Using Scrum for Implementing IT Governance with COBIT 5

Ana Cláudia Martins Amorim
Dep. of Computer Science and Engineering
Instituto Superior Técnico
Lisbon, Portugal
ana.claudia@ist.utl.pt

Abstract— COBIT 5 is a widely-used framework for implementing sound governance of enterprise IT (GEIT). Currently, the ISACA's official implementation solution follows a sequentially ordered process that is still not the most suited to practitioners in general, raising several issues related with lack of commitment from top management and misaligned solutions. Nevertheless, new project life-cycle strategies have emerged along with the agile paradigm for project management, providing flexible and adaptable environments for projects where the solution is complex and not clear, delivering the product incrementally with feed-back loops. With this research, we aim to eliminate some known challenges of COBIT 5 by providing a Scrum based approach to address COBIT 5 programmes. A Design Science Research Methodology was used to guide this work, where two iterations on the solution development, demonstration and evaluation activities were performed. With two different approaches and demonstrations, we were ableto identify some relevant finds with positive results regarding the objectives established.

***Keywords*— COBIT 5, Scrum, governance of enterprise IT, agile implementation.**

I. Introduction

Information technology (IT) and related information technologies are pervasive not only in simple day to day functions everywhere, but also playing critical roles in most organizations. As it grows, IT has become a business enabler, taking major responsibility in effective and efficient product delivery, bringing more value to stakeholders [1], [2].

Adding value, while managing enterprise's use of IT and mitigating the IT risks, made governance of enterprise IT (GEIT) a necessity for accomplishing enterprise's goals, and therefore, a fundamental concern of corporate governance [3].

A succeeded GEIT implementation requires a culture of well-defined enablers as organizational structures, principles and structured governance and management processes according the enterprise's vision [3]. Frameworks are the best tool to help this implementation [4] since they provide adaptable solutions for any type of organization. For that purpose, ISACA developed COBIT [3] as a framework to "*provide guidance in evaluating, directing and monitoring an enterprise's use of IT*" [5], giving an objective way for companies to align business strategies with IT goals.

Although COBIT 5 is wide recognized as one of the most used GEIT frameworks [13], it is also considered too large and complex, taking years for its full adoption [6], [10]. Regarding the framework's complexity and overarching scope, is almost impossible to manage a COBIT 5 programme without a defined project management methodology [12]. With the countless methodologies that one can currently find, one must choose wisely and consider all the project particularities and challenges [18].

The current approach for a COBIT 5 programme, detailed in the implementation official guide [5], has its basis in the traditional paradigm of project management, presenting a linear strategy for the implementation life-cycle. Traditional methods are defined by an exhaustive planning phase that eliminates the need for any changes in the remaining project life-cycle, so the process follows the original plan, going through all phases in an orderly fashioned way [18].

On the other hand, agile approaches such as Scrum and Kanban appeared as an alternative paradigm to the traditional ones. These are specially focused on building solutions that are not clear at the beginning [18], by putting the client's interests first, embracing constant feedback and welcoming changes in the requirements at any time of the process [7]. Although these methodologies are mainly applied in the software development field, organizations are starting to adopt this new paradigm in several areas beyond software development such as consulting, manufacturing, coaching, etc. [24].

In this document, we focus our research on Scrum, discussing the application of agile methodologies on GEIT implementations with COBIT 5, with the main goal of using agile principles to overcome some known challenges of COBIT 5 such as lack of support from top management, failure to understand the environment, resistance to change and scope misalignment [5].

Two iterations on Design Science Research (DSR) process were performed. Starting on the design and development step, each iteration of the solution was demonstrated in a distinct COBIT 5 programme, and evaluated with interviews in order to compare our goals with the results observed from its usage.

II. Research Methodology

Design Science Research Methodology (DSRM) was the methodology chosen to guide this research work. DSRM provides a process model for doing research in Information Systems and other applied resource disciplines, as well as a mental model for reviewers to evaluate researchers [8].

The main goal of design science in IS research is to produce and evaluate an IT artifact that supports the solution for an identified organizational problem [9]. The artifacts produced can be constructs (vocabulary and symbols), models (abstractions and representations), methods, (algorithms and practices), and instantiations (implemented and prototype systems).

The DSRM process establishes 6 phases [8] and our research conforms as described next:

1. **Problem identification and motivation**: Lack of adoption of formal project management methodologies on COBIT 5 programmes.
2. **Define the objectives for a solution**: Facilitate COBIT 5 adoptions, decreasing some known challenges.
3. **Design and development:** Scrum based methodology for GEIT implementation with COBIT 5.
4. **Demonstration:** Field study at Portuguese Finance Ministry (1st iteration) and field study at bank X (2nd iteration).
5. **Evaluation:** Models and frameworks, interviews and field studies.
6. **Communication**: Dissertation and papers.

III. Problem and Motivation

This section is related with the "Problem Identification and Motivation" step from DSR model (step 1), introduced in section 2 as the research methodology used in this work.

As aforementioned, COBIT 5 is a widely known framework for governance of enterprise IT (GEIT) implementation [13]. When correctly adopted, brings much value to organizations promising to minimize the risk of IT [13], [14]. Nevertheless, there are still a few more things that can be done to provide more support to practitioners when using this framework.

First of all, there is a known lack of literature on COBIT 5. Despite the existence of official manuals to support its adoption [4], [5], previous studies on COBIT 5 are scarce [6], specially regarding lessons learned of previews implementations and practical examples. Case studies and lessons learned can be fundamental for practitioners, especially when COBIT 5 is such a complex and overarching framework, having 7 enablers, 37 processes and more than 6188 interdependencies between them [10], [11], [14] - "Putting processes in places at any organization that covers even a fraction of what is encompassed in, for example, COBIT 5, is highly challenging" [28].

Moreover, ISACA itself aware in its manuals for some "practical issues that need to be overcome*"* [5] such as lack

of senior management commitment and support, communication issues, failure to understand the environment, resistance to change and scope misaligned with requirements. This leads to the main motivation of this research work.

The current implementation life-cycle for a COBIT 5 adoption programme can be one of the problems, since it seems to be a linear strategy. The basic idea of traditional approaches is that projects are predictable and have clear boundaries [17], [18]. This makes them easy to plan in detail so the project flows without rethinking the process, since changes are not expected.

Although this solution works for many projects, the strategy used must contemplate the project characteristics and there is no evidence that a traditional approach is the correct one to use in a GEIT implementation.

IV. Objectives

In this chapter, the authors intend to indentify the objectives of a solution for the problem stated in chapter 3 (step 2 of DSRM).

The main goal is to facilitate COBIT 5 adoptions, decreasing some known challenges through the usage of an agile approach for managing the programme.

Therefore, the objectives for this research are the following:

- Objective 1: Increase senior management commitment and support during the whole programme;
- Objective 2: Detect misalignements earlier in the development;
- Objective 3: Decrease the resistance to change;

V. Literature Review

In this section, the authors provide, based on literature, general understanding of COBIT 5 life-cycle as also provide definitions of key concepts regarding Scrum.

A. Using COBIT 5 for GEIT implementation

ISACA developed COBIT 5 as a framework to "*help enterprises implement sound governance enablers*" [5], giving an objective way for organizations to align business strategy with IT goals. For this purpose, the framework comprises 5 principles, 7 enablers, 26 roles and 37 management and governance processes and its correspondent good-practices, as well as seven sequential phases (described in table 1) [4], [15].

The seven enablers are the core of this framework since they are the pillars for an effective GEIT implementation [4], [5], [19], [20]. They are: Principles, policies and frameworks, Processes, Organisational Structures, Culture, Ethics and Behaviour, Information, Services, Infrastructure and Applications, People, Skills and Competencies.

B. Scrum

Agile is an iterative and incremental paradigm for product development that aims to create value under constant changing environments, being adaptability its key characteristic.

Although it was initially created as a solution for software development, its principles can be applied in other disciplines and it is starting to be adapted progressively more in areas beyond software [24], [27].

Scrum basic idea is to build the product incrementally using several short iterations (sprints) with feedback of the work done, instead of deliver a complete product near the end [7], [16], [17], [18]. Its mindset is the empiricism [16], which makes it simple and straightforward, since it is based on the idea that knowledge comes from experience "*simplicity is essential*" [7].

When applying Scrum, one must adopt some specific rules derived from its values of transparency, adaptation and inspection [16]:

- There are three main roles: Product Owner, Scrum Master and Development Team. Together, they form the Scrum Team.
- The project is divided into sprints – time boxes between 1 to 4 weeks. At the end of each sprint it must be delivered some valuable product increment.
- Inspection and adaptation are assured through formal events: at the beginning of each sprint (Sprint

Planning), at the end of each day (Daily Meeting) and at the end of each Sprint (Sprint Review and Sprint Retrospective).

VI. Related Work

Looking for related work in literature, the authors faced some lack of references for previous studies, not only concerning COBIT 5 programmes, but also Scrum based approaches for areas beyond software development.

Faced with this problem, the authors focused on studies about Scrum-based approaches for complex projects (although in software areas) and scaling agile frameworks. In this section these and other relevant research in this matter are analysed.

A. Requirement analysis and project planning

COBIT 5 is a framework for implementing GEIT, which identifies and describes a set of end-to-end processes, related base-practices and activities. In order to be in compliance with COBIT 5 good-practices and principles, the processes in which IT governance enablers are implemented, should suffer a readjustment according the target established.

There is a lot of work that must be addressed before and after the development of a governance solution, as showed in table 1, where projects begin only in step 5, performing 4 steps before the actual implementation of the solution. These 4 steps are mainly related with requirement analysis for projecting and planning the encompassed projects.

Indeed, the authors found special interest in the hybrid approach Water-Scrum-Fall [30], that is approached as the reality of most of the Scrum processes.

This approach is supported by the idea that business analysis and release management continue to follow more traditional approaches leaving Scrum only to the development team level [26]. For that reason, this approach suggests a more realistic life-cycle for Scrum projects of three distinct parts: Water, Scrum and Fall.

The Water phase is to plan and define the high-level requirements needed to set the project direction, timeline and budget. Following is the Scrum part, where the work is done. The initial requirements are not definitive, the main goal of this approach is to define an initial idea of the project that only means to set direction, requirements may change during Scrum phase. Finally, the Fall part of the project is for software test and rollout, this formal step is to determine the acceptance of the product as a whole [26], [30].

Regarding this subject, other solutions were considered particularly relevant to the authors, namely C. Thiemich and F. Puhlmann study [21] of Business Process Modelling in which they propose to improve each process incrementally using Scrum, with process releases each 5 to 10 sprints. For defining the project direction and evaluate BPM maturity, before the project starts there is a dedicated scoping phase. Additionally, at the beginning of each release there is a sprint 0 for defining the initial requirements, where Process Backlog is an output expected at the end. The next sprints are dedicated to the process increment (same as product increment).

The concept of process releases may be an interesting addition, however, when considering COBIT 5 one must not forget that is not just about process design and implementation. There are other enablers that must be contemplated in a GEIT implementation such as information, culture or even roles. Although the enablers must be implemented under specific and identified processes, not all cases of improving process maturity will be about designing a new process and implement new procedures; often means, for example, to design a necessary output or adding a new role.

Looking at these two solutions for scoping and requirements analysis, both scope and kick-off phases of [21] are in some way traditional steps, which makes these two solutions very similar at the Water level of the Water-Scrum-Fall approach.

B. Scrum for programmes or complex projects

Using Scrum to manage a COBIT 5 programme is not a straightforward fit, especially since Scrum was originally designed as a methodology for software development projects.

One of the challenges of developing a Scrum-based approach for COBIT 5 is the scalable factor. The original Scrum framework [16] is conceived for a single project, however COBIT 5 is approached as a programme [5].

Since Scrum is a framework, it is possible to shape it to the problem needs. The need to scaling up Scrum is not a novelty in literature, in fact, the authors found some interesting solutions and practical case studies that will be following discussed.

The major challenges to address in a complex project or a programme are mainly related with scaling the coordination mechanisms between teams [31], [34].

A scaled Scrum solution was designed and tested at Nokia and documented in [31]. The proposed framework suggests having 3 types of product backlog: one for the general requirements of the programme, one for each release and, at the release level, a scrum team backlog. In addition, a Program Manager above all Scrum Team Masters is proposed, as well as a Program Product Owner above all Team Product Owners. Meetings are also multiplied at the programme level between releases.

Another possible solution is the scaled agile framework SAFe [32]. This agile framework promises to manage complex projects and programmes, however it includes a considerable number of additional elements, increasing the complexity. With 2 levels (programme level and team level) the idea is to have programme releases composed of four iterations and an additional one to plan the next releases. Besides having all the basic concepts of Scrum, it includes more than ten other elements such as a Scrum of Scrums, programme backlog, System Eng., product manager, business owners, product increment objectives, etc. While this solution may be ideal for many complex projects or entire organizations that want to become agile, the additional complexity can be a very high cost to pay for a programme as challenging as COBIT 5.

Many are the solutions and possible deviations of Scrum. As a framework, one of the best things about Scrum is the possibility to apply its principles in a context, adapting and working out its practices considering the constraints.

VII. 1ST ITERATION

A. Proposal

In this section, the authors detail the solution designed to achieve the identified objectives (step 3 of DSRM).

The authors chose Scrum due to its nature of welcoming changes, perform continuous delivery and obtain constant feedback [7], which the authors intend to verify if contributes to strengthen the relation with stakeholders, and provide a collection of solutions that best enables IT governance considering the constant changing environment most organizations live.

Therefore, the authors propose an incremental strategy for developing IT governance solutions and to eliminate some known issues of COBIT 5 adoption programmes.

This chapter relates to the first iteration. We start by presenting the initial design and development of the methodology, followed by its demonstration and results. Finally, we present the evaluation performed using two sets of interviews to evaluate the artifacts and the results obtained from the demonstration.

1) Specification of Scrum concepts for a COBIT 5 programme

The first artifacts developed were the constructs. A construct provides the language in which he solution is defined, and it is after used to build the model.

The set of constructs are here represented in the form of three tables. In those the authors explain how this solution differs from the original definition in terms of events, artifacts and roles. Due to the complex nature of COBIT 5, the authors chose to adopt a simpler approach than the ones investigated, starting by cutting off some of the additional elements proposed by those solutions, addressing just the ones that seemed indispensable for this programme to work.

Regarding scrum roles and artifacts, this first iteration considers that exists an implementation team for each process/project with a sprint backlog for each team, a scrum master for each team and a single product owner for the programme that develops and maintain a single product backlog), similar to the LeSS basic version: “Multiple teams building a single product work from a single Product Backlog that defines all of the work to be done on the product. They do not each have their own Product Backlog. Product Backlog Items are not pre-assigned to the teams.”.

Concerning scrum meetings, once more, they need to be scaled to maintain the coordination at the programme level, as we observed from the analysis made to the scaling methods in literature (sec- tion 3.2), however, once again, to minimize the complexity, the authors chose to keep the meetings just at project level, since product backlog is also unique. Product Owner should maintain the integrity of the programme, and solve problems related with dependencies between the projects together with the respective scrum masters, considering possible Scrum of Scrum meetings.

2) COBIT 5 life-cycle

The model was designed following the West’s et al. Water-Scrum-Fall approach, previously analysed.

It was designed (Fig. 1) identifies the seven implementation phases in red circles aiming to show how they match with each phase of this new approach.

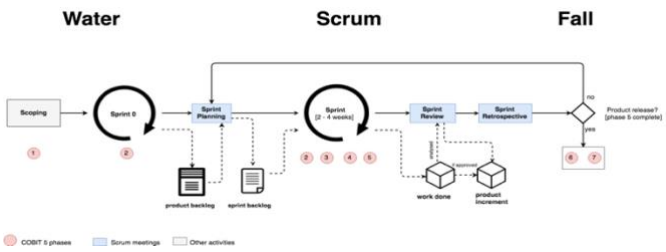

FIGURE 1. SCRUM FOR COBIT 5 LIFE-CYCLE

B. Demonstration

This section covers the step 4 of the DSR method. To demonstrate the feasibility of this solution and to prove that it can be used to improve the outcomes of COBIT 5 programmes, the authors will next discuss the results from applying it on a real programme of GEIT implementation with COBIT 5 in the Portuguese Finance Ministry.

The programme was carried out at the Portuguese Finance Ministry. With this GEIT implementation programme, they sought to obtain reliable guidance from a well-known framework, such as COBIT 5, to start their work system based on internationally accepted good-practices. This programme had a duration of 5 months with 8 sprints of 2 weeks each, and a pause for Christmas holidays of 2 weeks.

The programme was composed by 5 projects, where each of them was focused on a specific process. The processes chosen through the goals cascade tool were the following: APO02 Manage Strategy, APO03 Manage Enterprise Architecture, APO10 Manage Suppliers, BAI01 Manage Programmes and Projects.

Regarding the roles and the people involved, the initiative comprised: 1 product owner, 1 programme manager, 1 scrum master and 5 implementation team members. These members were fully outsourced, except the product owner.

For each process was established a scrum team, however, since the resources were limited in number, some implementation team members participated in more than one team. Furthermore, Scrum master was the same for every scrum team and participated in one of the projects as an implementation team member as well, being more than just a team manager, which worked out very well.

The deliverables produced were the following: the business and application layers of the enterprise architecture, the TO-BE processes designed and documented, workshops of COBIT 5 and Scrum and documentation to support the establishment of some new organizational structures.

To test the approach in a real environment, allowed the identification of some relevant issues.

First of all, the programme manager role is not part of the designed approach (explained in section 7). It was agreed, at the scoping phase of this programme, that a programme manager would be essential for managing the upper level and keep the integrity. This programme manager worked as a project manager for the programme, with scrum master responsibilities at the programme level.

Regarding scrum meetings, they were conducted at project level as established, however, since there were no

many people involved and scrum master was the same for all the projects, meetings were common for everyone. One common sprint planning, sprint review and sprint retrospective at the end of each sprint.

Considering the programme life-cycle: phase 1 began before the involvement of the rest of the team, as established by the Water-Scrum-Fall approach. This programme planning was responsibility of a joint work between product owner, programme manager and a COBIT 5 expert that is also member of the implementation team. Sprint 0 turned out to be a little mixed with the rest of the programme planning.

Phases 2 and 3 were also addressed as prior work, since the processes chosen had very low capability levels (between 0 and 1), so the target for improvement was stated at the beginning as high-level targets**.** However, for the processes of capability level 1, were conducted a series of interviews with the client to perform a more accurate assessment.

In phases 4 and 5, were planned and produced the agreed deliverables, although the high-level requirements were agreed during at scope definition, the solutions were constantly evolving during the closely contact with the customer, as foreseen by the Water-Scrum-Fall approach.

Phase 6 was out of this project scope, however is part of a second project that is starting with new requirements, where the monetarization of the processes designed and the corresponding deliverables are going to be addressed.

C. Evaluation

Evaluation phase (step 5 of DSR method) aims to measure how well an artifact supports a solution to the problem, comparing the objectives proposed to the results observed from use of the artifacts in the demonstration.

For that purpose, we will evaluate the results considering a few criteria selected from the hierarchy of criteria proposed by Prat et al.

The consistency of the artifact with each criteria will be assessed through the field study (the demonstration of the proposal instantiated on a real case). Additionally, to evaluate the correctivness of the methodology, we performed a set of interviews with experts that evaluated each element comprised in the solution and gave their opinion as practicioners.

The sub-artifacts that form the main artifact (a model and a construct) were consequently validated through the evaluation of the methodology.

Despite the fact that the solution was proved to be a simpler version of what was expected, developing these organizational changes in an iterative and incremental way was very effective overcoming some issues related with senior management commitment, resistance to change and misaligned solutions from the scope defined, as perceived by the interviews with the team involved in the demonstration, which provided satisfactory results regarding the objectives established.

Of the 6 interviewees, 100% agreed or strongly agreed that Senior Managers showed interest in the programme, acknowledging that an iterative strategy was fundamental to a greater support from top management when developing the solution (objective 1).

Also, when asked if the solution provided at the end was the one planned at the beginning, the answers were unanimous, all 6 recognized the solution was build according the client's needs and con- firmed it was adapted several times (objective 2).

Finally, although they acknowledge the client was not always convinced about the benefits, of the 6 interviewees, 4 strongly agreed and 2 agreed about the client showing a greater knowledge of the programme at the end (objective 3), which implies an improvement that interviewees related to the engagement created during the scrum mandatory meetings.

Although there is still much aspects to improve, this first iteration allowed the authors to verify that developing organizational solutions with agile methods can bring many benefits for the organization and for practitioners.

VIII. 2ND ITERATION

A. Proposal

At the end of the evaluation step of the first iteration, we stated that the results obtained addressed the defined research objectives, however, since some aspects of the methodology were criticized during the interviews with experts and perceived during the demonstration, we developed a new approach based on the results obtained from the first iteration.

After gathering insights on the first approach of a solution from experts, and analysing the results of its usage in a real environment, we decided to base our solution on SAFe and make a specification of concepts to a COBIT 5 adoption programme, similarly to the approach taken in the first iteration with Scrum.

The decision to use SAFe was sustained on the analysis performed, where it was derived the conclusion that its the most complete framework analysed and the one with the most prescriptive structure, which we expect to bring more consistency despite its complexity.

1) Specification of Scrum concepts for a COBIT 5 programme

Each Agile Team undertakes one process of COBIT 5, as decided in step 4. For this research we decided to use Scrum, so is equally valid to call it Scrum Team. In any case, each Team that will cover a specific process is composed by a Product Owner, a Scrum Master and an Implementation Team.

There is an initial Product Backlog designed according the initial high-level requirements for the whole programme and is managed by Product Manager. In the first Programme Iteration Planning, each agile team select their tasks to the upcoming PI into a Process Backlog. For the iterations (sprints) comprehended in the Programme Iteration, there is a Sprint Backlog, developed at the beginning of each sprint at Sprint Planning event, as Scrum prescribes, managed by each Product Owner.

One of the differences of SAFe from other analysed scaled methodologies, is that distinguishes an iteration of the programme from an iteration of the team. An iteration in the programme is called Programme Increment and encompasses a set of consecutive sprints.

2) COBIT 5 life-cycle

Figure 2 describes the life-cycle model.

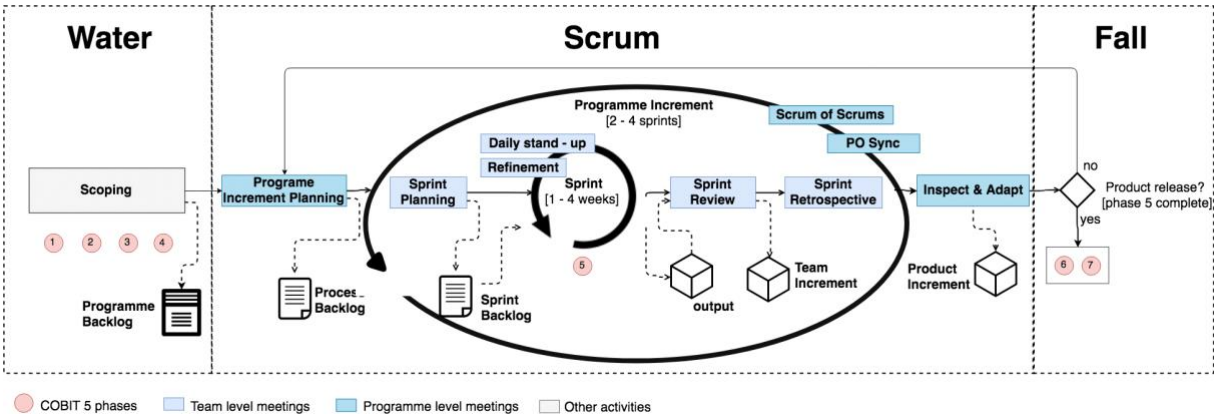

FIGURE 2. SCRUM FOR COBIT 5 LIFE-CYCLE 2ND ITERATION

B. Demonstration

The programme started in July and had a duration of 3 and a half months, with 8 Sprints of 1 week each, and Programme Increments of 3 Sprints each.

The programme was composed by 4 projects, each one focused in a specific process. The processes selected as key-processes to improve in matters of IT governance and management were: APO04, APO05, APO12 and DSS02.

Once again, the resources for this field study were limited in number, for that reason the people available for developing the solutions had the responsibility to participate in more than one Scrum Team.

We established 4 projects with only 3 implementation team members, which resulted in 3 Scrum Teams. Moreover, Scrum Master was the same for every Scrum Team and participated in one of the projects as an implementation team member as well.

To decrease the resistance to the methodology mindset, we implemented the roles as soon as the project started as well as agreed weekly and programme meetings every three weeks. Nevertheless, we only used accurately Scrum during the implementation of solutions, the difference was the ”system” was already established. To implement the system of roles and meetings at the beginning, allowed to increase the involvement of each process owner on the correspondent process from the very first step.

We felt their engagement in every decision required to first assess each capability level, then to define where we needed to go.

C. Evaluation

To evaluate the main artifact’s performance during the demonstration according the selected criteria from Prat et. al, we will use the feedback obtained through a set of semi-structured interviews to the people involved in the demonstration. To assess the quality of the life-cycle model we used the Moody and Shanks framework, and to assess the constructs created to build the rules of language, we used the Wand and Weber method.

1) Evaluation of the demonstration

For this second DSR iteration a set of 8 semi-structure interviews were performed to the people involved in the demonstration, similarly to the first iteration, to evaluate the artifact in terms of the criteria selected. The interviews were face to face and have taken an average time of 15 minutes.

Looking at the results, 71% of the interviewees agreed and 29% strongly agreed that the methodology enabled a greater involvement of process owners (senior managers). Therefore, we can conclude the **objective 1** was achieved.

Moreover, 71% of the interviewees totally agreed and 29% agreed the methodology was very useful for the people involved regarding the identification of early inconsistencies, largely because of the weekly and three weeks’ feedback loops at team and programme level, so we can consider the **objective 2** was met.

Respecting **objective 3**, we can only consider it was poorly met, since some of the people involved argued that despite the methodology forced them to be involved, they often hold back weekly meetings when more important work arise, which is a form of resistance.

Overall, the results obtained from the interviews with the people involved showed that the field experiment was way more accurate with the proposal than the first approach.

2) Moody and Shanks Framework

To assess the quality of our model artifact, we are going to use the Moody and Shanks quality management framework. With this framework for quality model evaluation, we intend to measure with identified metrics, some quality factors.

Overall, we could conclude with this evaluation that the model is **complete** for it contains all user requirements for the events and artifacts. Does not **simplify** the requirements using less elements. Is **flexible**, since is general enough to reflect any reality of implementation

with the methodology proposed, is **understandable** and **implemented** for we used it in a field study.

3) Wand and Weber

For the evaluation of the constructs' ontology we will use the Wand and Weber method (BWW-method) to analyse the effectiveness of the grammar represented in constructs. Using this method, we will be able to identify four ontological deficiencies: incompleteness, redundancy, excess and/or overload.

Overall, were found **incompleteness** and **excess** deficiencies in the grammar. The explanation is sustained on the fact that our goal was always to eliminate some elements of SAFe that are not necessary or either applicable in a COBIT 5 programme. In a similar way, the structure of COBIT 5 required non agile steps that were are not predicted by SAFe, which caused an excess deficiency.

Although we recognize these deficiencies between two sets of constructs, we also have to perceive that we build a new solution based on a few SAFe concepts and a few other practices from literature, which resulted in a specification of SAFe with additional concepts and not exactly a mapping between two frameworks.

IX. Conclusion

A. Discussion

To perform two iterations of the methodology, testing each one of them in a singular experiment of a COBIT 5 programme in different organizations, allowed us to identify some relevant findings that we are following discuss.

Regarding the first iteration, we realize that less is not always better and, in this case, a lighter approach resulted in confusion specially concerning meetings at team level, which caused a weak distinction between programme level and team level. Moreover, from the interviews with experts in Scrum and COBIT 5, we understood that the part of life-cycle that should incorporate Scrum was not a consensus for everyone, and some solutions like Sprint 0 were not very welcomed.

For the second iteration, our goal was to refine the initial solution according the findings gathered. More research was conducted in order to incorporate new scaled mechanisms to the methodology, and a new approach for the life-cycle was developed, where we eliminated Sprint 0 and limited Scrum adoption only to the implementation step, after the definition of each project scope.

Finally, the evaluation outcome confirmed the achievement of the solution's objectives by increasing senior management support, being effective regarding the detection of misalignments early in the development and reducing the resistance to change, which proved to facilitate COBIT 5 adoptions, not only from a solution' development perspective but also for communicating the benefits for the organization.

B. Limitations

- Lack of literature regarding lessons learned on COBIT 5 adoptions;
- We believe we would obtain a greater benefit if it has been done before the demonstration step, to gather more information and fill the gap in literature.
- The group of people responsible for designing and implementing the solutions was small in both demonstrations.

C. Contributions

1. A visual and detailed life-cycle model for COBIT 5 adoptions, as well as a set of definitions for each element foreseen in the solution;

2. Lessons learned and findings regarding two real COBIT 5 adoptions, enriching the literature on a subject that has such a considerable gap in literature, and that is so requested by practitioners;

3. An agile alternative to the traditional life-cycle of COBIT 5 for practitioners to use. With the demonstration experiment, we proved its applicability in COBIT 5 initiatives and that it can more effective achieving the final objectives within the schedule in opposition to a traditional approach;

D. Communication

The paper was already accepted, presented and published **2018 IEEE 22nd International Enterprise Distributed Object Computing (EDOC 2018)**:

- Amorim, Ana Cláudia et al. "Using Scrum for Implementing IT Governance with COBIT 5", IEEE 22nd International Enterprise Distributed Object Computing, IEEEXplore, 2018.

E. Future Work

- Demonstrating and evaluating the proposed methodology in more organizations, with different sizes and different teams in order to compare results and identify possible constrains;
- More research on COBIT 5 phases, seeking for new and improved ones, which was something we could not address in this thesis context;
- Integrate Lean framework as an opportunity to provide a set of principles as further guidance;
- Further research on new agile opportunities, specially regarding scaled agile methods, in order to find and test new approaches;
- Refine this methodology for the new COBIT that is expected to be released soon;
- Explore the contributions of this methodology for adopting other IT governance and management

frameworks (ex: ITIL, CMMI, etc.).

REFERENCES

[1] Melville, Kraemer, and Gurbaxani, "Review: Information Technology and Organizational Performance: An Integrative Model of IT Business Value," *MIS Q.*, vol. 28, no. 2, p. 283, 2004.

[2] S. de Haes and W. van Grembergen, "An Exploratory Study into IT Governance Implementations and its Impact on Business/IT Alignment," *Inf. Syst. Manag.*, vol. 26, no. 2, pp. 123–137, 2009.

[3] ISACA, COBIT 5: A Business Framework for the Governance and Management of Enterprise IT. 2012.

[4] ISACA, Getting Started With GEIT : A Primer for Implementing Governance of Enterprise IT. 2016.

[5] ISACA, COBIT 5: Implementation. 2012.

[6] S. De Haes, W. Van Grembergen, and R. S. Debreceny, "COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities," *J. Inf. Syst.*, vol. 27, no. 1, pp. 307–324, 2013.

[7] Manifesto for agile software development, 2001, http://agilemanifesto.org/. Last accessed April 12, 2018

[8] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.

[9] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[10] Y. Bartens et al., "A visualization approach for reducing the perceived complexity of COBIT 5," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8463 LNCS, pp. 403–407.

[11] Y. Bartens, S. De Haes, Y. Lamoen, F. Schulte, and S. Voss, "On the way to a minimum baseline in IT governance: Using expert views for selective implementation of COBIT 5," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2015–March, pp. 4554–4563, 2015.

[12] Y. H. Kwak, "Brief History of Project Management," *Story Manag. Proj. An Interdiscip. Approach*, no. 1916, pp. 1–9, 2000.

[13] J. Etzler, "IT governance according to COBIT," *Stock. Sweden R. Inst. Technol.*, 2007.

[14] M. Nicho, "Towards a Taxonomy of Challenges in an Integrated IT Governance Framework Implementation Governance Framework Implementation," *J. Int. Technol. Inf. Manag.*, vol. 25, no. 2, pp. 1–32, 2016.

[15] S. De Haes, R. Debreceny, and W. Van Grembergen, "Understanding the Core Concepts in COBIT 5," *ISACA J.*, vol. 5, no. 1, pp. 1–8, 2013.

[16] K. Schwaber and J. Sutherland, "The Definitive Guide to Scrum: The Rules of the Game," 2017.

[17] M. Špundak, "Mixed Agile/Traditional Project Management Methodology – Reality or Illusion?," *Procedia - Soc. Behav. Sci.*, vol. 119, pp. 939–948, 2014.

[18] D. J. Fernandez and J. D. Fernandez, "Agile project management - agilism versus traditional approaches," *J. Comput. Inf. Syst.*, vol. 49, no. 2, pp. 10–17, 2008.

[19] ISACA: COBIT 5 Enabling Processes, 2012.

[20] ISACA: COBIT 5 Enabling Information, 2013.

[21] C. Thiemich and F. Puhlmann, "An agile BPM project methodology," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, vol. 8094 LNCS, pp. 291–306.

[22] M. Cohn, User Stories Applied: For Agile Software Development. 2014.

[23] PMI, A Guide to the Project Management Body of Knowledge (PMBOK Guide) - 6th ed. 2017.

[24] S. Denning, "How to make the whole organization Agile," *Strateg. Leadersh.*, vol. 43, no. 6, pp. 10–17, 2015.

[25] A. M. M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," *Proc. - 2013 Int. Conf. Comput. Electr. Electron. Eng. 'Research Makes a Differ. ICCEEE 2013*, no. August 2013, pp. 160–166, 2013.

[26] D. West, "Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today," *Appl. Dev. Deliv. Prof.*, pp. 1–15, 2011.

[27] D. K. Rigby, J. Sutherland, and H. Takeuchi, "Embracing agile," *Harvard Business Review*, vol. 2016, no. May. 2016.

[28] R. S. Debreceny, "Research on IT Governance, Risk, and Value: Challenges and Opportunities," *J. Inf. Syst.*, vol. 27, no. 1, pp. 129–135, 2013.

[29] Project Management Institute, "Success Rates Rise 2017 9th Global Project Management Survey," *PMI's Pulse Prof.*, p. 32, 2017.

[30] S. Schlauderer, S. Overhage, and B. Fehrenbach, "Widely used but also highly valued? Acceptance factors and their perceptions in water-scrum-fall projects," *2015 Int. Conf. Inf. Syst. Explor. Inf. Front. ICIS 2015*, pp. 1–19, 2015.

[31] M. Laanti, "Implementing program model with agile principles in a large software development organization," *Proc. - Int. Comput. Softw. Appl. Conf.*, no. January 2008, pp. 1383–1391, 2008.

[32] Scaled Agile, SAFe, https://www.scaledagileframework.com/#. Last accessed May 22, 2018

[33] S. L. T. McGregor and J. A. Murnane, "Paradigm, methodolog34y and method: Intellectual integrity in consumer scholarship," *Int. J. Consum. Stud.*, vol. 34, no. 4, pp. 419–427, 2010.

[34] E. Hossain, M. A. Babar, and H. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review," *2009 Fourth IEEE Int. Conf. Glob. Softw. Eng.*, pp. 175–184, 2009.