVELOPMENT

ABSTRACT

The Agile Manifesto was written in February 2001. Since then, companies are discussing whether to migrate to agile and how to do it. Academia has been supporting industry to go through the transformation process (e.g., helping teams to become self-managed). Even agile are made for small and co-located teams, the adoption of agile software development are increasing in other settings (e.g. large companies) with the term 'scaling agile'. However, despite the vast literature on such topic, there is a gap in literature about how scale teams obtain fluency in large-scale agile software development. This research proposal presents the opportunity to characterize fluency in large-scale agile software development. To accomplish this goal, it will be created a Fluency Framework according to literature and a field study with industry personnel and agile experts who work/worked in a large-scale agile development. Finally, I will evaluate the Fluency Framework using member-checking technique. With this study, it will be expected that scale teams can understand in how stage of fluency they are and how to achieve others stages to gain the benefits brought in large-scale agile development.

Keywords: Agile, Large-Scale Software Development, Fluency, Scale Teams.

1. INTRODUCTION

For almost fifteen years Agile Software Development offers new values and motivate IT companies to deliver faster to the customer high-quality products. Right from the start, companies are discussing whether to become agile and how to go about transforming themselves to achieve such 'agility'. The transformation process involves more than deciding on which agile method to adopt or to change a way to produce software. It refers to acquire the 'powerful agile mindset' thought behavior and culture [55]. For instance, teams will learn to become self-managed, communication between teams over than to define processes to produce software in a short time controlled by interactions, the customer should collaborate with the software production, among other aspects.

Related to agile transformation, academia has been supporting industry as long as the agile manifesto [6] has been defined ([58]; [9]). Despite agile were made for small, co-located development teams, its have been increasingly applied in other settings. Several large projects [70], with a number of teams that develop complex systems [32], with distributed or dispersed teams [52] have started to use agile development at scale.

The term of "scaled agile development or agile development at scale" was introduce in academia by Sutherland [69], in his study "Agile can Scale" that presents five different companies size that adopted Scrum method in their processes, and concludes that Scrum works in any environment even in large settings. After that, many others studies use this term to identify companies that adopted agile at large enterprises ([35], [32]). And also, the large-scale agile development term was defined to identify the companies who works at scale but in a context much more complex, with a range of 2-9 teams as suggested in a taxonomy proposed by Dingsøyr *et. al.* [16].

However, achieving success of large-scale agile development as reported by Fry and Greene [21], for example, is a complex process and brings numerous challenges to organizations [34]. For instance, how much can requirements keep changing when they cross hundreds of applications at a time? Dingsøyr and Moe [17] presents a research agenda on the topic showing that there are still several open questions to discuss about coordination of work between scale teams; organization of large development efforts; variability factors in scaling; metrics - key performance; release planning, architecture and others. Waardenburg and Vliet [72], on the other hand found that increased landscape complexity and lack of business involvement are both challenges factors when a large company adopt Agile. The study also presents communication strategies to mitigate this challenges.

Given the considerable amount of large companies migrating to agile (e.g. [50] [35], [32], [10] [44]). Other several studies of maturity models (e.g. [64], [20], [23], [62], [66]) for agile transformation and even for scaling agile development [67]. There is a needed to consolidated more extensive body of knowledge of fluency of scale-teams in large-scale agile development. The word ’fluency’ means the scale-teams’ maturity to perform large-scale agile development.

Due the area is still immature and with few works, as it will be described in Chapter 5 (Related Work). In this work I will understand what are the characteristics defined to large-scale agile development and how scale-teams can acquire fluency in these aspects of large-scale agile development.

To support that it will be created a Fluency Framework according to aspects found in literature about large-scale agile development and its related concepts. Also in a Field Study with industry personnel and agile experts to understand how scale teams get Fluency in Large-Scale Agile Development. Finally, I will evaluate this Framework using Member-Checking technique.

It expected with this work that scale-teams can understand in how stage of fluency they are and how to achieve others stages to gain the real benefits brought in large-scale agile development.

This remainder of the document is organized as follows:

- CHAPTER 2 - Presents the motivation of work;
- CHAPTER 3 - It will be shown the research goal and the main questions of this work;
- CHAPTER 4 - Reviews the Theoretical Foundation relating the main concepts;
- CHAPTER 5 - Proposes the research methodology with the phases of work;
- CHAPTER 7 - Presents the detailed research work and the current progress with schedule and on going activities;

2. MOTIVATION

Agile Development offers new values to motivate IT companies to deliver high-quality products faster and produce satisfied customers. Since the Agile Manifesto was written, companies are discussing whether to become agile and how to achieve agility. According to the data in 9th Annual State of Agile Survey [73], 20% participants worked in very large organizations with more than 20.000 people. And the number-one tip for scaling agile successfully is having consistent process and practices (42%), followed by executive sponsorship (40%) and the implementation of a common platform across teams (39%). This is an interesting issue, cause the agile has been proposed to small and medium settings [12] and is increasingly appearing in other settings with successfully implementation.

Then, despite the benefits (including higher satisfaction, a feeling of effectiveness, increased quality and transparency, increased autonomy and happiness and earlier detection of defects) brought in this Agile transformation, as reported at Nokia [37] and others cases of successfully adoption of Agile methods in large enterprises (better teams communication and customer collaboration) [52]; (self organizing scenario based teams, customer collaboration) [33]; (organizational adaptation, safe self-organizing teams) [7]. The transformation is arduous and there are many gaps, challenges and concerns that these companies have to face to adopt agile methods at scale.

For example, how large companies with globally-distributed teams go thought an successfully Scaled Agile Transformation?, Paasiavara [50] has an unsuccessful case in applying scaling scrum transformation. what are the main themes and challenges in requirements in large-scale agile software development? Rolland [59] in this paper identify this open question for future research. And about communication challenges in geographically distributed agile development? Alzoubi *et.al.* [1] in this systematic review found six key challenges categories when a company scaling agile (distance differences, team configuration, customer communication, project characteristics, organizational factors and human factors).

Thus, to support and mitigate the main challenges and concerns of agile transformation in large settings were developed best practice books and frameworks to Scaled Agile Development. Larman and Vodde [[38], [39]], Schiel [61]) describe in their books a detailed guide for conducting agile adoption on a large scale. Providing a step by step guidance for leading into an enterprise transformation. Following of Scaled Agile Framework SAFe [[42], [43]] that was introduced to a wide audience in the Agile 2013 conference in August, Less [40] that was introduced by Craig and Vodde, and Disciplined Agile Delivery (DAD) [4] by Ambler in the International Conference of Software Engineering in May 2013. Even, well-known Pioneers in Agile Project Planning and Management, VersionOne [46], are also supporting Scale Agile Development helping organizations of all sizes scale agile faster, easier and smarter. All these models have been created by practitioners.

Although there are models to adopt Scaling Agile Software Development, being 'agilist' involves much more than purely matters of adoption practices. Even for the agile practices involve behavioral issues and exchange of culture, as stated in the manifesto 'individual over process and tools' [6]. However, there are some academics as well as industry works observing aspects related to team behavior. The following studies describe the perspective from studying the teams maturity through the adoption of agile methods. More details of these works present in the next paragraphs it will be shown in Related Work, Chapter 5.

The first one was created by 'practitioners to practitioners' due to complaints of organizational leaders that they're not getting the benefits from Agile they expected. Diana Larsen and James Shore create an model called Agile Fluency that through four distinct stages, each with its own benefits, costs of adoption, and key metrics shows the capability of the individuals inside of an agile team. The true agile fluency is a skillful, routine practice that persists when your mind is distracted with other things [41].

Other framework described by Gandomani *et. al.* [22], related through an agile transformation and adoption following a set of characteristics including: practice selection, adapting, assessment, retrospective and adjustment. This framework are iterative, gradual, continuous, and value-based and show promise of being useful in software companies and organizations, regardless of size.

Fontana *et. al.* [20], on the other hand identified the following categories to acquire maturity in agile development: practices to become agile, team composition and behavior, deliveries (evolution from traditional to continuous delivery), requirements (transition from traditional requirements elicitation to use stories), product (practices to improve the software product), and customer relationship.

And also, relating to maturity, in my initial literature review I found one study that creates a maturity model for adopting scaling agile and Safe (Scaled Agile Framework) practices [67]. By the way, this work despite being a maturity model it was only used to map the evolution of a framework. As aforementioned, there are other frameworks in the market to scale agile development is so difficult to know whether this case will serve for all scale-teams, even those that do not adopt frameworks or to adopt different frameworks. This work also seems to prescribe a process to achieve scaled agile.

Thus, I finally conclude that large companies can scaled agile development and to achieve benefits and improvements in software development [45], [27] even with the difficulties encountered and the efforts to be made for the adoption [47], [1], [50]. To help them there are some books and frameworks in literature that support the use of scaled agile practices and adapting them to their contexts.

Due this topic is still emergent and under development [36] there is a needed to understand how scale-teams can acquire fluency (the exchange of mindset, behavior) in a large-scale agile development practices. The contribution of this work is two-fold. The first is to understand which fluency level any scale-team are. And the second is to define how scale-teams can achieve higher levels to brought the benefits in large-scale agile software development.

3. RESEARCH GOAL AND QUESTIONS

This section presents the main research goal and associated research questions of my proposal as well as the related objectives to achieve the posed goal.

3.1 Main Research Goal and Questions

The main goal is to characterize fluency in large-scale agile development. To accomplish that, one research question will be used to drive the research goal:

Research Question (RQ):
How large-scale teams acquire fluency in large-scale agile development?

3.2 Objectives

To achieve the main goal and answer the posed research questions, I define the following objectives:

(Obj1). To understand the related concepts of Agile Software development and Agile Transformation.

(Obj2). To identify the state of art of the Large-Scale Agile Development in literature.

(Obj3). To identify the concept of Fluency in Large-Scale Agile Development in practice.

(Obj4). To define a Fluency Framework based on the findings from Objectives 1, 2 and 3.

(Obj5). To evaluate and to collect feedback about this proposed Fluency Framework in order to investigate its applicability.

4. THEORETICAL FOUNDATION

This chapter presents a theoretical background about the three main concepts related to this research: Agile Software Development, Agile Transformation and Large-Scale Agile Software Development.

4.1 Agile Software Development

Agile Software Development arise in the early 2000s with the purpose to attend the need to define a new way to produce software in a scenario driven on changes, in which traditional methods (e.g. waterfall) was not more satisfiable. Agile Software Development are based on releases and faster deliver to the customer, as defined to Sommerville [65]:

"Agile methods are incremental development in which the increments are small and typically new releases are created and available to the customer every two or three weeks. They involve the customer in the software development process for quick feedback on changes in requirements. They minimize the documentation using an informal communication rather than formal meeting with written documents."

They are based on the tendency known as the Agile Manifesto, proposed from empirical knowledge of 17 renowned developers, authors and software business consultants as the authors Martin Fowler and Kent Beck. The manifesto was based on four values, which are presented below, according to the Agile Manifesto [6]:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan;

According to Agile Manifesto [6], while there is value in the items on the right, they value the items on the left more. To understand the value of Agile Development, after the Manifesto was written twelve fundamental principles as follow:

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software;
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage;

- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale;
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity–the art of maximizing the amount of work not done–is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Development has individuals values and principles going beyond of a vision of following a software development process. In essence, the agile methods promotes values and practices to achieve the term 'agility'. Agile not define a process with beginning and ending with a list of procedures to follow, Agile is about culture and behavior from the perspective of people [13].

Its values are based on the interaction between individuals to share the knowledge to produce software with quality in a short time (releases). That promotes customer collaboration and more communication between the team. This frequent interaction between individuals does not require to have a formal documentation [11] as waterfall and traditional methods prescribe.

4.2 Agile Transformation

The agile transformation process (ATP) has been encouraged to remedy inherent problems of traditional software development [22] and is defined as *the process of leaving the traditional way to development software and adopting the agile philosophy, tools, and principles* [55].

A true ATP must focus on 'being' agile rather than 'doing' agile. Indeed, agile adoption is not only following some specific practices defined by agile methods [34] or even adopt some agile method by the book. Is the change of mindset, contain people value first of all and following the Agile Manifesto values [57]. Adopting agile practices is a process of continuous learning and improvement. Then, the transformation to agile requires hard work, intense focus, and strict discipline [60].

This is the main reason that makes Agile Transformation Process more difficult than expected. Nerur *et. al.* [47] write on their paper that organizations must carefully assess the readiness before treading the path of agility. And they identify a set of key challenges when an organization decides to go thought to Agile Transformation, as follows: management and organizational (organizational culture, management style, organizational form; management of software development knowledge; reward systems); people (working effectively in a team, high level of competence, customer relationship); process (people-centric approach, adaptive and iterative software development, scalable projects, selecting the agile method appropriate) and technology (new skill sets, appropriateness). On the other hand Gandomani *et. al.* [24] identified impacts of human aspects in an agile transition and adoption. This study showed that people and their mindsets and behaviors strongly affect all transformation related activities as is depicted.

To characterize an Agile Transformation Process Gandomani *et. al.* [22], identify a set of categories that has an agile transformation as follows: prerequisites to become agile, training on methods and what it is about, facilitators (people who will guide the process), transition framework (a stepwise view on how to do it), managing the transition, assessment of progress, reasons for aiming to agility, coaching the transition, technical issues, human aspects-related issues, customer-related issues, selection of pilot projects for the transition, and agile method selection.

Observing successfully cases of Agile Transformation, there are adoptions with different sizes. Even large enterprises as Yahoo [10], IBM [3], Amazon [5], BMC [26] or in small cases [53]; [58] since the manifesto was created.

4.2.1 Agile Fluency

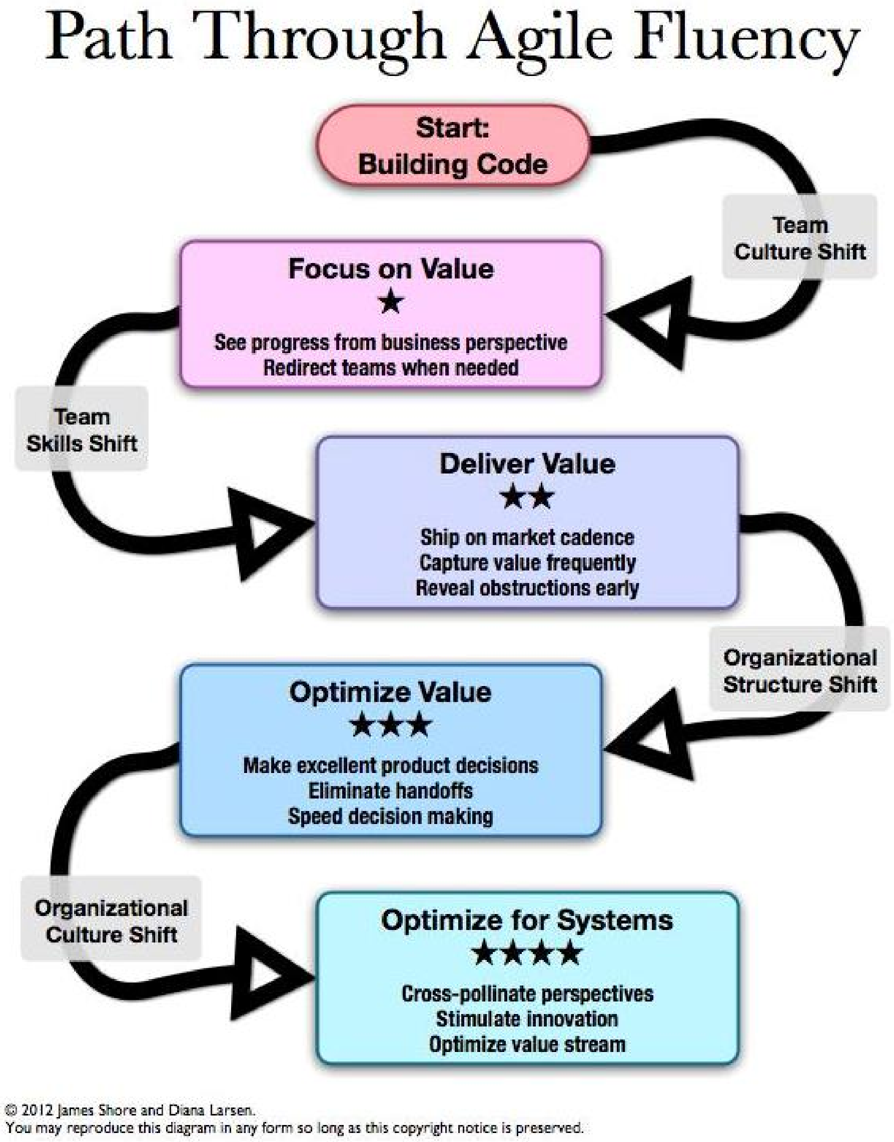
Agile Fluency model helps an agile teams to growth thought the usage of agile benefits. The Fluency evolves through four distinct stages, each with its own benefits, costs of adoption, and key metrics (see in 4.1). Diana Larsen and James Shore Agile Fluency model creators have been helping teams transition to Agile since the beginning of Agile Manifesto. And with this model they believe to brings what is needed to achieve the benefits promised by Agile.

Fluency means when a team develops software when is under pressure. Anyone can follow a set of practices when given time to focus in a classroom; true fluency is a skillful, routine practice that persists when your mind is distracted with other things [].

In the Agile Fluency model, is mainly consider the team fluency rather than individual or organizational fluency. Agile development is fundamentally a team effort, and your organization's success with Agile will depend on the fluency of your teams.

Teams progress through four distinct stages of Agile fluency, which we describe with a "star" system. Each star includes fluency at all previous levels. Teams' practices become more deeply and reliably grounded when they work this way. The next paragraphs are described the four stars in detail []:

- The One-Star the teams create business value. Rather than planning in terms of technical considerations, such as software layers, one-star teams plan in terms of the ben-



© 2012 James Shore and Diana Larsen.
You may reproduce this diagram in any form so long as this copyright notice is preserved.

Figure 4.1 – The stages of Agile Fluency model

efit their sponsors, customers, or users will see from their software, typically with user stories.

- Two-Star the teams deliver must be fluent besides focusing business value, they realize that value by shipping as often as the market will accept it. This is called “shipping on the market’s cadence.”
- Three-Star the teams must optimize their value for their investment. They understand what the market wants, what your business needs, and how to meet those needs. Or, as in a start-up environment, they know what they need to learn and how to go about learning it.
- Four-Star the teams contribute to optimizing the system to enterprise-wide success. Team members understand organizational priorities and business direction. Four-star teams will sacrifice their own needs to support the needs of a product more critical to business success. They work with other teams and with managers to optimize the overall value stream.

In Agile Fluency model the teams follow a typical progression in their understanding of Agile and they brought the real agile benefits to the organization. This progression or evolution was grouped into four stages of fluency. Each stage is characterized by unique benefits and distinct challenges to adoption.

4.3 Large-Scale Agile Software Development

Agile methods were made to small and collocated teams [58]. However, over time and with agile having goods results to best suit the necessity of time-to-market, others environments showed the interested in to migrate to agile. Large organizations (e.g. [21]) that already done such transformation were satisfied with the results of the transformation, having teams with greater flexibility and adaptation, even in complex environments [3].

The common nominator to different aspects of Large-Scale Agile Development is that people have used agile thinking to solve problems in discipline as Architect, Design, Marketing, Portfolio Management or Program Management, etc [36]. For example, Dingsøyr and Moe [17] identified four principles to be observed in large-scale agile development, namely: *architecture* - figuring out how work is coordinated; *inter-team coordination* - creating effective knowledge networks is essential due to the knowledge-intensive nature of software development; *portfolio management* - providing continuous feedback from the portfolio to project levels enables the teams and project members to take decisions that are consistent with the goals of the large-scale agile portfolio, and *scaling* - describing the context for agility and scale is essential for understanding how to improve agility in large-scale agile.

In conclusion we can define that i.e. Scaled Agile is when a company would be in an agile thinking on all levels of the organization [36], as seen in Figure 4.2 .

The characteristics aforementioned related to Scaled Agile Development are seen in empirical studies in literature as described bellow:

- The first one characteristic found in literature is that the teams can be distributed [52], dispersed or globally distributed [30]. To compile and define such topic a taxonomy was proposed [16] starting the number of teams to consider large is more than two teams that needed coordination, and practices such as Scrum of Scrums or feature teams.

- The second one is the relevant large number of the projects couple [18]. And the main concerns are related to coordination and collaboration between the entire team (business, developer, tester) [28]. To solve the problem events such as Communities of Practice [51] promote the communication and spread agile to entire company.

- The legacy code (some legacy data sources are questionable at best, or the owners of those data sources difficult to work with, yet that doesn't given an agile team license to create yet another database) [2] and complexity of the systems (value clean code that it is the best assurance against degradation of team velocity over time) [71] are others aspects to attempt in Large-Scale Agile Software Development.

- To solve the aforementioned challenges the architecture is the core characteristic of Large-Scale Agile Software Development. And it was defined ([71]; [48]) o common vocabulary and a equal culture to speak about the system, a systematic way to control

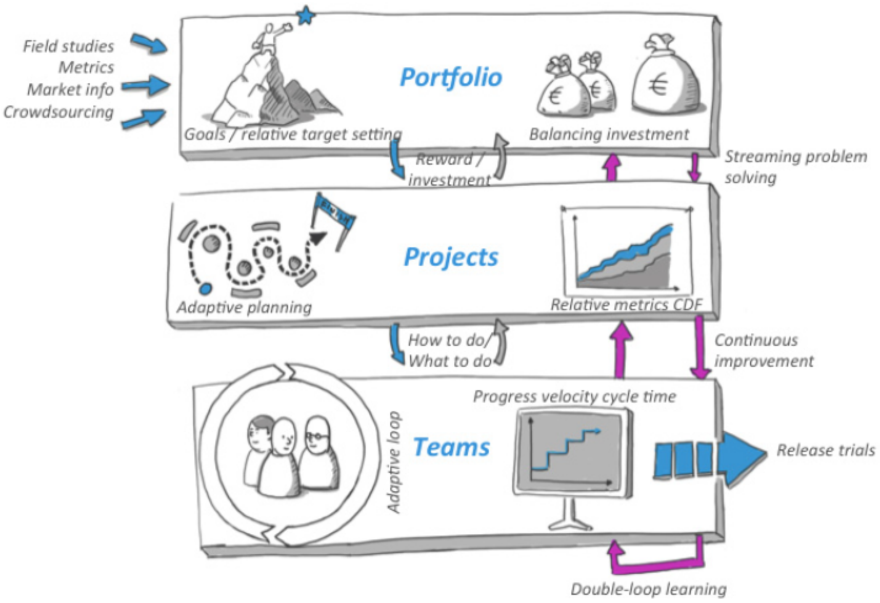


Figure 4.2 – Model for an Adaptive Organization. Taken from: [36].

dependencies—, a way to keep technical debt in check,a guide for release planning and configuration management.

- Other characteristic but not least is the change of organizational structure. It is very strongly advised to be divided by Areas or Product Areas ([49];[2]).

What is perceived is that these organizations face many challenges to do such transformation due the factors related to this adaptation with the aspects mentioned before. Some studies reveal difficulties when applying in a Large-Scale Agile Transformation, as we seen at IBM Rational [2]: the team size, the geographical distribution of the team, the entrenched culture, the system complexity and legacy, the regulatory compliance, the organizational distribution, the governance and the enterprise focus are the main factors to Scaled Agile. On the other hand, Ganesh and Thangasamy [25] defined that to track work among large teams, to achieve the system-wide properties of the software, and to guarantee the simultaneous releases of cross-cutting features are others key challenges in a transformation.

Laanti [36] defined twenty-one principles for Scaled Agile. Each principle was extracted by Toyota [], or Lean [], or Beyound Budgeting [] or Blue Ocean principles [], as follows:

- The content is the key: use the feedback from user and the intrinsic knowledge based on expertise and experience to create the best you can dream of.
- Co-creation: groups are faster solving problems than individuals. Let the software evolve together, as the sum of the whole is more than its parts.
- Feedback is the fuel to learning: use rapid and concrete feedback on all work done.
- Business agility: Releases generate revenue. The business model must dictate the release rate and user interest defines the business model.
- Use of automation as leverage: use automation to leverage the manual effort needed.
- Scale using fractals: Fractals are nature's way to scale, and fairly permanent structures. Use higher abstraction levels and nested systems, such as nested control loops.
- Avoid combinatorial explosions: Complexity is best tamed by splitting it to smaller pieces. Internal releases must be as small as possible.
- Sequence for maximal throughput: Modular architecture increases speed. Find the maximum throughput for your portfolio by balancing what can be done in parallel, and what must be done in sequence.
- Appreciate deep knowledge: Only more than five years experience creates deep knowledge. Use the best experts to tackle the most important and wicked problems.

- Work Leveling: even distribution of work and elimination of unnecessary work and waiting time based on measured performance.
- Simplicity: seek simplicity in solutions.
- Situationality: use Pareto principle to avoid making processes overly complex. Not all cases need to be treated equally.
- Control process, not items: create simple rules for decision-making, instead of controlling each decision individually.
- Growth mindset: do more of what created success. Best leaders do not reject faulty attempts, but instead twist them to create more success.
- Listen to employees, they know all the problems: value is created in the front-line. The rate at which you are able to remove impediments of progress or service correlates with the improvement done for business.
- Detect and use patterns: use and apply patterns. Your problems have already been solved by someone and somewhere.
- Cost innovation: ease the user's burden with a solution that costs less. Provide better service or fill the gaps between value chains.
- Utilize tacit knowledge: crowdsource the strategy. Use tacit knowledge of people to tell if you are heading in the right direction or not.
- Learning happens between teams: create collective knowledge that share the same vision and ambition. Collective must have multitalented semi-permanent teams combined with deep individual knowledge.
- Fast is better than perfection: Maximize the work undone. If it is not broken, do not fix it. Tolerate small imperfections.
- Prevent problems when small: success hides small problems. In order to stay successful do not become ignorant for small problems.

According to principles seen previously, it is necessary for Large-Scale Agile Development to change and to add some agile practices in accordance with context of its company [18]. Ivček *et. al.* [30] report the adaptation of agile practices in Ericsson that helps teams to achieve Agile benefits in their Globally Distributed Large Scale context, as follows:

- Requirements: The requirements are prioritized prior of being inserted into the projects. Also, the project activities are committed according to organizational capability such as budget and available resources;

- Project Planning and Monitoring: The first task of project planning is to identify pieces of system enhancements from a set of high level features and system requirements;
- Software Design: The design is divided into verifiable system changes and behaves like a software factory that frequently delivers a new component tested system version, also called the Latest System Version (LSV);
- Testing and verification: Focus is on automatization of all new test cases enabling also automatic regression test in future. The second level of tests is then done by the integration and regression test team.
- Software integration:Every week software delivery is integrated and regression tested on system level in the target environment. Regression test ensures that functionality that worked in the previous LSV is not broken. Weekly build and regression test of components is highly automated;
- Teams: organized as a cross-functional team or a component design team depending on the design environment, product complexity and product organization.

In conclusion, Large-Scale Agile Development has been used to describe agile development in everything from large teams to large multi-team projects to making use of principles of agile development in a whole organization [17].

5. RELATED WORK

Agile adoption involves much more than follow a list of prescribed practices, so the transformation can be seen from the perspective of the behavior of the team [23]. In a large-settings that agile has not been propped for, there are many difficulties and implications to take care of [15], [50]. This chapter describes studies of Agile Transformation and about Large-Scale Agile Software Development related to this work.

Relating to Agile Transformation and Maturity in Agile Development and Scaled Agile, some aforementioned works that underlying the goal of my work are present in details in the next paragraphs.

Gandomani *et. al.* [23] proposed a framework for Agile Transition and Adoption. They conducted a research with Agile Experts and collected data using Grounded Theory approach [14]. The framework developed comprises two distinct parts; structural characteristics (value based on business values, Iterative feature in software development, continuous focus on non-stop cycles of development and Gradual transition) and key activities (practice selection to a period of adaptation, adaptation, assessment, retrospective meeting to promote team collaboration, adjustment agile practices to particular contexts).

On the other hand Fontana *et. al.* [20], proposed the Progressive Outcomes framework to describe the agile software development maturing process. It is a framework in which people have the central role, ambidexterity is a key ability to maturity, and improvement is guided by outcomes agile teams pursue, instead of prescribed practices. The complex adaptive systems theory was used as a lens for analysis and four case studies were conducted to collect qualitative and quantitative data.

The Agile Fluency Model [41] was designed to follow the scale teams progress through four distinct stages that was described with a "star" system. Each star includes fluency at all previous levels that it is theoretically possible for a team to be fluent only at a particular level, or even to progress through the stages in a predictable order. The first start focus on value, then the second start focus on deliver value, the third start in optimize value and the final start focus on optimize for systems. This model was created by practitioners to practitioners.

Stojanov *et. al.* define a list of levels of maturity in Scale Agile Softawre Development using Scaled Agile Framework [practices. The SAFe Maturity Model (SAFe MM) allows assessing the level of SAFe adoption and provides a roadmap for the implementation of SAFe practices in an enterprise. The study has five levels of Maturity (Collaborative, Evolutionary, Effective, Adaptive, Encompassing) and also five principles of Scale Agile Development (Embrace Change to Deliver Customer Value; Plan and Deliver Software Frequently, Human Centricity, Technical Excellence and Customer Collaboration).

About Scaled Agile and Large-Scale Agile, their characteristics, aspects and principles it will be presented in details researches that compose the related work in this directions of define the body of knowledge about Scaled Agile and Large-Scale Agile Development.

Laanti [36] defined eight different aspects of agility, detected in large organizations: (1) Strategic Agility to reinvent the core businesses; (2) Business Agility to create a responsive organization for business benefit; (3) Agile Organization to combines Informal Networks and the Formal Organizational structure without to lose momentum; (4) People Agility to focus on change; (5) Tools Agility; (6) Organizational Agile Culture; (7) Agility of the Product that is Built and (8) Agility of payoff functions.

And he defined twenty-one principles and aspects to Scaled Agile Development, including: feedback is the fuel to learning; modular architecture increases speed; find the maximum throughput for your portfolio by balancing what can be done in parallel, and what must be done in sequence; The rate at which you are able to remove impediments of progress or service correlates with the improvement done for business; maximize the work undone. He concludes that Scaled Agility can be understood as an attempt to solve process problems other than software development on team level using agile mind-set and tools.

Dingsøyr and Moe in turn, presented a list of four aspects to Large-Scale Agile Development, described bellow and a revised research based on hot topics to the Workshop in Large-Scale Agile Development at XP Conference (including: inter-team coordination, large developments efforts, factors in scaling, architecture) [15].

- Architecture is a key role in defining how work is coordinated in large-scale development efforts;

- To have an inter-team coordination to effective knowledge networks are in large-scale development due to the knowledge-intensive nature of software development;

- The portfolio management to handle several agile projects in the same portfolio with Continuously feedback to project levels enables;

- Scaling is essential for understanding how to improve agility in Large-Scale Agile Development.

Power [54], on the other hand defined a model for when scaling agile are proper to the context of an organization. Using Human Systems Dynamics [19] as a lens to direct the research the results describes a list of aspects:

- If the system is small and relatively decoupled, even within a much larger system, and there are a low number of exchanges with other system, then the concept of “Scale Agile” likely does not apply.

- If there are a high number of systems, with a high degree of transforming exchanges between them, the organization itself could be agile. Whether approaches to scaling agile apply depends on the third scenario.
- If you have multiple system that have a high degree of transforming exchanges between them, and the agents in those systems are working toward a shared outcome, then once the number of systems and exchanges goes above a certain threshold, there is benefit in applying principles of large-scale agility. Then, it becomes increasingly important to consider system architecture at this scale.

In the same way, a taxonomy of scaled of agile software development was proposed [16] inspired by a taxonomy in requirements engineering [56].The aim of such taxonomy is to design studies in order to be more precise on selection criteria in case studies. Small-scale means one team. This team can be coordinated using agile practices to coordination and communication. The Large-scale has 2 until 9 teams. The coordination of these teams can be achieved in a new forum such as a Scrum of Scrums forum. The Very Large-scale has 10+ teams, with Several forums are needed for coordination, such as multiple Scrum of Scrums.

Other study, Brown *et. al.* [8] elaborate three principles and presents practical recommendations for achieving improved agility in large-scale software delivery enterprises. These principles are: **Economic governance** with an objective economic foundation for planning, decision-making, and progress reporting that resolves uncertainties earlier and unifies constituencies on a shared set of expected target outcomes. **Measured improvement** by quantifying change trends in the code and test base. And a **disciplined agile delivery** to ensure progress is goal-driven, using a hybrid approach to IT solution delivery.

Following the same directions, in turn, the study of Woodward *et. al.* [74] describes a combination of tree aspects (1) well-known organizational change management principles; (2) communities; and (3) an agile project management framework) in QSE IBM Quality Software Engineering that with a combination of Kotter's eight essential principles of change management, communities, and agile methods can yield consistent positive transformation results. This combination of methods addresses known inhibitors to change, allows multiple practice transformation efforts to concurrently take place, and adapts to change.

According the impacts related to adoption Paasivaara and Lassenius [50] presented a ongoing case study on adopting and scaling Scrum in a large software development project distributed across four sites. The reported showed a list of aspects that involve the scaling scrum process. The results were collected with a semi-structured interviews with project personnel of a large, global product development company, as follows:

- Area Product Owners Structure: the idea was that each product area would have a couple of development teams, and each feature would be implemented by a specific team;

- Common Sprint Planning: the user stories for the sprint were briefly presented and assigned to the teams. Afterwards, the specific teams moved to team spaces for detailed planning;
- Scrum-of-Scrums: One representative from each team participated in the daily Scrum-of-Scrums reporting the impediments the team has experienced;
- Common Sprint Demo: the program manager and PO visited each team, and the team gave a demo to them and other interested.
- Common Retrospective: In this new format the participants first brainstormed the biggest impediments and so on.

Based on the results of aforementioned literature review, to the best of my knowledge there are only a few research work that defined an agile transformation based on teams and large-scale agile software development works based on the aspects and principles related to that, but none of them discuss about how can teams acquire fluency in Large-Scale Agile Software Development. This gives me the opportunity to better explore more the topic in the context of software development given the importance of collaboration to this kind of knowledge-based work.

6. PROPOSED RESEARCH METHODOLOGY

To accomplish the posed research goal *"to characterize fluency in large-scale agile software development"* it was proposed a qualitative-based research organized in four major phases as presented in Figure 6.1. The phases will be present in the next sections.

6.1 Phase 1: Foundation

The phase 1, named *Foundation*, aims to build the foundation of the main research goal and is organized in two major sub-phases.

The *Literature Review* sub-phase will aim to review definitions for the concepts related to the study: large-scale agile development, agile transformation, agile software de-

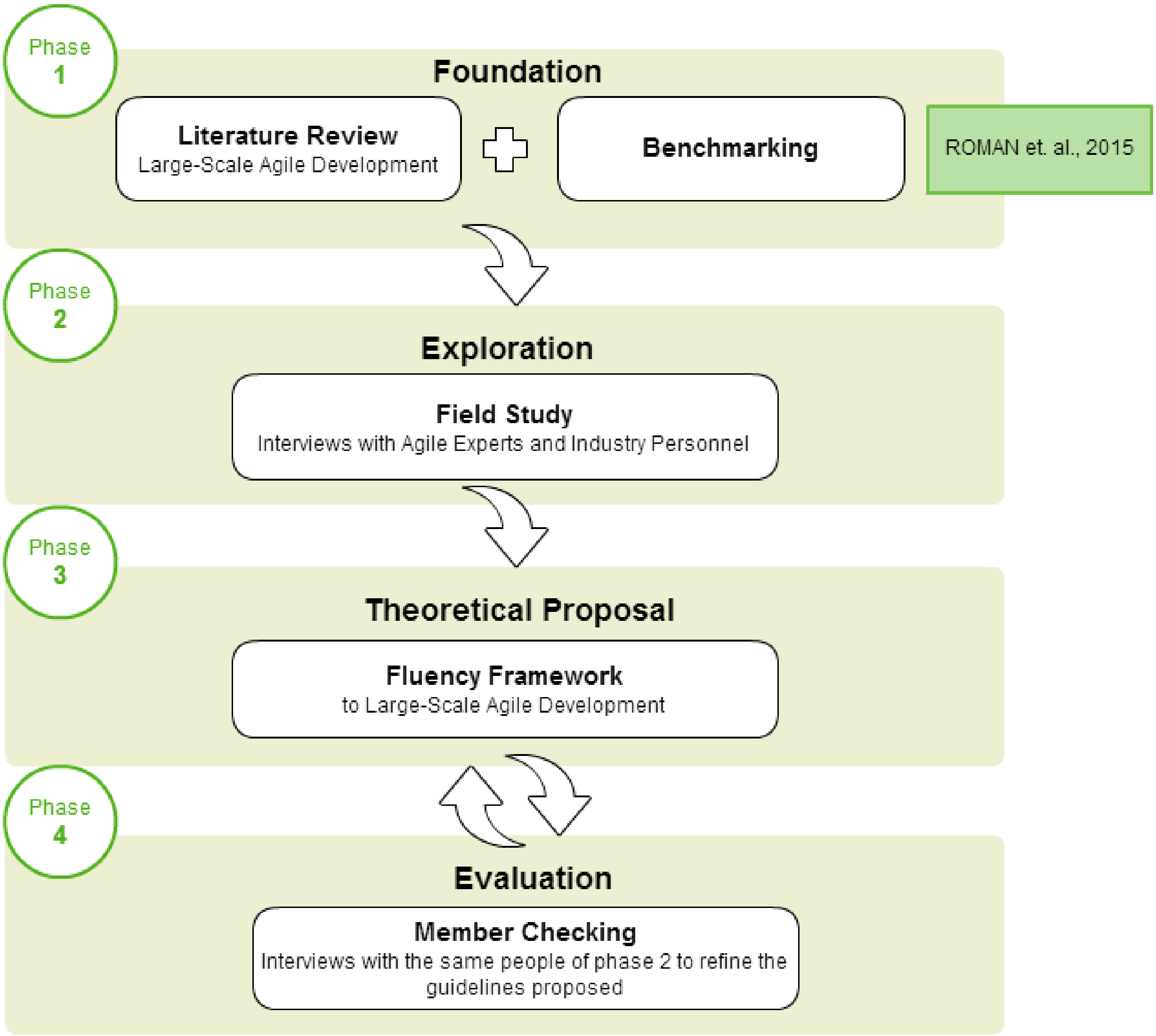

Phase 1
Foundation
Literature Review
Large-Scale Agile Development
Benchmarking
ROMAN et. al., 2015
Phase 2
Exploration
Field Study
Interviews with Agile Experts and Industry Personnel
Phase 3
Theoretical Proposal
Fluency Framework
to Large-Scale Agile Development
Phase 4
Evaluation
Member Checking
Interviews with the same people of phase 2 to refine the guidelines proposed

Figure 6.1 – Proposed research design

velopment and agile fluency. It will also serve the purpose of identifying related work and theoretical foundation that will help to better understand what has been done in such topics. This sub-phase response the Objective 1.

To this phase I will also accomplish a systematic review to understand *what are the evidences or characteristics related to large-scale agile development*, answering the Objective 2. To support this sub-phase we will follow Kitchenham [31] guidelines.

The second sub-phase *Benchmarking* aims to provide a deep empirical knowledge about how large-scale software development companies have gone through an agile transformation to become a large-scale agile company. In this sub-phase will be list initial aspects identified during the literature review sub-phase to guide the Benchmarking study and maximize our learning with this study.

6.2 Phase 2: Exploration

The phase 2, named *Exploration*, composed of a field study, aims to further the knowledge acquired in Phase 1 by observing the identified aspects in a more comprehensive manner.

The *field study* will be based on semi-structured interviews with industry personnel and agile experts who work with industry partners aiming to empirically identify what is the large-scale agile development from their perspective and which mechanisms they used or have been used to achieve fluency in Large-Scale Agile Development. Interviews will be conducted either during scheduled visits to companies within Brazil or abroad and during the attendance to conference. Candidate conference venues are:

- International Conference on Agile Software Development (XP);
- Agile Development Conference (AGILE);
- Brazilian Conference on Software Engineering (SBES);
- Agile Brazil Conference (Agile Brazil)).

6.3 Phase 3: Solution

The phase 3, named *Solution*, aims to have the Fluency Framework to large-scale agile development defined (version alpha). I will systematically organize the insights from the prior phases (Phase 1 and 2) in the format of policies or procedures that will indicate a course of actions to be taken for each of the characteristics related to the fluency that scale-teams needed.

In other words, the draft of the framework is currently composed of a set of categories related to aspects of large-scale agile development (e.g., architecture, multiple-team, complex projects) and the Fluency organized around these categories. The present practices will be identify from a review of literature and from a benchmarking work (Phase 1) and interviews with industry professionals and agile experts (Phase 2).

6.4 Phase 4: Evaluation

In Phase 4, named *Evaluation*, I will evaluate whether the proposed Fluency Framework (Phase 3) is helping companies that work in a large-scale agile development. I will conduct this phase using *member checking technique* [63]. Based on interviews with professionals to have participated in Phase 2, I aim to have them pointing out whether each of the proposed fluency guideline are proper to the aspect they mean to address and how they could be improved (to investigate the utility, quality and efficacy of the guideline applicability). Once I compile the provided feedback, I will generate a new version of the Framework (version Beta)–as part of Phase 3.

In this time, with a version Beta I will interview professionals who are currently involved in Large-Scale Agile Development and that would be willing to consider adopting my Framework. I will invite them to select a sample to try in their companies and later ask them about their perception of the results. I will also ask permission to follow the adoption and record the results. In case permission is not granted, I will ask the professionals authorization to have them interviewed later on, once they have implemented some of my policies or procedures in the Framework. They will then report their perceptions in this second round of interviews instead of having us observing the changes ourselves.

7. DETAILED RESEARCH PLAN AND CURRENT PROGRESS

The following sections present the planned activities, schedule and the research progress so far.

7.1 Planned Activities and Schedule

Below, I present the detailed plan (activities) for each of the proposed research phases that address the posed research goal and questions.

7.1.1 Phase 1: Foundation

1.1 To execute the Systematic Literature Review about the topic on Large-Scale Agile Development. This protocol will contain the pilot and approval protocol. After, I will have peer review SLR findings. And finally I will write about the SRL findings.

1.2 To execute the Literature Review about agile software development, agile transformation to investigate the best practices, occurs mainly in large-scale settings.

1.3 To design, to conduct and to analyse the Benchmarking semi-structured interviews with an Large-Scale Company in a Agile Transformation.

1.4 To consolidate the results of the two sub-phases in a single document. A document describing in detailed the findings of each phase to register the information of this entire Phase.

7.1.2 Phase 2: Exploration

2.1 To identify candidate participants. In this phase I will identify Agile Experts and Industry Personnel who are involved or have been involved with Large-Scale Agile Development. Here, I will use Conferences (XP, AGILE, Agile Brazil) to identify the correct participants of our study.

2.2 To create the first draft of the interview script. In this phase I will create the design of the script in a semi-structured way about how scale teams can obtain fluency in Large Scale Agile Development.

2.3 To pilot and review the interview script. This activity aims to validate the interview script and ensure that interviews can be accomplished. The first two interviews should be scheduled and focus on piloting the script and validating it. I will also use this activity to learn how to take notes during the interviews and improve the data collection process as a whole.

2.4 To create the final interview script version Once I have accomplished the 2.3 activity, I can generate the final version of the interview script.

2.5 To schedule and to conduct the interview. When I have the final version of the interview script, I will schedule the interviews. I will record each interview and collect the participants' authorization for the use of the data collected. And also, at the same time, after in each interview that is over, I will writing a debriefing and transcribe all of interviews.

After all, I will **2.6 analyze transcriptions** following procedures of Grounded Theory [68] to search patterns and to consolidate findings. **2.7 To consolidate data in a single document.** With all the data collected and with all the activities in this phase accomplished, I will write a document describing in detail the results from this phase.

7.1.3 Phase 3: Theoretical Proposal

3.1 To consolidate Foundation Phase and Exploration Phase. Obtaining the results collected from foundation (Phase 1) and from the exploration (Phase 2), an informal document it will be written having the consolidation of the findings about Large-Scale Agile Development and Fluency in Large-Scale Agile Development.

3.2 To structure the Guide. To consolidate findings from Phases 1 and 2 in a Guide format, a Fluency Guide. Here, I have categories and fluency characteristics extracted in the Large-Scale Agile Development. The idea is to do an easily used and understood Guide.

7.1.4 Phase 4: Evaluation

4.1 To contact previous participants. In this phase I will contact previous participants (Agile Experts and Industry Personnel) and to invite them once again to evaluate the solution.

4.2 To create the first draft of the interview script. I will create a first draft of the semi-structured interview script to evaluate the proposed framework. Also, I will review the consent document to address any specific reference to the evaluation process if necessary.

4.3 To review the interview script. This activity aims to validate the script interview and ensure that interviews can be accomplished. Following the same process as described in activity 2.3.

4.4 To create the final interview script version. Similarly to activity 2.4, I will here generate the final version of the interview script to be used in the interviews that aim to evaluate the Guide.

4.5 To schedule and to conduct the interviews When I have the final version of the interview script, I will schedule the interviews. I will record each interview and collect the participants' authorization for the use of the data collected. And also, at the same time, after in each interview that is over, I will writing a debriefing and transcribe all of interviews.

4.6 To transcript and to analyze interview transcriptions. Following procedures of Content Analysis [29] I will analyse all the transcriptions to consolidate findings.

4.7 To refine the Fluency Guide. According the findings related to interviews we will refine the proposed Fluency Guide.

4.8 To write a paper on the final results. Once I have finished the evaluation phase I will report the final version of the Guide. I intend to write a paper presenting the Guide and its development process.

7.1.5 Additional Activities and Documents

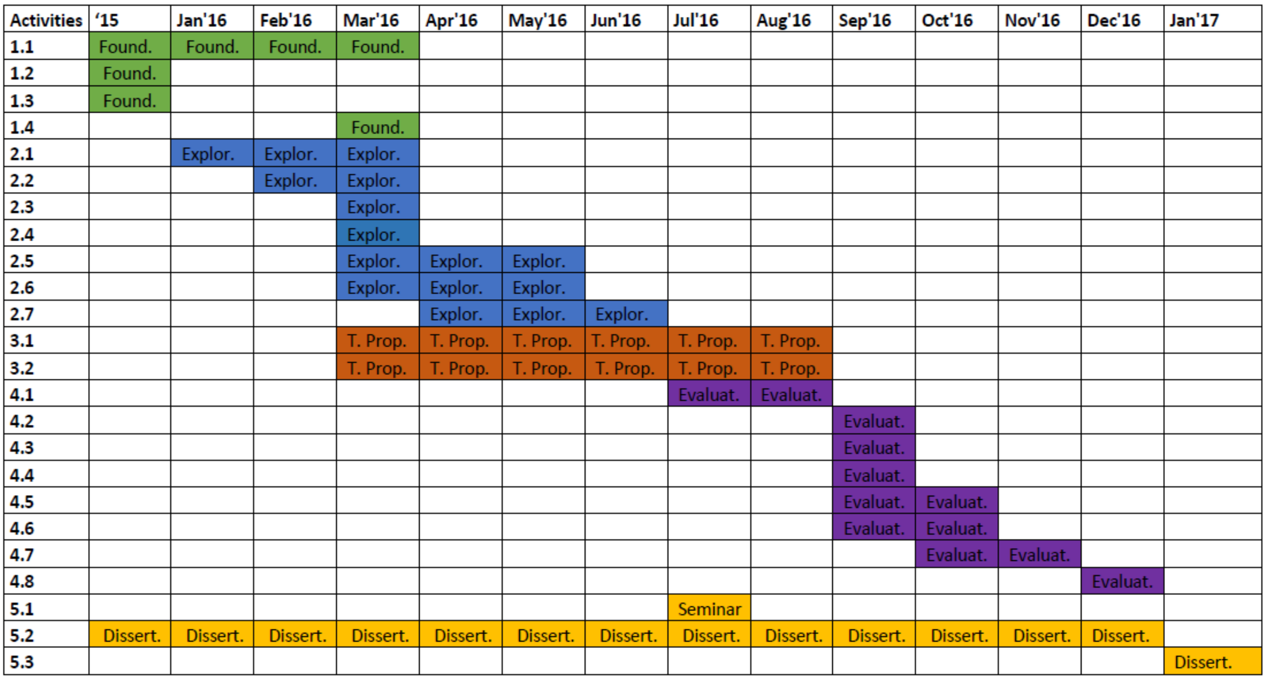
5.1 To write and to present the Seminar. The seminar is a document about the ongoing and current stage of the research. It is mandatory in the Master's program at PUCRS. I should report on Phase 1 and Phase 2, which should be completed by the expected presentation time of the Seminar as presented in the Schedule (see Table 7.1).

5.2 To write the Dissertation. The Dissertation consolidates the results of the conducted research and should be presented by the end of the Master's program. I should be writing it as presented in the Schedule (see Table 7.1).

5.3 To defend the Dissertation. I will defend the Dissertation right after writing it. The date is still to be agreed among the Supervisory Committee members and approved by the Graduate Program Coordination Committee. Table 7.1 indicates an estimated period for the defense.

7.1.6 Planned Schedule

Table 7.1 presents the planned schedule for the activities listed above, including the additional documents.



Activities	‘15	Jan'16	Feb'16	Mar'16	Apr'16	May'16	Jun'16	Jul'16	Aug'16	Sep'16	Oct'16	Nov'16	Dec'16	Jan'17
1.1	Found.	Found.	Found.	Found.										
1.2	Found.													
1.3	Found.													
1.4				Found.										
2.1		Explor.	Explor.	Explor.										
2.2			Explor.	Explor.										
2.3				Explor.										
2.4				Explor.										
2.5				Explor.	Explor.	Explor.								
2.6				Explor.	Explor.	Explor.								
2.7					Explor.	Explor.	Explor.							
3.1				T. Prop.	T. Prop.	T. Prop.	T. Prop.	T. Prop.	T. Prop.					
3.2				T. Prop.	T. Prop.	T. Prop.	T. Prop.	T. Prop.	T. Prop.					
4.1								Evaluat.	Evaluat.					
4.2										Evaluat.				
4.3										Evaluat.				
4.4										Evaluat.				
4.5										Evaluat.	Evaluat.			
4.6										Evaluat.	Evaluat.			
4.7											Evaluat.	Evaluat.		
4.8													Evaluat.	
5.1								Seminar						
5.2	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	Dissert.	
5.3														Dissert.

Table 7.1 – Planned Schedule for the Proposed Activities

7.2 Current Progress

The proposed research in this document have been discussed since May 2015. First, the research was developed thought literature review about Agile Transformation. Hence the knowledge was consolidated, then I submitted a paper at Second Latin American School of Software Engineering to discuss about the topic and to review the first research plan of my study. The camera-ready copy of the paper is available in Attachment A.

Meanwhile I planned interviews that support the benchmarking study. The first draft of the script interview was conducted and this reviewed by two researchers in order to adaptations. While the adjustments were made, I schedule the participants to interviews. These participants are staff of a Large-Complex Globally Distributed company that was in an initial transformation to Agile Development. This period lasted two months.

After, the interviews were conducted in two months. During the interviews I take-notes and after each interview it was made the transcription. Eighteen interviews were conducted and analyzed using techniques of Grounded Theory [14]. At the end, a paper has been submitted to dissemination of results in the Brazilian Workshop of Agile Methods. The camera-ready copy of the paper is available in Attachment B.

Also, while the writing of this document, I was started the systematic protocol of large-scale agile development (available in Attachment C). I define the protocol after I read a few papers related to large-scale agile development.

During my presentation in Agile Brazil of last year (2015) I did network with Industry Personnel and Agile Experts that may be the contacts to the Field Study (Phase 2).

REFERENCES

[1] Alzoubi, Y. I.; Gill, A. Q.; Al-Ani, A. “Empirical studies of geographically distributed agile development communication challenges: A systematic review”, *Information and Management*, vol. 53–1, 2016, pp. 22–37.

[2] Ambler, S. “Agile software development at scale”. In: *Balancing Agility and Formalism in Software Engineering*, Meyer, B.; Nawrocki, J.; Walter, B. (Editors), Springer Berlin Heidelberg, 2008, pp. 1–12.

[3] Ambler, S. W. “Ibm agility@ scale™: become as agile as you can be”. Available at: https://www.ibm.com/developerworks/community/blogs/ambler?lang=en, Nov 2015.

[4] Ambler, S. W.; Lines, M. “Disciplined agile delivery: A practitioner’s guide to agile software delivery in the enterprise, First Edition”. IBM Press, 2012.

[5] Atlas, A. “Accidental adoption: The story of scrum at amazon. com”. In: Agile Conference, 2009, pp. 135–140.

[6] Beck, K.; colleagues. “Manifesto for agile software development”. Available at: http://www.agilemanifesto.org/, Nov 2015.

[7] Benefield, G. “Rolling out agile in a large enterprise”. In: Proocedings of the Hawaii International Conference on System Sciences, 2008, pp. 461–.

[8] Brown, A. W.; Ambler, S.; Royce, W. “Agility at scale: economic governance, measured improvement, and disciplined delivery”. In: Proceedings of the International Conference on Software Engineering, 2013, pp. 873–881.

[9] Christopher, M. “The agile supply chain: competing in volatile markets”, *Industrial marketing management*, vol. 29–1, 2000, pp. 37–44.

[10] Cloke, G. “Get your agile freak on! agile adoption at yahoo! music”. In: Proceedings of the Agile Conference, 2007, pp. 240–248.

[11] Cockburn, A. “Agile Software Development, First Edition”. Addison-Wesley, 2002, 278p.

[12] Cockburn, A. “Crystal clear: a human-powered methodology for small teams, First Edition”. Pearson Education, 2004, 336p.

[13] Cockburn, A.; Highsmith, J. “Agile software development: The people factor”, *Computer*, –11, 2001, pp. 131–133.

[14] Corbin, J.; Strauss, A. “Basics of qualitative research: Techniques and procedures for developing grounded theory, Fourth Edition”. Thousand Oaks, CA: Sage Publications, Inc., 2014, 456p.

[15] Dingsøyr, T.; Moe, N. B. “Research challenges in large-scale agile software development”, *ACM SIGSOFT Software Engineering Notes*, vol. 38–5, 2013, pp. 38–39.

[16] Dingsøyr, T.; Fægri, T.; Itkonen, J. “What is large in large-scale? a taxonomy of scale for agile software development”. In: *Product-Focused Software Process Improvement*, Jedlitschka, A.; Kuvaja, P.; Kuhrmann, M.; Männistö, T.; Münch, J.; Raatikainen, M. (Editors), Springer International Publishing, 2014, pp. 273–276.

[17] Dingsøyr, T.; Moe, N. B. “Towards principles of large-scale agile development”. In: Proceedings of Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation, 2014, pp. 1–8.

[18] Elshamy, A.; Elssamadisy, A. “Applying agile to large projects: new agile software development practices for large projects”. In: *Proceedings of the Agile Processes in Software Engineering and Extreme Programming*, Berlin, Heidelberg: Springer, 2007, pp. 46–53.

[19] Eoyang, G. H. “Conditions for self-organizing in human systems”, Ph.D. Thesis, The Union Institute, Circle Pines, MN, 2001, 198p.

[20] Fontana, R.; Meyer, V.; Reinehr, S.; Malucell, A. “Progressive outcomes: A framework for maturing in agile software development”, *Journal of Systems and Software*, vol. 102, 2015, pp. 88–108.

[21] Fry, C.; Greene, S. “Large scale agile transformation in an on-demand world”. In: Proceedings of the Agile Conference, 2007, pp. 136–142.

[22] Gandomani, T.; Zulzalil, H.; Nafchi, M. “Agile transformation: What is it about?” In: Malaysian Software Engineering Conference, 2014, pp. 240–245.

[23] Gandomani, T. J.; Nafchi, M. Z. “An empirically-developed framework for agile transition and adoption: A grounded theory approach”, *Journal of Systems and Software*, vol. 107, 2015, pp. 204–219.

[24] Gandomani, T. J.; Zulzalil, H.; Ghani, A. A. A.; Sultan, A. B. M.; Sharif, K. Y. “How human aspects impress agile software development transition and adoption”, *International Journal of Software Engineering and Its Applications*, vol. 8–1, 2014, pp. 129–148.

[25] Ganesh, N.; Thangasamy, S. “Lessons learned in transforming from traditional to agile development”, *Journal of Computer Science*, vol. 8, 2012, pp. 389–392.

[26] Gat, I. “How bmc is scaling agile development”. In: Agile Conference, 2006, 2006, pp. 6–pp.

[27] Gruver, G.; Young, M.; Fulghum, P. “A practical approach to large-scale agile development: how HP transformed LaserJet FutureSmart firmware, First Edition”. Addison-Wesley Professional, 2013, 208p.

[28] Heikkilä, V.; Rautiainen, K.; Jansen, S. “A revelatory case study on scaling agile release planning”. In: EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2010, pp. 289–296.

[29] Hsieh, H.-F.; Shannon, S. E. “Three approaches to qualitative content analysis”, *Qualitative health research*, vol. 15–9, 2005, pp. 1277–1288.

[30] Ivček, M.; Galinac, T. “Adapting agile practices in globally distributed large scale software development”. In: Proceedings of the International Conference on Telecommunications and Information of the International Convention MIPRO, 2009.

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

[32] Koehnemann, H.; Coats, M. “Experiences applying agile practices to large systems”. In: Proceedings of the Agile Conference, 2009, pp. 295–300.

[33] Koehnemann, H.; Coats, M. “Experiences applying agile practices to large systems”. In: Proceedings of the Agile Conference, 2009, pp. 295–300.

[34] Korhonen, K. “Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context”, *Software Quality Journal*, vol. 21–4, 2013, pp. 599–624.

[35] Laanti, M. “Agile transformation study at nokia – one year after”. In: *Lean Enterprise Software and Systems*, Abrahamsson, P.; Oza, N. (Editors), Springer Berlin Heidelberg, 2010, *Lecture Notes in Business Information Processing*, vol. 65, pp. 3–19.

[36] Laanti, M. “Characteristics and principles of scaled agile”. In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Springer, 2014, pp. 9–20.

[37] Laanti, M.; Salo, O.; Abrahamsson, P. “Agile methods rapidly replacing traditional methods at nokia: A survey of opinions on agile transformation”, *Inf. Softw. Technol.*, vol. 53–3, March 2011, pp. 276–290.

[38] Larman, C.; Vodde, B. “Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum, First Edition”. Addison-Wesley Professional, 2008, 368p.

[39] Larman, C.; Vodde, B. “Practices for Scaling Lean and Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum, First Edition”. Addison-Wesley Professional, 2010, 624p.

[40] Larman, C.; Vodde, B. “Large-Scale Scrum: More with LeSS, First Edition”. Addison-Wesley Professional, 2016, 272p.

[41] Larsen, D.; Shore, J. “Your path through agile fluency”. Available at: http://agilefluency.com/, Nov 2015.

[42] Leffingwell, D. “Scaling Software Agility: Best Practices for Large Enterprises, First Edition”. Addison-Wesley Professional, 2007, 384p.

[43] Leffingwell, D. “Agile software requirements: lean requirements practices for teams, programs, and the enterprise, First Edition”. Addison-Wesley Professional, 2010.

[44] Lindvall, M.; Muthig, D.; Dagnino, A.; Wallin, C.; Stupperich, M.; Kiefer, D.; May, J.; Kahkonen, T. “Agile software development in large organizations”, *Computer*, vol. 37–12, Dec 2004, pp. 26–34.

[45] Lyon, R.; Evans, M. “Scaling up pushing scrum out of its comfort zone”. In: Agile, 2008. AGILE’08. Conference, 2008, pp. 395–400.

[46] Madden, S. “Versionone agile made easier”. Available at: https://www.versionone.com/about/, Dec 2015.

[47] Nerur, S.; Mahapatra, R.; Mangalaraj, G. “Challenges of migrating to agile methodologies”, *Communications of the ACM*, vol. 48–5, 2005, pp. 72–78.

[48] Nord, R.; Ozkaya, I.; Kruchten, P. “Agile in distress: Architecture to the rescue”. In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Dingsøyr, T.; Moe, N.; Tonelli, R.; Counsell, S.; Gencel, C.; Petersen, K. (Editors), Rome, IT: Springer International Publishing, 2014, pp. 43–57.

[49] Paasivaara, M.; Heikkilä, V. T.; Lassenius, C. “Experiences in scaling the product owner role in large-scale globally distributed scrum”. In: International Conference on Global Software Engineering, 2012, pp. 174–178.

[50] Paasivaara, M.; Lassenius, C. “Scaling scrum in a large distributed project”. In: International Symposium on Empirical Software Engineering and Measurement (ESEM), 2011, pp. 363–367.

[51] Paasivaara, M.; Lassenius, C. “Communities of practice in a large distributed agile software development organization–case ericsson”, *Information and Software Technology*, vol. 56–12, 2014, pp. 1556–1577.

[52] Papadopoulos, G. “Moving from traditional to agile software development methodologies also on large, distributed projects.”, *Procedia-Social and Behavioral Sciences*, vol. 175, 2015, pp. 455–463.

[53] Poon, D. “A self funding agile transformation”, 2006.

[54] Power, K. “A model for understanding when scaling agile is appropriate in large organizations”. In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Dingsøyr, T.; Moe, N.; Tonelli, R.; Counsell, S.; Gencel, C.; Petersen, K. (Editors), Rome, IT: Springer International Publishing, 2014, vol. 199, chap. Lecture Notes in Business Information Processing, pp. 83–92.

[55] Ranganath, P. “Elevating teams from ’doing’ agile to ’being’ and ’living’ agile”. In: Agile Conference, 2011, pp. 187–194.

[56] Regnell, B.; Svensson, R.; Wnuk, K. “Can we beat the complexity of very large-scale requirements engineering?” In: *Requirements Engineering: Foundation for Software Quality*, Paech, B.; Rolland, C. (Editors), Springer Berlin Heidelberg, 2008, vol. 5025, chap. Requirements Engineering: Foundation for Software Quality, pp. 123–128.

[57] Rising, L. “The power of an agile mindset”. Available at: http://www.infoq.com/presentations/agile-mindset, Sept 2014.

[58] Rising, L.; Janoff, N. S. “The scrum software development process for small teams”, *IEEE software*, vol. 17–4, 2000, pp. 26–32.

[59] Rolland, K. H. “’desperately’ seeking research on agile requirements in the context of large-scale agile projects”. In: Proceedings of the XP, 2015, pp. 5:1–5:6.

[60] Schatz, B.; Abdelshafi, I. “Primavera gets agile: a successful transition to agile development”, *IEEE software*, vol. 22–3, 2005, pp. 36–42.

[61] Schiel, J. “Enterprise-Scale Agile Software Development Applied Software Engineering Series, First Edition”. CRC Press, 2009, 382p.

[62] Schweigert, T.; Nevalainen, R.; Vohwinkel, D.; Korsaa, M.; Biro, M. “Agile maturity model: Oxymoron or the next level of understanding”. In: *Software Process Improvement and Capability Determination*, Mas, A.; Mesquida, A.; Rout, T.; O’Connor, R.; Dorling, A. (Editors), Springer Berlin Heidelberg, 2012, vol. 290, pp. 289–294.

[63] Seaman, C. B. “Qualitative methods in empirical studies of software engineering”, *IEEE Trans. Software Eng.*, vol. 25–4, 1999, pp. 557–572.

[64] Sidky, A. “A structured approach to adopting agile practices: The agile adoption framework”, Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2007, 337p.

[65] Sommerville, I. “Software Engineering, Ninth Edition”. Boston, MA: Pearson, 2010, 792p.

[66] Soundararajan, S.; Arthur, J. “A structured framework for assessing the "goodness" of agile methods”, April 2011.

[67] Stojanov, I.; Turetken, O.; Trienekens, J. J. “A maturity model for scaling agile development”. In: Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2015, pp. 446–453.

[68] Strauss, A. L.; Corbin, J. M. “Basics of qualitative research, Second Edition”. SAGE Publications, Inc, 1990, vol. 15, 272p.

[69] Sutherland, J. “Agile can scale: Inventing and reinventing scrum in five companies”, *Cutter IT journal*, vol. 14–12, 2001, pp. 5–11.

[70] Sutherland, J. “Future of scrum: Parallel pipelining of sprints in complex projects”. In: Proceedings of the Agile Conference, 2005, pp. 90–99.

[71] Therrien, I.; LeBel, E. “From anarchy to sustainable development: Scrum in less than ideal conditions”. In: Proceedings of the Agile Conference, 2009, pp. 289–294.

[72] Van Waardenburg, G.; Van Vliet, H. “When agile meets the enterprise”, *Information and software technology*, vol. 55–12, 2013, pp. 2154–2171.

[73] VersionOne, I. “9th annual state of agile™ survey”. Available at: http://stateofagile.versionone.com/, Dec 2015.

[74] Woodward, E. V.; Bowers, R.; Thio, V. S.; Johnson, K.; Srihari, M.; Bracht, C. J. “Agile methods for software practice transformation”, *IBM Journal of Research and Development*, vol. 54–2, March 2010, pp. 3:1–3:12.

ATTACHMENT A – Paper Accepted at the ELA-ES 2015

On the Transformation to Agile in a Large-Complex Globally Distributed Company: A Research Plan to Define Guidelines

Greice Roman, Sabrina Marczak, Alessandra Dutra

[1]Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681 – Partenon – 90.619-900 – Porto Alegre – RS – Brazil

greice.roman@acad.pucrs.br, {sabrina.marczak, alessandra.dutra}@pucrs.br

Abstract. *The transformation to agile is not a simple process and although there is vast literature on the topic, there is still no consolidated body of knowledge on how to proceed when this transformation happens in large-complex globally distributed companies. This paper presents the research plan to follow the transformation into agile of a large-complex distributed IT organization aiming to serve as an exploratory case study for our long-term goal of proposing a set of guidelines to guide the transformation in such type of company.*

1. Introduction

The Agile Manifesto [Beck and colleagues 2001] was written in February 2001. It offers new values to motivate software companies to deliver high-quality products faster and produce satisfied customers. Since then, companies are discussing whether to become agile and how to go about transforming themselves to achieve such 'agility'. The transformation process involves more than deciding on which agile method to adopt. It refers to making changes in such a way that the company and its projects will 'become' agile.

Academia has been supporting industry to go through the transformation process for as long as the agile philosophy is out there. However, achieving success in large-scale companies as reported in [Fry and Greene 2007]), for example, is a complex process and brings numerous challenges to organizations [Korhonen 2013]. For instance, how much can requirements keep changing when they cross hundreds of applications at a time? [Dingsøyr and Moe 2014] consolidated a research agenda on the topic showing that there are still several open questions. Given the significant number of large companies migrating to this philosophy, there is a need for a consolidated and more extensive body of knowledge.

This paper presents the research plan to follow the transformation into agile of a large-complex distributed IT organization aiming to serve as an exploratory case study for our long-term goal of proposing a set of guidelines to guide the agile transformation process in large-complex globally distributed software companies. We will contribute to furthering the body of knowledge on the topic for this type of company.

2. Agile Transformation

The agile transformation process (ATP) has been encouraged to remedy inherent problems of traditional software development [Gandomani et al. 2014] and is defined as *the process of leaving the traditional way to development software and adopting the agile philosophy, tools, and principles* [Ranganath 2011]. A true ATP must focus on 'being' agile rather than 'doing' agile. This is the main reason that makes ATP more difficult than expected.

[Gandomani et al. 2014] identified a set of categories of an ATP, as follows: prerequisites to become agile, training on methods and what it is about, facilitators (people who will guide the process), transition framework (a stepwise view on how to do it), managing the transition, assessment of progress, reasons for aiming to agility, coaching the transition, technical issues, human aspects-related issues, customer-related issues, selection of pilot projects for the transition, and agile method selection. [Fontana et al. 2015], on the other hand, identified the following categories: practices to become agile, team composition and behavior, deliveries (evolution from traditional to continuous delivery), requirements (transition from traditional requirements elicitation to use stories), product (practices to improve the software product), and customer relationship.

In large-scale agile transformation, the key challenges seem to be managing a large number of agile teams, dividing work among those teams, achieving the system-wide properties of the software, and guaranteeing the simultaneous releases of cross-cutting features [Ganesh and Thangasamy 2012]. In their study, [Dingsøyr and Moe 2014] identified four principles to be observed in large-scale agile development, namely: *architecture* - figuring out how work is coordinated; *inter-team coordination* - creating effective knowledge networks is essential due to the knowledge-intensive nature of software development; *portfolio management* - providing continuous feedback from the portfolio to project levels enables the teams and project members to take decisions that are consistent with the goals of the large-scale agile portfolio, and *scaling* - describing the context for agility and scale is essential for understanding how to improve agility in large-scale agile.

The discussion becomes even more interesting when a large company is physically distributed and develops complex-interrelated applications.

3. Our Long-Term Research Plan and Case Study Design

To achieve our research goal we will follow a qualitative approach organized in four major phases as presented in Figure 1. Phase 1, named *Foundation*, aims to build the foundation for our two-year long investigation and is organized in two major sub-phases. The *Literature Review* sub-phase will aim to review definitions for the concepts related to our study such as agile transformation, large-scale development, development of complex applications, and global software development, and define the concepts we will adopt. It will also serve the purpose of identifying related work that will help us better understand what has been done and refine the research scope. As a result, we expect to identify an initial list of aspects that have to be considered when going through the referred transformation (period: Jan to May'15). The *Benchmarking* sub-phase aims to provide us with a deeper empirical knowledge about how large-scale complex globally distributed companies have gone through the ATP themselves. We will visit companies in Europe who have gone through this process and discuss the aspects identified during the Literature Review with them. We plan two months of work including visits and data analysis (period: Jun-Jul'15).

Phase 2, named *Exploration*, composed of a case study, aims to further the knowledge acquired in Phase 1 by observing the identified aspects in a more comprehensive manner (period: Aug-Dec'15). Section 3.1 presents the case study design in more details. Phase 3, named *Solution*, aims to have our guidelines for agile transformation in large-scale complex globally distributed companies defined (version alpha). We will systematically organize the insights from the prior phases in the format of policies or procedures

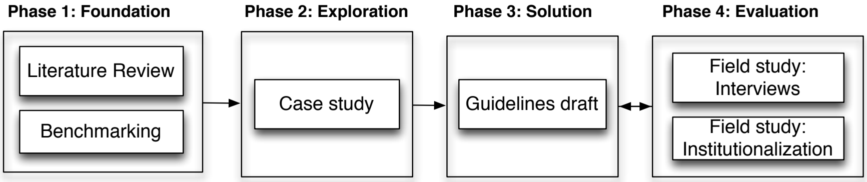


Figure 1. Proposed research design

that will indicate a course of actions to be taken for each of the aspects related to the transformation when a company wants to become agile (period: Jan-Apr'16).

In Phase 4, named *Evaluation*, we will evaluate whether our proposed guidelines document is fit to help large-scale complex globally distributed companies to become agile. We will conduct a field study composed of semi-structured interviews with professionals who have been involved with agile transformation in the context we are interested in aiming to have them pointing out whether each of the proposed guidelines are proper to the aspect they mean to address and how they could be improved (period: May-Jul'16). Once we compile the provided feedback, we will generate a new version of the guidelines document (version beta)–as part of Phase 3 (period: Aug'16), and then consult a new set of professionals to evaluate this version–as part of Phase 4 again (period: Set-Dec'16). This time we will interview professionals who are currently involved in the transformation process and that would be willing to consider adopting our guidelines. We will ask them what they think of the guidelines and will invite them to select a sample to try in their companies and later ask them about their perception of the results. There is a risk that guidelines will not be institutionalized given the short time, however, we still think that having a preliminary try to observe changes and results is valuable to have a more refined feedback on how fit is our set of guidelines.

3.1. Case Study Background and Design

The case study will be investigate in a large IT multinational company chosen by convenience. The company, named ORG (fictitious name), has started its agile transformation in Jan'15. ORG's IT department develops software products to support the organization' business processes. Demands to develop or to update these products come from the several business departments distributed around the world. The IT department is organized by business area. Each IT team attends a business area only and is composed of the following functions: project management, business and requirements analysts, developers, and architects. The Test team is a separate organizational unit and has its members allocated per project by business area as requested by IT teams. IT personnel are distributed among the headquarters' office located in the US and also in Brazil, India, and Malaysia.

Projects are defined once a year as part of the IT roadmap plan. Once approved, projects are assigned to project managers who allocate their teams and start a discussion with business representatives acting as project owner proxies of what features should be developed first. A Sprint plan is then approved and development starts. A project backlog is kept and rediscussed each iteration until the scope is finished. Each project might tackle business requests that might cross software applications, presenting the organization with the challenge of having to manage the complexity of large systems interlocks.

Each business IT team is free to organize itself as it wants as long as the team respects the worldwide guidelines defined by the IT board, such as adopt the tools to support software development defined by the organization and follow agile practices proposed by Scrum and XP only. Managers anticipate that this 'freedom' will increase the complexity of coordinating software development cross business areas and system interlocks.

We will observe four projects selected as pilots by the organization for the ATP. These projects are receiving training and guidance from a company specialized in Agile Transformation. Each project belongs to a distinct business area and are distributed in at least two sites each. We will be present at the Brazilian site but we also got permission to participate in virtual meetings and conference calls with remote team members.

We aim to gather data about how the teams are organizing themselves and why such organization, which practices they are adopting and why, which issues they are going through and how they are solving them, and which needs they have that require feedback from senior management. We will take into consideration the work environment they have (e.g., applications and team members background, imposed organizational guidelines, etc) and collect the perception they have on how such environment influences the transformation. Data will be collected and analyzed simultaneously to allow for follow-up clarifications and exploration of new insights coming from the observed data.

4. Final Remarks

This paper presents our long-term goal to define a set of guidelines to support the ATP in large-complex globally distributed companies. We discuss in more details one of the four research phases, Phase 2, composed of a case study, highlighting the importance to collect empirical evidence in this kind of study. Although we have just recently started our investigation, we have already identified a set of aspects that have to be observed during the ATP from literature. We expect that our findings will shed some light in understanding the phenomena and providing guidance to companies who want to go through this process.

References

Beck, K. and colleagues (2001). Manifesto for agile software development.

Dingsøyr, T. and Moe, N. (2014). Towards principles of large-scale agile development. In Dingsøyr, T., Moe, N., Tonelli, R., Counsell, S., Gencel, C., and Petersen, K., editors, *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, volume 199 of *Lecture Notes in Business Information Processing*, pages 1–8. Springer.

Fontana, R., Meyer, V., Reinehr, S., and Malucelli, A. (2015). Progressive outcomes: A framework for maturing in agile sw development. *J. of Systems and Sw*, 102:88–108.

Gandomani, T., Zulzalil, H., and Nafchi, M. (2014). Agile transformation: What is it about? In *Proc. Malaysian SEng. Conf., 2014, Langkawi, Malaysia*, pages 240–245.

Ganesh, N. and Thangasamy, S. (2012). Lessons learned in transforming from traditional to agile development. *Journal of Computer Science*, 8:389–392.

Korhonen, K. (2013). Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. *Software Quality Journal*, 21:599–624.

Ranganath, P. (2011). Elevating teams from 'doing' agile to 'being' and 'living' agile. In *Proceedings of the Agile Conference, Aug 2011, Salt Lake City, USA*, pages 187–194.

ATTACHMENT B – Paper Accepted at the WBMA 2015

On the Agile Transformation in a Large-Complex Globally Distributed Company: Why Boarding this Journey, Steps Taken, and Main Foreseen Concerns

Greice Roman, Sabrina Marczak, Alessandra Dutra, Rafael Prikladnicki
MunDDos Research Group - Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, RS - Brazil
greice.roman@acad.pucrs.br, {sabrina.marczak, alessandra.dutra, rafael.prikladnicki}@pucrs.br

***Abstract*—ORG, fictitious name, is a large company with complex-globally distributed projects that has recently started a ”big-bang” transition to agile. About a year ago the new CIO announced it was time to aim for faster deliveries and decided on a company-wide roll-out to agile, including legacy systems. In this paper we report on an interview-based qualitative study that aimed to identify the reasons of why the company is moving to an agile approach, the steps towards becoming agile took during this first year of work, and the main concerns of senior management given the size of the company, the complexity of the projects developed, and the team's global distribution. We interviewed 18 senior managers, including members of the board committee that assess the CIO in strategic decisions. Our findings add to the current literature on the topic by discussing the transformation in a large-complex scenario that, to the best of our knowledge, has never been reported in literature.**

I. Introduction

Companies need to constantly improve their processes and revisit decisions made in order to keep up with new industrial paradigms such as globalization and time-to-market. Almost 15 years ago, new values such as frequent deliveries to the customer and agility have been foreseen as key to companies survive in a market that started to require rapid and flexible responses to change. The Agile Manifesto [1] was then born and agile development has set foot in Software Engineering.

In the agile era, companies that had previously been following an hierarchical organizational structure and well-defined processes based on quality models such as CMMI, have to make several decisions and reinvent themselves if they wish to stay competitive. The decision-making process to become agile is not simple and it involves much more than knowing ”how to do” and ”what to do” [2]. The core of this transformation, defined as *the process of leaving the traditional way to development software and adopting the agile philosophy, tools, and principles* [3], is in changing organizational culture and people's behavior and as such it requires organizational alignment among all departments and support from senior management.

Despite successful stories of large-scale companies that has been migrated to agile (e.g., [4]), literature reports that this complex process brings numerous challenges to the organizations [5]. Therefore, it is important to be well-aware of the reasons of why going through such a change, to understand the ”day after day” of this transformation to realize how puzzling it is to ”acquire” such new culture, and to be as familiar as possible of the potential pitfalls ahead.

This paper presents a report of the initial steps of the company-wise transformation at ORG, a large company with complex-globally distributed projects. In our interview-based qualitative study we sought *to understand what are the main reasons that motivated ORG to enter this journey, to identify the steps performed so far to implement the transformation and aim to become agile, and to name the majors concerns foreseen by seniors management to succeed in such endeavor.*

Our main findings from our 18 interviews with senior managers located in the USA and in Brazil are: reasons for going agile are aiming to reduce delivery time to customers and to reduce the communication gap between business and IT departments; a major reorganization including redefining roles and responsibilities and the command-chain, and training people on agile practices and ceremonies were the main first actions taken to prepare the organization for the change; and the complex ecosystem of applications and their dependencies as well as the teams' global distribution with no or little overlapping working hours are among the most cited concerns by the participants. We discuss these findings in light to current literature and by considering the company's background.

II. Company Background

ORG (fictitious name) is a large IT multinational company with offices located over 5 continents. Software products to support the organizational processes are developed by the IT department. Demands to develop or to update these products come from the business departments, mainly located in the headquarters' office in the USA but with business representatives spread out over 30 countries. IT development teams are distributed among the headquarters office and in Brazil, India, and Malaysia. There also also IT employees in China, Japan, Ireland, and Russia.

The IT department, at the beginning of the transformation initiative, used to follow a matrix structure based on business areas (e.g., sales) and IT functions (e.g., developers). Development assignments were mainly organized in projects that vary from the development of new products to the maintenance of legacy systems, and project teams would mainly follow the waterfall model. Some Scrum practices were scarcely adopted in a project-based fashion to mainly support project management. Software development processes would vary from formal (following CMMI Level 3 practices) to informal (defined by the project members upon their needs).

A well-known practice at ORG is still in place: an annual project roadmap is defined in December based on the requests made by business representatives and recorded by business analysts. Business analyst managers in conjunction with project managers prioritize the requests and define a set of projects to be developed throughout the year. Priorities are defined based on business impact and on development costs, and approved by a committee board composed of senior business and IT personnel who directly report to the CIO board.

Distributed software teams were then formed to develop the elected projects. Members were assigned to projects based on their skills and domain knowledge, despite of their physical location. Therefore, a project often has its roles distributed over several locations. By mid February each team would receive a business request document. The software team would start working to translate the business into software requirements led by the software requirements analysts. These have to consult with business analysts to clarify business requirements and, when necessary, business representatives are invited to join the discussion. Project managers would monitor the project progress based on a set of organizational performance measures that would be reported to senior management in a regular basis. Results from these measurements were used to determine whether a project failed, attended, or exceeded its performance goals. Although ORG has gone through a major reorganization, this roadmap assignment process has not been reconsidered yet as mentioned in Sections IV and V.

III. Method

To attend our, we conducted an interview-based qualitative study. The semi-structured interviews were conducted in person on-site at the American and Brazilian IT offices. The interviews with the 10 American-based representatives took place on the fourth month after the agile transformation process kick-off (Dec'14) and was conducted by one of the researchers while the 8 remaining ones with the Brazilian-based representatives took place during the ninth and tenth months (May and Jun'15) and were conducted by two other researchers.

The interview was focused in 4 major themes as follows: *participant's background*, we asked the participant to introduce herself and talk about her job description and responsibilities within the company; *reasons for the transformation*, we asked her to elaborate on the reasons the company is going through such transformation that she was aware of; *actions taken*, we invited the participant to introduce us to the activities she was engaged on or has been communicated about related to the transformation; and *concerns*, we requested the participant to express her concerns about the transformation process as a whole considering the organization background.

Participants were pointed out by a senior manager, the focal point of the research project that this study is part of, and then invited to participate on a volunteer basis. All 19 participants accepted our invitation; however, a US-based member called out in sick during our visit to the USA office and had the interview canceled. Thus, we interviewed 18 participants.

TABLE I. Participants' Profile

ID	Job Title
US1	Senior Manager on Process Improvement, Member of the CIO Board
US2	Senior Quality Assurance Analyst (Process-based quality)
US3	Senior Director on Process Improvement, Head of the CIO Board Committee and of the World-wide Agile Transformation Initiative
US4	Senior Manager on Quality Management (Product-based quality), Member of the CIO Board
US5	Senior Business Analyst Team Leader
US6	Senior Business Analyst, Leader of the World-wide Business Analyst Community of Practice
US7	Senior Business Analyst Manager
US8	Senior Portfolio Planning Manager, Member of the CIO Board
US9	Senior Portfolio Planning Manager, Head of the Roadmap Planning Department, Member of the CIO Board
US10	Senior Director on Finance Application Development area
BR1	Senior Manager on Services Application Development area
BR2	Director on Infrastructure area
BR3	Senior Manager on Finance Application Development area
BR4	Senior Manager on Services Application Development area
BR5	Senior Manager on Financial Services Application Development area
BR6	Director on Infrastructure area
BR7	Director on Services Application Development area
BR8	Manager on Financial Services Application Development area

All participants are senior managers at ORG and are either a member of the CIO committee board, a senior director, or a senior development manager. They are working at ORG for at least 4 years and at most for 16 years. All participants are currently a member of the IT department but 5 of the US-based representatives had worked in the business department either as a salesman, a manufacturing manager, or a business analyst. Also, out of our pool of respondents, 8 of them are male (1 in the USA, 7 in Brazil) and 10 are female (9 in the USA, 1 in Brazil). Table I summarizes the participants' job position.

All interviews were voice recorded and later transcribed by each one of the interviewees. The shortest interview lasted 20 minutes and the longest 90 minutes, and they took in average 53 minutes. Our subsequent analysis was guided by grounded theory procedures [6]. We coded the interviews identifying factors for each of our topics of interest–reasons for the transformation, actions taken, and concerns–until we have exhausted an understanding and reached a final set of merged factors. Coding was done by each of the interviewees and later reviewed by an independent senior researcher that is part of our research project. Next, codes were merged into a single document.

IV. The Transformation First Steps

In this section we present the first steps of ORG's transformation to become agile. We describe first findings about the reasons why the company decided to move towards this new trend and next what was done so far to promote the change.

A. Why Agile?

ORG has started its transformation to agile about a year ago. It all started when a senior manager with experience in a large e-commerce company was hired to lead the development of ORG's Online Store. One of his major changes as a manager was to introduce agile to the Online Store development team. ORG's Online Store is an independent application and has a dedicated team to take care of it. For over three years the team improved its processes and store itself based on the expertise developed using agile to guide their development. A recent reorg has put this senior manager in charge of the IT department and as the new CIO his first world-wide announcement that the company would go agile.

One of the main reasons behind this decision was **to deliver faster to the customer** given that *"the faster, the better"*, as argued by Senior Manager on Process Improvement [US1].

One of the Senior Managers on Services Applications [BR1], the Senior Manager on Finance Applications [BR3], and the Manager on Financial Services Application Development [BR8] also reported, respectively:

"Agile aims to deliver faster to the customer. Instead of spending 10 months specifying requirements, then 5 more months coding them and 5 more testing, the customer will be able to quickly see added value to the application in a shorter time." [BR1]

"It would be great if we could deliver faster since we have this daily pressure to reduce costs–as any other IT company also has." [BR3]

"(...) requirements that take four months to be developed and two more to be deployed, we want to change such scenario and deliver more quickly and often to the customer." [BR8]

The current long time window between the customer requesting a software and receiving it is also an important motivation for this transformation, said the Financial Services Manager [BR5]. He also said: *"Agile means frequent deliveries, causing the impact of the customer's decision to be taken care in smaller slots of time and allowing changes during the process"*. This same reason was also mentioned by the Director on Infrastructure area: *"(...) there is a gap in the customer expectations since IT takes too long to deliver what is requested."* [BR6]

Agility as **an attempt to be more innovative** is another important reason according to the Director on Infrastructure area. He mentioned: *"We seek agility as a 'driving force' to innovation. We need to (...)"* [BR2]

The transformation to agile also aims **to reduce the communication gaps between business and IT** as mentioned by the current Director on Infrastructure area:

"[Agile] expects closer interactions between business and IT, having the customer more involved. This approximation is welcomed and we do need it, but it can also be risky if the customer does not get engaged." [BR2]

Senior Business Analyst Team Leader added based on his large experience working on the business department: *"We can not forget that a lot that takes place in an organization goes through informal channels. Thus, this agile model will likely be good to make informal things formal."* [US5]

In addition, participants also reported that another important reason for adopting agile company-wise is **to be more flexible to changes**. For instance, *"The requirements are defined early in the year and sometimes they are obsolete when development starts."* as reported by the Director on Services Application Development area [BR7]. Therefore, it is important that the development teams have a closer interaction with business to frequently discuss and re-prioritize what requests are more important at a given time, as mentioned by the Senior Manager on Services Application Development area: *"Flexibility means we can change requirements with a cheaper cost and more easily."* [BR4]

B. Actions Taken

Once the decision has been, the CIO and his committee board discussed a set of actions to be taken to promote the transformation and engaged senior management in promoting them. At mid of the fourth month, a world-wide announcement is made to the entire IT department marking the kick-off of the agile transformation. At this time, teams were informed that by the end of a 12-months period all projects had to be "acting agile". We report next all actions taken during the four months of discussions at the executive and strategic levels and the 7 months of changes at the operational level.

The company prioritized **reorganizing the former organizational structure**, which was a matrix structure. As reported by Senior Manager on Services Application Development area, *"The new structure is defined now by business areas that have their own internal functions."*. He supplemented: *"An interesting change we made was that the executive leadership was completely realigned. We have now a mix of new and 'ancient' people of the business that are within a portfolio area to ensure the new ideas will be welcomed but that we will also not lose important implicit knowledge."* [BR1]

"Leadership is slowly realizing how positive this change will be. It is the matter of internalizing it and later sharing with their team members", added the Senior Director on Process Improvement, the head of the world-wide agile transformation initiative [US3].

One of the main advantages of such organization change and moving more towards a model in which teams are responsible for what was assigned to them is *"To have clear accountability for decisions and deliveries. In our matrix form we had too many people in control and no one controlling anything."*, said the US-based Senior Director on Finance Application Development area [US10].

After the major discussions about the organization structure, the focus turned to **training teams on agile practices**. Team members start then to learn how to adopt such practices in the context of ORG. Several presentations and debate sessions are then organized by senior managers, senior development leads and architects to promote discussions on how agile within an enterprise with complex applications and that is globally widespread can take place. For instance, Senior Manager on Services Application Development area reported:

"Local presentation sessions to all members of a certain office were organized to take place wold-wide in the same week in which teams already going through the transition were motivated to report their experience to others as a way to encourage the adoption of agile and to share good practices." [BR4]

At the same time, some members were encouraged to carry out the Scrum Master's certification and use their knowledge to help other project members to align their actions to the transformation initiative, as mentioned by the Manager on Financial Services Application Development area: *"Some members were already trained as Scrum Master, but on Monday four more people of my team will be going to a training to learn new things. They are very motivated about it. They enjoy to learn new things."* [BR8]. In addition, managers are also learning along with their team member as said by the Director on Infrastructure area: *"We are receiving several trainings, both at the technical and at the managerial level on agile practices and we has been learned from each other then as we put them in practice."* [BR2]

Communities of Practices was a key mechanism behind the success of the large-scale agile implementation in Ericsson, helping teams to mitigate some pressing issues of the transformation [7], and also a source of motivation to ORG's employees too, argued the Senior Business Analyst who leads world-wide the Business Analyst Community of Practice [US7]. The Senior Manager on Quality Management added: *"As important as training people is sharing what we are learning with one another, and here is a good way to do it. I hope they do not cut this practice off."* [US4]

Yet another action taken was **to identify and prioritize pilot projects** to serve as testbeds for assessing the side effects of the reorganization and overall changes promoted during the initial months of the transformation as commented by the Director on Infrastructure area:

"We are prioritizing some projects to be pilot projects based on our understanding that they are more prone to be adherent to the agile philosophy. We have a priority list but we still keep some legacy projects running on 'traditional' [Waterfall] processes." [BR2]

Also, these pilot projects have been closely **supported by coaches hired to assist the company**. *"Local coaches were hired to support each of the IT offices. We looked for experienced professionals who have faced similar issues than ours in other large corporations."*, said the Senior Director on Process Improvement [US3].

The benefits of having experienced coaches working with the teams is recognized as follows: *"They can more easily and quickly to recognize ways to enable the transformation."*. The Senior Manager on Finance Application Development area supplements by saying: *"The coaches are already working with some pilot projects supporting refining the organization structure when projects involve other projects and teams. You know, we have all these dependencies (...)"* [BR3]

Another reported advantage of having coaches supporting the transition process is to ensure the agile practices are correctly used as exemplified by the Senior Manager on Financial Services Application Development area: *"The coaches participate in all the ceremonies and help us understand if we are doing it right. For example, they helped us to revisit our team structures and set up the feature teams in a way that makes much more sense now."* [BR5] and argued by the Senior Director of the same area, located in the USA: *"We are finally learning how to do things. We do understand what it means to do a stand-up meeting. We are not playing anymore."* [US10]

The teams are also aware of the role and extent of responsibility of the coaches as reported by the Senior Manager on Services Application Development area:

"The coaches are responsible for helping the organization to make the transformation happen, but they are not responsible for the transformation itself, this is responsibility of each portfolio area. They provide us with the tools and helps us with their expertise. We have to make it work." [BR1]

Another major action taken was **to refine the team structures** based on the new organization structure. The teams began to take shape as 'feature teams' as described by the Senior Manager on Financial Services Application Development area:

"Before I usually had a team with resources allocated to it and the team members worked on the project, from day 1 to the delivery. When another project was up, then maybe I would have another resources allocated to work with me to deliver it. Today I have an organized structure responsible for a major feature and this structure receives demands from several different portfolios and systems programs and we try to prioritize the demands according to the customer needs, by sprints, not as it used to be." [BR5]

Still about the refinement of the team structures, the Director on Infrastructure area [BR6] and the Director on Services Application Development area mentioned [BR7], respectively:

"We have teams still working on a project-based form but trying to create new structures like feature teams. It is not an easy transition, for example, earlier today I discussed with a colleague about the testing area. The performance testers are still separated from development. This is something we have to change for us to be completely agile. We are not sure how to do it but I think we are going on the right track." [BR6]

"Before, a projects manager would receive reports from all members allocated to projects of a certain portfolio. He would manage how these people were working. Now, with the agile transformation, these managers are responsible for the portfolio management as a whole, they need to know how the feature(s) the portfolio is responsible for is progressing." [BR7]

The US-based Senior Director on Finance Applications believes that *"It is great to have a multidisciplinary team, or this feature thing–I can not recall the name–where everyone needs to know all skills and to be trained on all aspects. This will make people more focused and committed. I like that!"* [US10]

For that, the company is also *"refining the job descriptions and revisiting skills and competencies they expect each role to have"*, commented the Senior Business Analyst Team Leader [US5]. Senior Business Analyst reinforced: *"It is important that we reconsider the competencies each role requires in this new model. For instance, we had 6 levels of business analysts, now we have business analysts and business software analysts merged in one single role, we are still revisiting their set of expected competencies and skills."* [US6]

Further, some teams identified the need to **mix agile practices from different methods** to compose what they need to support their work as highlighted by the Manager on Financial Services Application Development area: *"Some teams are adopting only Kanban, some only Scrum. But there* *are several teams that are using 'Scrumban', as we named here. This is when a team only uses the concept of Sprint from Scrum but adopts the Kanban way to work."* [BR8]

V. Concerns

Despite the initial actions taken by the company, there are still open questions in the opinion of the senior managers. They are aware that in such a transformation process one can expect that as decisions are taken and implemented new issues will come up. Certain actions will be taken and will result in positive outcomes, others will have to be reconsidered. They agree that this is part of any maturation process. However, there are key points they consider critical, mainly due to their large-complex globally distributed configuration, and that are still to be discussed as presented next. The biggest concern reported by each and every of the participants is the transformation to be occurring in a company with **a complex ecosystem of software applications.** This is of concern given the large number of interconnected applications that attend multiple business areas (e.g., sales, manufacturing, finances, HR, etc) and legacy products maintained by several organizational departments and by highly distributed teams. Such concern was expressed by the US-based Senior Business Analyst as follows: *"Interlocks–as interdependencies are mostly named at ORG–will be a big headache."* [US6].

The Senior Manager on Services Application also mentioned: *"We have over 2 thousand applications forming a well-connected web, an extremely complex data stream that is globalized and yet serving regional needs in several cases where our client–several company's departments–have not globalized processes yet."* [BR1]

Another concern about the high amount of interdependent applications is how the 'feature teams' are being set up. Some managers understand that a team at ORG will never be completely independent from other teams given the dependency among the applications. Thus, releases have to be coordinated. The Senior Manager on Services Application Development area reported:

"All applications are very well interconnected, then now we have a concept of release. In theory, it would be ideal to have a release every 3 months, but in practice this can vary because I depend on others to get my feature done and some projects were still not migrated to this release idea." [BR4]

The Director on Infrastructure area also mentioned about this same issue: *"If I need to automate a process, the process will probably hit several systems to complete the task. Thus, the automation needs to have all applications with a 'ready status' to be able to be delivered."* [BR6]

The **limited spread of the transformation to parts of the IT department** is among the major concerns of several senior managers. For instance, training and coaching have been only assigned to support development teams. Infrastructure and services teams are having to provide services to development in an 'agile fashion' without having been included in the transformation initiative as reported by one of the Director on Infrastructure area: *"It is necessary to look at the software life-cycle as a whole, including the question of provisioning infrastructure. The training offered is very good for the development team, but infrastructure has no clear guidelines defined yet so we feel kind of lost"* [BR2]. He added: *"The technological complexity of the environment, I think, is another important factor. We are talking thousands of database, thousands of applications [silence]. Then, when I have to talk about a transformation that will affect, for example, 'refreshing a hardware structure' to all applications of a portfolio, then I have to ensure that I will be able to handle such major change and this has not been discussed yet but I am already facing such issue."* [BR2]

The other Director on Infrastructure area is concerned about the available infrastructure itself: *"How can we support constant deliveries if we are not sure which are the infrastructure needs for that and we do not have enough DBAs to work or even servers to support the applications?"* [BR6]

On the other hand, the Senior Director on Finance Application Development worries that *"the quality of the service provided by production support people can be jeopardized since their way of work will not be in sync with development. These guys still have to be fast to provide solutions to live issues but they might not know how to interact with development teams anymore since they are not aligned with what the coaches are doing."* [US10] Senior Business Analyst has a similar worry: *"We need to ensure consistency in some level so we stay functional and make people's life easier."* [US6]

To make the customer adopt agile in their processes is another concern related to the transformation boundaries within ORG. For instance, the Senior Director on Finances Application shared: *"What we need is a proper customer involvement. This is really a 'big sticking part of all this'. We need customer who can use requirements in an effective way. They have to be committed to do it and we also need to learn to be disciplined about it."* [US10]

Senior Portfolio Planning Manager added: *"We urgently need a centralized solution for business personnel make their requests and prioritize them."* [US8]

The Financial Services Application Senior Manager commented: *"Customer representatives are aware that the company-wise is going agile, but are not directly involved in the actions taking place so far, so I guess that they have not yet realized that they will have to be more active overall, to respond faster to our requests and to more quickly consider what are their priorities, to be able to handle new deliveries in a faster pace, among so many other things."* [BR5].

The Senior Business Analyst Team Leader argued: *"What customers have to realize is that we need a day-to-day proximity with them throughout the development cycle but when coding the contacts will likely slow down. We do understand that interacting with IT people takes away from their daily job duties but in the end we are providing them with solutions that will help them in the end to better do their work. So they need to find a balance."* [US5]

"We can work with PO [product owners] proxies. We are okay with that", concluded the Process Improvement Senior Manager [US1].

Senior Manager on Quality Management supplemented: *"We believe that by involving more the customer we can* *increase the overall customer satisfaction with IT services. This is what my team is looking for in this transformation: to have better results in our periodic customer satisfaction surveys."* [US4]

The Director on Services Application Development area highlighted: *"We definitely need the customer closer to the IT department, and fast."* [BR7]

The **annual roadmap** to decide on budget for the departments is also a concern. The Senior Manager on Finance Application Development area is afraid ORG might collapse in the coming year:

"There is no one looking at the roadmap so far. We are still learning how to distribute our projects in an annual basis mind now (...) In fact, what we need is to learn how to prioritize requests as the year goes by so the customers will always get what they need faster. It really does not matter what model we are following, we need rules to decide on what adds more value, so prioritization will be made easier and more clear." [BR3]

The US Finance Senior Director argued: *"Agile fits well when there is a lot of unknowns' but it cannot be good to fixed contracted models like ours."* [US10]. His office partner mentioned: *" Our budget is fixed, we will likely never change that. Effective prioritization is the key."* [US10]

In addition, the Director on Infrastructure area reported: *"The company has an annual roadmap, so we plan in accordance with the resources and budget we receive; I still have no clear vision of how it will look this really with agile."* [BR2]

The Senior Portfolio Planning Manager, responsible for the annual roadmap planning confessed: *"We do know we have to change the way we do our budget forecast but we still have not found a way and the clock is ticking. It is dependent on the business budget funds so it is not just changing the process, it is more complicated than that."* [US9]

Senior Manager on Process Improvement added: *"We need to work based on priorities."* [US1] Her colleague supplemented added: *"(...) We just are not sure how to move from a cost-based model to a priority-based one. I think the first step is to have a demand supply staff, like a roadmap change management board, to ease things down next year. Then we buy ourselves some time."* [US8]

The **global distribution of the teams** is another factor that concerns most of the senior managers. ORG started in the USA, later created an office in Brazil, next in India and Malaysia. There are also groups of IT professionals allocated along with business offices in Ireland and Russia, or example. Over the last 13 years, the teams went from co-located to distributed between two countries (e.g., USA-Brazil, USA-India) and finally to distributed over three continents (e.g., USA-Brazil-Ireland-Malaysia). Differently from a large number of agile companies in Europe that are distributed up to 5 countries within the same time zone or, at most, 1 or 2 hours apart, at ORG teams have the challenge to have to coordinate with remote teams members that are often 8-14h distant from one another. This concern can be perceived in the following excerpts by the Director on Infrastructure area, the Senior Manager on Finance Application Development area, the Senior Manager on Services Application Development area:

"We work with distributed teams. Everything we do is distributed, my team is over 4 continents. It is very challenging from the point of view to have a synchronous meeting, even when we make an effort to compromise our working hours. Not even to mention that for more that we try, it will never be the same than working side-by-side." [BR2]

"There are just too many people around too many countries around the world to effectively be flexible and agile in our processes." [BR3]

"We have five thousand people distributed among Americas, Europe, and Asia, and this will not change because it is how our business survive. We need to be where our end-client is (...)" [BR4]

Most of applications at ORG were born when the company was still small and with a single office. Despite, they have been maintained by teams of senior professionals that are well aware of how the applications work. As people retire, new members are moved from junior to senior positions and being assigned to be in charge of keeping the applications working. With this company-wise transformation, **old technologies and legacy systems** become a serious concern as reported by the Senior Portfolio Planning Manager: *"We are still learning about whether agile is fit for all projects we have. Some might never be able to go agile like the legacy systems. We are still not sure yet."* [US9]

The Senior Manager on Services Application said: *"Our applications were not designed thinking on agile methods. They are 10-15 years-old with legacy code that is very difficult to have a feature teams responsible for it, for example."* [BR4]

The Senior Manager on Finance Applications said: *"In a complex environment with different systems communicating with each other and integrated by old technologies, it is insane to try to move legacy systems to agile."* [BR3]

Other concern is about **Brazilian's laws and fiscal year's budget** as mentioned by the Senior Manager on Financial Services Application:

"The shares must be tendered and we are still managed by quarter to quarter within the fiscal year. Management is tied up to the fiscal year, which ensures pre-delivery and planning visibility different from what happens in agile, I think. The cost is likely discussed for the short term deliveries, without considering a closed scope." [BR5]

The Senior Manager on Finance Application Development reported: *"I have some demands that vary greatly according to the [Brazilian] government and the law under ORG is hosted here [in Brazil]. For instance, if the law changes, we need to change to be compliant with it. Also, the [Brazilian] government is always changing rates, thus we have to adjust the systems in a very frequent fashion."* [BR3]

Agile evangelists is also a concern because they often believe only in agile in itself, not in a process that can be effective through a transition and that represent in a culture change. Transformation is a slow process, so many people do not believe that teams can be effective in cases where the company still running some projects on 'traditional' processes. This issue is mentioned by the Services Application Development and Financial Services Application senior managers, respectively:

"Evangelists do not believe that change can occur and be effective. They just believe in what is described in the agile manifesto. They end up damaging the transformation process. We know it will be slow (...)" [BR4]

"(...) a problem that I see today is that agile is kind of a religion, there are many people who strongly believe that it can even cure cancer [laughs]." [BR5]

Lack of formal documentation for requirements is another issue cited by a few participants as showed by the excerpt below:

"So, if you do not have any documentation, it has happened several times to us, to have projects that goes back and forth and then our customers say no–customer here is always internal departments. They say 'it was not what I wanted', then we say 'but it is what you documented' so we have how to defend our position. We always had this fear to miss formalities despite all the interactivity that agile offers." [BR1]

The adaptation and the redesign of tools to support work throughout the organization is a less concerning issue but something the company knows that will have to be considered sometime sooner than later. For instance, the Process Improvement Senior Manager mentioned: *"We will need tools to support virtual stand up meetings, visualization of data exchanged among people to facilitate comprehension of what is going on given that most team members do not work with co-located colleagues and have large time zone differences, and so many others that I could spend the entire hour listing here."* [US1]

VI. Discussion

As other companies [8], ORG has main and different motivators to aim to 'become' agile. We could perceive that the most important reason why ORG decided to go agile is similar to those reported in literature (e.g., [9]): to deliver faster. This is one of the main characteristics of agile development and proven to be still a common issue in industry.

The reorg of the former organizational structure reported by some of the participants is a natural reaction when an organization is moving to a new paradigm (e.g, from waterfall to agile). When the company was using waterfall model it was fit to have a matrix structure. However, when the migration to agile started it was easy to perceive that the organizational structure needed to change to support teams working based in features using evolutionary and iterative development.

Training team members on agile practices is an incisive step to the transformation that includes qualifying people on the 'basics' of the new mindset (e.g., self-training, trial and error, changing priorities), as reported by Gandomani [10]. Our findings also indicated that such trainings were one of the first actions to be taken at ORG, despite the large amount of employees already familiar with agile practices. The training does not only discuss the concepts but put them into perspective at ORG's context, cited as one of the benefit of having them.

Pilot projects are common strategies to test whether something is working and it was also used by ORG. It started with a set of a few projects only, and now is a company-wise strategy, as mentioned by our participants. This was consider one of the successful factors to support the transformation in large companies such as in Gap, as reported in [11].

Coaches hired to assist the company is among other of the successful factors cited by Gandomani [10]. Such action has allowed people at ORG to discuss their points of view, to ask for clarifications, to have mediators to guide discussion, and to have experts mentoring their actions.

The Primavera Systems [[12]] reported that when going to such a transformation often companies choose to follow a certain agile method as the main method to be followed such as Scrum or a mix of such methods like 'Scrumban'. Professionals report that having a method to follow helps them to guide what has to be done with different needs of projects. This also took place at ORG and was also pointed out as a facilitator.

About the reported senior managers' concerns, some are similar and some are new to literature, the next paragraphs we will describe about it.

The concern related to complex ecosystem of software applications reported in still an open question in literature. We did found challenges and limitations reported related to the usage of agile practices in a distributed environment (e.g., [13]) and on a project with dependent projects (e.g., [4]) but none referred to such a complexity as in ORG.

The limited spread of the transformation to the IT department only is another concern. For being a large company with several financially independent departments, the transition has been so far just been discussed among IT personnel. Business people are aware of the transformation but they still are not involved in it in practice. Torgeir and Nils [14] proposed a research agenda for agile in large-scale in which they cite that customer collaboration is still an open issue in such setting.

As per the annual roadmap concern, we found a study from Borland [15] that showed that a roadmap was one obstacle, having the company moved to a catalyst solution. The new strategy has been established based on a common understanding among the involved parts and the agility was achieved across their product delivery value chain.

Also, global distribution of the teams is an impact in adoption agile practices mentioned by IT director and seniors managers, in literature we found agile transformation with a background of a company that has the same situation. However about the concern, Kirsi [16] identify that this does not itself imply or even affect to improvement in the quality of working practices.

To old technologies and legacy systems concern, we found in the literature a successful case when implementing agile [17] and they reported practices to large legacy applications to move from fragility towards to agility, as follows: Inspect and Adapt (having agile attitude); Go slow to Go Fast (to establish a plan and work incrementally); Prepare to Sustain (to have a dedicated role to inspect and adapt the process); Divide and Conquer (break monolithic projects into smaller projects) and

Piggyback (to try initiate something already accepted rather than something new).

Lack of formal document for requirements is a problem described as the barrier of project team had in terms of capturing business requirements in an easy and effective way defined by Cristal, Wildt and Prikladnicki [18] and also mentioned as a concern to the IT Senior Manager on Services application development area. On the other hand, Brazilian's laws and fiscal year's budget and agile evangelists are still open concerns reported not mentioned in literature.

Agile transformation is not a simple process to pass thought. However we already have many papers who are current reporting about the topic and showing success cases doing the transformation happen in specifics context [4], we still have some open questions related to the concerns about large-complex globally distributed scenario (e.g., the limited spread of the transformation to the IT department and complex ecosystem of software applications).

VII. Final Remarks and Future Work

This paper presented a report of the initial steps of ORG' company-wise transformation to agile. ORG's two particular configurations in relation to other large companies that have already been through this transformation are: it has an ecosystem of applications that are dependent on each other and it has large teams globally distributed around the globe with no or little overlapping work hours. Both characteristics make ORG's situation quite unique, thus likely requiring specific measures to leverage success.

Despite the challenges and its early stage of the transformation process itself, we could observe that ORG is following the same direction path of most successful transformation initiatives as reported in literature. However, we do know that the concerns revealed by our study will have to be discussed and handled by ORG when the time comes.

We are currently designing a second field study. We aim to interview professionals of large companies, senior coaches and consulting professionals that have participated on such a transformation is large companies in order to identify what are the main challenges they have gone through and which good practices they propose. This work is inspired in an informal benchmarking we conducted while attending an agile-based international conference in May'15. Our main goal with this second study is to add to Fontana and colleagues [19] that identified the following categories, for maturity in agile: practices to become agile; team composition and behavior; deliveries (evolution from traditional to continuous delivery); requirements (transition from traditional requirements elicitation to use stories); product (practices to improve the software product) and customer relationship, and work by supplementing the mechanisms they point out to support agile teams to evolve.

Acknowledgment

We would like to the participants of our study.

References

[1] K. Beck and colleagues, "Manifesto for agile software development," 2001. [Online]. Available: http://www.agilemanifesto.org/

[2] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. ACM*, vol. 48, pp. 72–78, May 2005.

[3] P. Ranganath, "Elevating teams from 'doing' agile to 'being' and 'living' agile," in *Proc. of Agile Conf., Salt Lake City, USA*, 2011, pp. 187–194.

[4] C. Fry and S. Greene, "Large scale agile transformation in an on-demand world," in *Proc. of the Agile Conference, Washington, USA*, 2007, pp. 136–142.

[5] K. Korhonen, "Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context," *Software Quality Journal*, vol. 21, 2013.

[6] B. Al-Ani, S. Marczak, R. Prikladnicki, and D. Redmiles, "Revisiting the factors that engender trust of global systems engineers," in *Proc. of the Int'l Conf. on Global Software Eng., Bari, Italy*, 2013, pp. 31–40.

[7] M. Paasivaara and C. Lassenius, "Communities of practice in a large distributed agile software development organization – case ericsson," *Information and Sw Technology*, vol. 56, no. 12, pp. 1556–1577, 2014.

[8] R. Valade, "The big projects always fail: Taking an enterprise agile," in *Proc. of the Agile Conference, Toronto, Canada*, 2008, pp. 148–153.

[9] G. Cloke, "Get your agile freak on! agile adoption at yahoo! music," in *Proc. of the Agile Conference, Washington, USA*, 2007, pp. 240–248.

[10] T. Gandomani, H. Zulzalil, and M. Nafchi, "Agile transformation: What is it about?" in *Proc. Malaysian SEng. Conf., 2014, Langkawi, Malaysia*, 2014, pp. 240–245.

[11] D. Goodman and M. Elbaz, "It's not the pants, it's the people in the pants - learnings from the gap agile transformation what worked, how we did it, and what still puzzles us," in *Proc. of the Agile Conference, Toronto, Canada*, 2008, pp. 112–115.

[12] B. Schatz and I. Abdelshafi, "Primavera gets agile: a successful transition to agile development," *IEEE Software*, no. 3, pp. 36–42, 2005.

[13] D. Turk, R. France, and B. Rumpe, "Limitations of agile software processes," *arXiv preprint arXiv:1409.6600*, 2014.

[14] T. Dingsøyr and N. B. Moe, "Research challenges in large-scale agile software development," *ACM SIGSOFT Software Engineering Notes*, vol. 38, no. 5, pp. 38–39, 2013.

[15] D. Wilby, "Roadmap transformation: from obstacle to catalyst," in *Proc. of the Agile Conference, Nashville, USA*, 2009, pp. 229–234.

[16] K. Korhonen, "Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context," *Software Quality Journal*, vol. 21, no. 4, pp. 599–624, 2013.

[17] V. Shah and A. Nies, "Agile with fragile large legacy applications," in *Proc. of the Agile Conference, Toronto, Canada*, 2008, pp. 490–495.

[18] M. Cristal, D. Wildt, and R. Prikladnicki, "Usage of scrum practices within a global company," in *Proc. of the Int'l Conference on Global Software Engineering, Bangalore, India*, 2008, pp. 222–226.

[19] R. Fontana, V. Meyer, S. Reinehr, and A. Malucelli, "Progressive outcomes: A framework for maturing in agile sw development," *Journal of Systems and Software*, vol. 102, pp. 88–108, 2015.

ATTACHMENT C – Research Protocol of Systematic Review Literature

Protocol for studying the main Evidences and Aspects related to Scale Agile Development

Greice Roman and Sabrina Marczak

Background

The purpose of the study described in this protocol is to review the state-of-the-art of the main evidences and aspects related to Scale Agile Development. We will conduct a Systematic Literature Review (SLR) aiming to identify Scale Agile Development related literature, including empirical or theoretical work.

Unit of analysis

We are interested in identifying work that describes how agile has been implemented in large scale.

Research Questions

The research questions to be addressed by this study are as follows:

- What is the state-of-the-art in the studies of scale agile?
- What are the project and enterprise evidences found in the large-scale agile researches studied?
  - Does the material found include definitions of what is considered large-scale agile?
  - Does the material found include definition of the organization researched and project complexity?
  - What are the characteristics of the organizations?
  - What are the characteristics of the projects studied?
  - Is there a definition of high-complexity project?
  - Does high-complexity involve distributed development? In case yes, what kind of distribution across cities, states or overseas?
  - What are the agile frameworks used in implementations studied?
  - What are the lessons learned and challenges faced by the material studied?
  - What are the state-of-art limitations pointed by the literature and work researched?
  - What are the limitations of this SLR work?

Keywords: Large Agile; Scaled Agile; Large-Scale Agile; Agile Enterprise; Agile Organization; Scrum of Scrums, Large Scrum; Large XP

Search Process

The search process will focus on papers and journals since 2000. The nominated journals and conferences are shown in the following Table.

Sources to be Searched

Source	Responsible
IEEE Xplore	Roman
ACM	Roman
Springer	Roman
Scopus	Roman
Science Direct	Roman

This research will be conducted in partnership with PhD. Rafaela Fontana from Federal University of Parana and PhD. Josiane Kroll from PUCRS.

Inclusion criteria

Articles or Papers on the following topics, published between Jan 1st 2000 and Dec 31th 2015, will be included.

Exclusion Criteria

The following types of papers will be excluded

- Papers with no defined research questions, no search process, no defined data extraction or without data analysis process will be excluded.

When a study has been published in more than one journal/conference, the most complete version of the survey will be used.

Primary study selection process

The results will be tabulated as follows:

- Number of papers per year per source
- Number of candidate papers per year per source
- Number of selected papers per year per source.

The relevant candidate, selected studies rejected studies will be checked by another researcher. We will maintain a list candidate papers that were rejected with reasons for the rejection.

Quality Assessment

Are the review's inclusion and exclusion criteria described and appropriate?

- Is the literature search likely to have covered all relevant studies?
- Did the reviewers assess the quality/validity of the included studies?
- Were the basic data/studies adequately described?

The questions are scored as follows:

- Question 1: Y (yes), the inclusion criteria are explicitly defined in the paper, P (Partly), the inclusion criteria are implicit; N (no), the inclusion criteria are not defined and cannot be readily inferred.

- Question 2: Y, the authors have either searched 4 or more digital libraries and included additional search strategies or identified and referenced all journals addressing the topic of interest; P, the authors have searched 3 or 4 digital libraries with no extra search strategies, or searched a defined but restricted set of journals and conference proceedings; N, the authors have search up to 2 digital libraries or an extremely restricted set of journals.
- Question 3: Y, the authors have explicitly defined quality criteria and extracted them from each primary study; P, the research question involves quality issues that are addressed by the study; N no explicit quality assessment of individual papers has been attempted
- Question 4: Y Information is presented about each paper; P only summary information is presented about individual papers; N the results of the individual studies are not specified.

The scoring procedure is Y=1, P=0.5 and N or Unknown=0.

The data will be extracted by one researcher and checked by another.

Data Collection

The data extracted from each paper will be:

- The source (i.e. the conference or journal).
- The year when the paper was published. Note if the paper was published in several difference sources both dates will be recorded and the first date will be used in any analysis. This is necessary in order to track the research subject activity over time.
- Information related to projects, organizations, architecture, large –agile implementation findings and challenges.

The data will be extracted and uploaded in in an online database .

Data Analysis

The data will be tabulated (ordered alphabetically by the first author name) to show the basic information about each study. The number of studies in each major category will be counted.

The tables will be reviewed to answer the research questions and identify any interesting trends or limitations in current related research as follows:

- Question 1 What are the researches made related to the implementation of agile methodologies in high complexity and large enterprise projects? This will be addressed by simple counts of the number of agile and large enterprise projects related papers per year.
- Question 2 What are the characteristics of the Enterprise organizations studied? This will be addressed by listing the characteristics of enterprise organizations part of the papers analyzed.
- Question 3 What are the characteristics of the projects studied? This will be addressed by listing the characteristics of the projects part of the papers analyzed.
- Question 4 What are the agile frameworks used in implementations studied?

This will be addressed by listing the frameworks (XP or SCRUM) part of the papers analyzed.

- Question 5 What are the conclusions found? This will be addressed by listing results / conclusions found part of the papers analyzed.

- Question 6 What are the limitations of current research? We will review the range of SE topics, the scope of SLRs and the quality of SLRs to determine whether there are any observable limitations.

Dissemination

The results of the study should be of interest to the software engineering community as well as researchers interested in implementation of agile methodologies in high complexity and large enterprise projects. For that reason we plan to publish the results on a journal.

References

1. Kitchenham, B. Guidelines for performing Systematic Literature Reviews in Software Engineering, Keele, UK, Keele University, 2007.