Self-Organizing Is Not Self-Managing: A Case Study about Governance Challenges in an Agile IT Unit and Its Scrum Projects

Magali Simard
Université du Québec à Rimouski
magali_simard@uqar.ca

James Lapalme
École de Technologie Supérieure
james.lapalme@etsmtl.ca

Abstract

This paper presents a case study on the internal governance of Scrum projects and their relationships with their organization's governance within a rich research setting: an IT agile unit and its mature Scrum project teams. This study reveals ambiguities about the meaning of self-organizing versus self-managing, and the associated challenges for governance processes, especially those related to HR governance, which can lead to unresolved issues and conflicts. Interestingly, these ambiguities are also found in the current IS literature, which rarely differentiates self-organizing from self-managing in agile projects. Thus, this paper enhances our knowledge of governance processes and associated challenges, particularly for mature Scrum project teams, which are still little covered in the IS literature.

1. Introduction

It is commonplace today to state that agile methodologies have experienced unparalleled growth in the software engineering field [6, 4]. In addition, organizations have undergone increasing "projectification" [20] in a context where they are becoming increasingly complex and are facing unprecedented challenges in terms of limited resources for both ongoing operations and innovation [2, 10, 21]. Moreover, the development of information technology (IT) is becoming more important for organizations, so the number of IT projects is growing [12].

Agile methodologies contrast with traditional project management approaches (e.g., waterfall method) by emphasizing flexibility, embracing uncertainty, change and customer interaction, and relying on a modified project team organization [33]. However, there has been limited literature on the relationships between agile methods and the mechanisms adopted to manage software projects and the professionals involved in those projects [36]. Consequently, governance processes are still poorly covered in the literature, particularly in Scrum projects (as defined in section 2.2) and their relationships with the organization. Interestingly, Scrum is very much concerned with "how" the work must be done; it mainly covers aspects related to project management using a team design perspective, which is based on self-organizing, cross-functional teams. Conversely, many contributions from the agile community focus instead on "what" must be done [8, 32].

The empirical setting for this research is an *agile* IT unit in a finance organization in Canada. Within this unit, three teams had been using the Scrum method for their projects for three years. This setting offered a rich context for the study of governance processes including the relationship between traditional and new (agile) ways of governing. Through the research questions *What is project governance in Scrum projects?* and *How does governance facilitate or hinder Scrum projects and self-organizing?,* the goal of this study is to contribute to a better understanding of project governance in Scrum projects and of its relationship with other governance processes within the organization. Our results also suggest that ambiguity about the meaning of self-organizing versus self-managing may provoke unresolved issues and conflicts, especially in the area of HR governance.

The next section presents the conceptual background of this study, introducing the notions of governance, project governance, agile Scrum and self-organizing. The remaining sections present the research design, case study, findings, and finally, the discussion and conclusion.

2. Conceptual background

2.1. Governance

Corporate governance is the system relating to the management and control of organizations. Its structure specifies the distribution of rights and responsibilities and dictates the rules and procedures governing decision-making processes [27]. In fact, governance can be seen as a combination of processes,

HICSS

responsibilities and mechanisms to identify and reach a set of goals [31]. According to Stoker [35], the essence of the concept of governance is its emphasis on mechanisms for administering, and its ultimate goal is to create the conditions for orderly, collective action. The scope of governance is widest at the corporate level and narrower at the level of functional units, groups of projects and individual projects [24, 29].

2.1.1 Project governance. The principles of corporate governance influence projects through project governance [25]. The general purpose of project governance is to ensure that a project will meet the goals and expectations defined by various stakeholders [1]. Project governance is performed at the boundary between a project and the wider organizational context [40]. However, little has been said about how governance is designed and implemented for projects, and even less about how it is enforced through rules and/or values [26]. Moreover, questions regarding how governance systems and project systems interact or their reciprocal impacts on project execution and outcomes also remain unanswered [3, 34].

In addition, the specific topic of project governance in agile Scrum IT projects is particularly in need of further investigation. There have been few studies of the governance mechanisms adopted to manage agile software projects and the professionals involved in those projects [36]. This is especially true of projects using the Scrum method and even more so for projects with mature Scrum teams. Indeed, Scrum studies are usually more concerned with governance challenges during the implementation of Scrum itself [17].

2.2. Agile Scrum

According to Scrum.org, an authoritative body, Scrum is a method whereby people can address complex adaptive problems. At the heart of Scrum is the notion of Scrum teams, small individual teams that are highly flexible and adaptive. Indeed, Scrum relies on self-organizing teams and informal communication rather than formal controls at the organizational level or document-based communication [36].

This framework defines team roles (Product Owner (PO), Development Team, and Scrum Master) and a number of events, artifacts and rules that the team must follow. The core delivery event is called a *sprint*, which is a one-month (or less) period during which the Scrum team works. Each sprint comprises *sprint planning*, *daily Scrums*, a *sprint review*, and a *sprint retrospective*. The main artifacts used by Scrum teams are the *product backlog* and the *sprint backlog*. Thus, Scrum prescribes certain components that create regularity. What distinguishes the Scrum framework

from other contributions by the agile community is its concern with "how" (from a team design perspective) the work must be achieved instead of "what" has to be done [32]. Moreover, the literature on Scrum explicitly states that Scrum teams rely on self-organizing [14, 16]. Self-organizing refers to choosing how best to perform work, rather than being directed by people outside the team.

2.2.1 Self-organizing. The notion of self-organizing teams is quite common in the agile and Scrum literature [13]. However, this literature also contains a significant number of other terms that authors seem to use as synonyms, such as self-managing (or self-management) teams [23], self-governing teams [19], autonomous teams [16], empowered teams [23], self-regulating teams [39], self-directed teams, and self-disciplined teams [16]. Some authors explicitly state that they consider certain terms to be synonyms [23] but many do so tacitly. Moreover, Hoda et al. [13] reported that little research on self-organizing teams was available.

To conclude, the ambiguous language used by the agile and Scrum literature with regard to self-organizing teams, as well as the lack of research on the governance of agile or Scrum teams, makes it seem reasonable to ponder the question: Are all the terms used really synonyms? How are these terms (or concepts) applied in practice? And what is the impact on governance processes in Scrum project teams and between Scrum teams and the governance of the organization? The intention of this study is to help fill this gap. Consequently, the generic research questions are: *What is project governance in Scrum projects?* and *How does governance facilitate or hinder Scrum projects and self-organizing?*

3. Research methodology

3.1. Research design

This study was designed as an exploratory study with a flexible design that has embedded units of analysis [41]: an IT unit and its three agile Scrum project teams. It uses techniques such as narrative strategy, temporal decomposition and visual mapping to analyze and interpret collected data. Two selection criteria were established for the field setting. The first criterion was the presence of an IT software development unit that had been using the Agile Scrum method consistently and steadily in its projects for more than two years. The second criterion was the requirement that the identified IT unit demonstrates maturity concerning its Scrum teams and associated

2

processes through its commitment and adherence to the principles of Scrum and the sustainability of efforts to improve. The sampling for this case study was purposeful—specifically, intensity—sampling because this case provides rich information about a successful, mature implementation of Agile Scrum that has not gone to extremes. Therefore, it provides a rich example of the phenomenon [30].

This empirical exploratory study was carried out in 2015–2016 in a finance organization in Canada. This paper presents the results of the first part of the study, which focuses on the only fully agile IT unit in the organization, the EDB unit (not its real name). This unit used agile methods in all its projects and applied the agile philosophy in its management. The unit was in charge of the evolution an internal software product, the enterprise data bus. This unit had around 50 employees organized in six agile teams. Three of these teams, which were the most experienced with the agile methodology, had been assigned to EDB and had been using Agile Scrum for about three years. This study focuses on these three mature Scrum teams.

3.2. Data collection

The research data sources include semi-structured interviews, meeting observations, observations on site, documentation, a logbook, notes and memos. The sampling method for interviews and meeting observations was typical case sampling [30] in order to select participants and meetings representing different groups and points of view. Eleven participants were interviewed (Table 1) and a total of twelve meetings and nine daily meetings were observed (Table 2). The participants were considered representative of the different points of view; they had different roles, perceptions and opinions about projects and Agile Scrum. The average duration of the interviews was 1.5 hours. The documentation covered the teams' sprint metrics, Aldo status reports, Zebra processes/services and pictures, mainly taken during daily meetings (e.g., sprint boards). At the organizational level, the documentation comprised the organization chart, project management processes, strategic plan and information available from the Internet.

3.3. Data analysis

The analysis was carried out in four major steps. First, the transcripts were coded according to the interviews' chronology, followed by the coding of meeting observations, field observations and relevant documentation using a process perspective [15]. Most of the codes were adjusted to make them more contextual, and new ones were created; codes were also added *in vivo* as needed. Afterward, major themes were identified. The second step was to further analyze the documentation and triangulate findings with observation and interview data. The data analysis revealed some ambiguity in the participants' understanding of self-organizing. The following questions arose: What is the impact of this ambiguity on governance processes? And, is this ambiguity discussed in the literature? Consequently, the third step was to review the literature and the fourth step was to further analyze the data. It should be noted that, during these activities, the case history and timeline were drawn up and updated accordingly.

Table 1. Interviews

Role	Qty	Data types
Manager	3	Recording, notes
Team – Developer	2	Recording, notes
Team – PO (Product Owner)	1	Recording, notes
Team – Scrum Master	3	Recording, notes
IT PMO (Project Management Officer)	1	Notes
Organizational agile coach (Former Scrum Master in Zebra)	1	Recording, notes

Table 2. Meeting observations

Meetings	Qty	Dur	Data types
Daily meeting	9	15 m	Notes, Pictures
Planning Teams 1 and 2	1	15 m	Notes
Planning	1	2h25	Notes
Planning sprint X+1	1	1h	Notes
Grooming Teams 1 and 2	1	1h	Notes
Grooming	1	2 h	Notes
Priorization sprint X+2	1	20 m	Notes
Review meeting	3	1h	Notes
Great Mass – Zebra meeting	2	30m	Notes, Meeting PowerPoints
Meeting (Senior Director and Agile coachs)	1	30m	Recording, Notes

The validity criteria of this study are mainly met by data triangulation through the use of research data from interviews, documentation and observations. In addition, the semi-structured interviews were conducted with open-ended questions using the same detailed interview guide to ensure uniformity in the questions asked and the information gathered. NVivo software was used to code and analyze the research data. Finally, interpretations made during analysis were validated with a participant informer to prevent potential biases and distortions.

4. Case study presentation

First, the context of the IT unit is presented, followed by the description of how Agile Scrum was implemented in this unit.

4.1. Organizational context

This organization is a major player in the financial domain in Canada. At the time of the study, IT projects were usually executed the traditional way (waterfall), except for the projects of the Zebra (not its real name) IT unit. This unit, which had been created three years previously, was the only one that exclusively delivered projects using agile methods. The decision to create the Zebra unit stemmed from major recurring issues in the delivery of EDB evolution projects. Previously, the IT unit that was in charge of the EDB product had experienced chronic delays and budget overruns in its project deliveries. At one point, this situation became untenable, which led the IT VP to seek drastic solutions: the use of agile methods and the creation of the IT unit, Zebra. The purpose of this study is to focus on the projects delivered using the Scrum method in the Zebra unit.

4.2. Scrum implementation history

Originally, this unit was composed of some 15 members in two agile teams. Since Scrum is characterized by short iterations, the frequent deliveries were a real challenge because the release process was not optimized and was error-prone. Consequently, there were many failures in the beginning, which tarnished the unit's image.

During the first six months, the unit experienced different sprint durations, team sizes, and test environment tools. It also improved the release process and gave the unit's members a ramp-up period on the Agile Scrum process. In order to become self-organized teams, they had to learn empowerment, which was a real challenge; they were used to the classic command and control system: "*Before, they [Team] came to us a lot... We [managers] engaged in a lot of pushback, saying: well no, you resolve it as a team. Then, if it really doesn't work, we get involved*" *(Manager)*. Equality among team members and interchangeability were also challenging, because all members of the agile development team were considered developers, although they had diverse titles in the company. After the first six months, a third agile team was created. This team became more focused on performing small agile EDB deliveries to large projects that were executed using the traditional mode.

At the time of the fieldwork, there were still three teams in Zebra for the EDB product. Interestingly, since Zebra's creation, no employee turnaround and minimal consultant turnaround had been noted. The teams were now delivering two-week sprints and were following a continuous improvement philosophy. The Zebra director was now a senior director overseeing three directors, who mainly acted as resource managers. No team was entirely assigned to a specific director, only subsets of teams. According to the research data, team members were highly motivated and their empowerment was now taken for granted. Zebra was a success story and next goal was to improve the teams' predictability: "*Today, maturity reigns, confidence reigns, transparency reigns. And that's it, it's appreciated, it's recognized, it's even mentioned, and we've moved on to something else*" *(Scrum Master)*.

5. Findings

The findings are divided into four parts. First, the virtuous circle that was put in place during the Scrum implementation is presented, followed by the Scrum process description, the governance findings, and finally, the findings on HR governance.
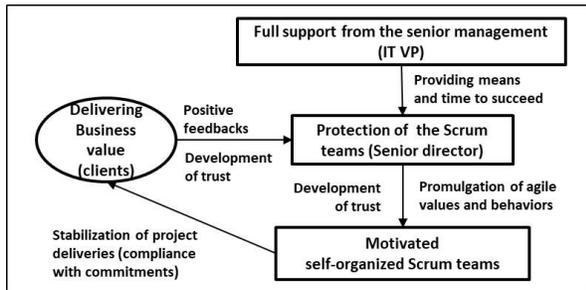
5.1. The virtuous circle of agile implementation

As indicated above, the Zebra unit was created to solve significant delivery problems with the EDB projects. Many employees who were part of the former failing projects were now assigned to Zebra. The dynamics of the change put in place had the following three characteristics:

1. Full support from senior management (IT VP): Despite the difficult situation, senior management understood that drastic changes usually mean that it may get worse before it gets better. Therefore, ramp-up times and errors were understood to be part of the learning process. In addition, sufficient budget was allocated to provide the means (Scrum consultants and training) for this change. Moreover, the senior director was allowed to select the members of the new unit, whereas these members had no choice on whether to accept or not. Some of them were reluctant to join the new unit while others were motivated by the change, "*... when it was announced [creation of Zebra unit], there were people who practically cried in meetings ... they did not want to join, and, uh, he [senior director] had a scary reputation... So there was a little apprehension. I think people didn't know him well*" *(Manager)*.

2. Constant shielding of Scrum teams: The senior director shielded the teams from politics, criticisms and problems whenever a project failure happened: "*He [senior director] acts as a very good layer of abstraction from everything political in general. So for that, we all thank him because it frees our mind from this kind of problem*" (PO).

3. Self-organized agile teams: empowerment was the first behavior developed in the Scrum teams. At the time of the field study, the teams seemed to be fully empowered with respect to decisions concerning their work, which also included the continuous improvement of their work processes.

Figure 1. Virtuous circle of agile implementation

These three characteristics acted in a virtuous circle during the agile implementation (see Figure 1) and afterward, which resulted in consistent improvements in project deliveries. This virtuous circle enabled trust to be developed between the teams and the senior director because his constant shielding of the Scrum teams fostered their self-organization and commitment to continuous improvement. All participants indicated their high motivation about working in an agile environment.

5.2. Agile Scrum process
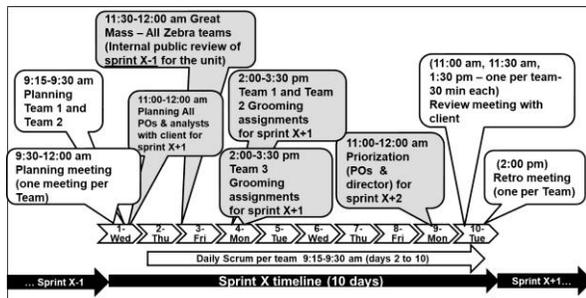
Figure 2. Timeline of meetings during a sprint

Figure 2 provides an overview and timeline of the sprint process, which was followed by the three Zebra teams. It shows their usual meetings during a two-week (10-day) sprint period. Meetings about the current sprint (X) are in white boxes, while meetings about past and future sprints (X–1, X+1, X+2) are in gray. Each box indicates the meeting's name, time/duration, participants and sprint. In addition to these meetings, there is also one meeting per sprint for the quality improvement committee, and one (potentially two) for technology improvement committees. These meetings may generate items for future sprints that would have to be prioritized during prioritization meetings.

5.3. Project governance

Our findings are presented regarding governance *in* Scrum projects, governance *of* Scrum projects, and their relationships with the organization's governance.

5.3.1 Governance in Scrum projects. In Zebra, the Scrum framework, which had been implemented with some customization, mainly follows the prescribed Scrum governance roles and mechanism. Table 3 presents an overview of the project/Scrum roles, their associated formal titles in the organization, and the meetings each role holder should attend. For roles, the main customization affected the PO role, which was divided into two roles: (1) PO: assigned to the senior director who is the EDB product owner; (2) Proxy PO (one per team): assigned to the business analysts who assure liaison between the PO, the client and the development teams.

Table 3. Project roles and meetings

Scrum roles	Formal titles	Decision types	Meetings
PO	Senior director	Strategic	Review, Daily Scrum (Optional attending) **New: Prioritization, Great Mass/ Zebra review, Operational Committee Workgroup (OCW)**
Proxy PO (New)	Business Analysts, consultant	Functional Team specific process improvements	Team Planning, Review, Retro, Daily scrum **New: Prioritization, Team Grooming assignations, Great Mass/ Zebra review,** **Team coordination meetings (teams 1 & 2): Team planning, Team grooming assignations** **Team coordination meetings (All EDB teams): Planning All POs and IT analysts**
Development team (developers)	Consultant, Analyst-programmer, Senior-analyst programmer, Solution designer, *On an as-needed basis: IT experts & IT analyst*	Technical Team specific process improvements	Team Planning, Review, Retro, Daily Scrum **New: Team Grooming assignations, Great Mass/ Zebra review** **Team coordination meetings (teams 1 & 2): Team planning, Team grooming assignations** **Daily production meeting:** one Scrum team member assigned to these meetings **Quality improvement committee:** On a voluntary basis **Technology improvement committees:** technology experts only
Scrum Master	Consultant,	None – facilitates Scrum process	All meetings except **OCW** meetings

For meetings, the main customizations were the addition of meetings to handle item prioritization, early planning, task breakout (grooming) activities and multi-team coordination (planning, team grooming, Zebra review). These meetings are highlighted in bold in Table 3. Overall, these meetings facilitate the generation of a rapid, continuous flow of micro-decisions in a context of two-week sprints.

5

Table 3 also identifies the decision types associated with each role. The following three types were identified: strategic, functional, and technological. In addition, process improvement decisions can be made for all teams or for a specific team; for example, one of the teams had set a maximum number of items to be assigned per person for grooming activities during a sprint in order to combat the tendency of some individuals on this team to overload themselves.

Finally, within project/Scrum teams, external control was closely tied to the sprint deliveries, which occurred every two weeks, the daily Scrum meetings, the sprint backlog and the review meeting. In addition, a release note is accessible for each delivery (successful or not). It presents all the changes and additions made to the product in a transparent manner; errors or inability to deliver in accordance with the commitments are clearly indicated. It is important to note that information on the teams' velocity was not distributed outside the Zebra unit and its project Scrum teams.

5.3.2 Governance of Scrum projects. For the governance of Scrum projects, there are two cases: (1) Evolution projects for the EDB product—Teams 1 and 2: In this case, the senior director acts as portfolio manager by prioritizing and identifying future deliveries with the support of the Operational Committee Workgroup (OCW) committee, which acts as a portfolio committee. (2) Sub-project of the large Aldo project (not its real name)—Team 3: The sprints' scope is not planned in advance, only the number of sprints and the budget. The Aldo functional analyst acts as the PO and prioritizes the backlog. The Aldo project is being executed in conventional mode and the results of each sprint delivery (scope, cost, duration) are included in the project status reports.

Finally, the control of projects is connected to the sprint deliveries, the review meetings and product backlogs in both cases.

5.3.3 Relationships with organization governance. The main governance relationships of IT projects are with the organization's IT governance and HR governance. For IT, the main governance mechanism is the IT management committee, which is directed by the IT VP; the Zebra senior director attends meetings, and so do the other senior IT directors. Project deliveries are reported back to the IT PMO, which aggregates all project information to report back to the IT management committee and higher levels of management. For HR governance, the Zebra unit, its projects and the project team members are aligned with the organization's goals through the performance appraisal process, as for the other units. HR

governance is further discussed in the following section.

5.4. HR governance and Scrum teams

The project Scrum teams—self-organizing teams—of the Zebra unit were unusual for this organization. Its HR governance processes were not adapted to the new way of organizing. Although this unit had to comply with the company's HR governance processes, special efforts were made to adapt these processes as much as possible to the unit's context. Three examples of these adaptations follow:

1. The directors had a resource manager role instead of a direct supervisor role, as was the case in the other units, because of the flattened hierarchy within the Zebra unit. They were in charge of a subset of employees from different teams instead of being responsible for specific teams: "*it's not … hierarchical. Yeah, it's for the HR side, managing career development … but they [employees] do not relate to them [managers] functionally … you can see it's a bit like in consulting … you have a resource manager … he is not involved in the day-to-day work*" *(Manager)*. Their primary responsibilities were performance appraisals and career development planning for their people. Interestingly, they saw themselves as part-time resource managers because they still had technical tasks to perform in the unit: "*In Zebra, there are no managers; it's a major in technological hands on, and a minor in management*" *(Manager)*.

2. The unit no longer had to submit timesheets after the employees challenged this process because they found no value in it. In fact, employees are not paid overtime when they are assigned full-time to projects. This change was considered a victory and participants reported it proudly during interviews. It also contributed to building trust in their senior director: "*they [employees] are not paid overtime … we [managers] had no arguments … we went back to the VP saying, 'Well, we think that Zebra should not use timesheets, it has no value, we don't see why. Explain to us why so we can pass the message back…' … nobody had an intelligent answer, so we don't do it anymore*" *(Manager)*.

3. The senior director and the directors did not use closed offices: they were located in the open working area with the other employees. However, for the senior director, HR policies were quite strict and he was forced to retain his office even though he did not want it. Thus, he treated it as a meeting room and a closet for his belongings: "*it's my closet… I have things, but there is nothing personal. I'm rarely here. I'm here when there are confidential things. But the office*

6

doesn't belong to me… [HR] didn't want me to get rid of my office because it wasn't normal for a senior director to have no office; but me, no, not at all." Interestingly, all the other participants identified this room as his office.

In addition, based on the research data, HR governance processes that were related to the *soft* side had gray areas that generated tension among the employees. These tensions related primarily to conflict management and performance appraisals. These two topics are discussed in the following subsections.

5.4.1. Conflicts. The Scrum framework provides a mechanism, the retrospective ("retro") meetings, which should be a safe place to discuss and decide on potential improvements, not only in processes, but also in coordination, which should include relationships between team members, and thus the resolution of conflicts. However, during interviews, participants who were Scrum team members made it clear that teams were self-organized, not self-managed: *"No, we are not a self-managed team, we are a self-organized team. So we organize for work, but there are still managers in place" (Team member).* Therefore, they did not want to act as managers by managing conflicts. They referred specifically to conflicts generated by frustrations provoked by underperformers, because other members then had to compensate in order to deliver the sprints the team had committed to. They were also reluctant to escalate these issues to the directors—to be snitches—but they were willing to give feedback, when asked, for performance appraisal purposes: *"… there were people in the team who thought I wasn't working hard enough… I would have liked to know it before the appraisal, when it happened, rather than waiting months afterward to find that there was a demerit in my appraisal. Uh, that was difficult, it was tough" (Team member).* They were also reluctant to get Scrum Masters involved because these people were not their bosses and they would push the problem back, for them to solve; moreover, since they were not managers, they considered it was not part of their role.

More specifically, conflicts between two permanent employees were the most difficult to resolve. Although conflicts with consultants were infrequent, they were considered to be easier to resolve, and Scrum Masters were sometimes involved in the resolution process. Interestingly, during interviews, this kind of ambiguity about self-organizing and self-managing versus the roles of directors, including the challenge with retro meetings, was not identified by directors. Moreover, team members felt that retro meetings were not an appropriate place for conflict resolution, while directors thought they were. Based on the research

data, conflict resolution at retro meetings was considered potentially embarrassing: *"The Scrum Master really wanted me to bring the subject up at the retro, but I didn't want to… I couldn't be sure that I wouldn't blush or lose control because it was something very disturbing. So, I didn't want to bring up the point. I wanted others to bring up the point. But it didn't happen" (Team member).* Interestingly, participants indicated that, at the beginning of the Scrum implementation, retro meetings were much more confrontational and emotional and now they were much calmer.

During the fieldwork, some unresolved conflicts were observable. For example, the suggestions of one employee, who was considered an underperformer by a senior team member, were rarely taken into account, and people changed topics quickly. During a grooming meeting, when this employee volunteered to do one item, it was then decided that another person would be added to help out; this decision was taken only in this case and without consulting the employee.

5.4.2. Performance appraisal. As indicated previously, directors had a resource manager role and each one was assigned a subset of the employees. The performance appraisal process was based on goal alignment from the top to the bottom of the hierarchy; there were personal objectives and unit objectives. Regarding unit objectives, each director had to align theirs with the Zebra senior director's, which were aligned with the IT VP's. Correspondingly, the employees had to align their objectives with those of their assigned director. In addition, since directors did not directly manage their assigned employees, they usually informally asked for feedback from the most senior team members.

The performance appraisal process also contained an employee development plan, which could generate discussions and objective setting about employees' development and career path. During interviews, some issues emerged concerning career paths because there is a limited hierarchy in an agile team and the goal is to make people as interchangeable as possible, and that was understood as leaving less room to stand out from the crowd. In addition, the organization had no career paths tailored for agile teams; they were still hierarchy-based. Moreover, being part of an agile team was not an important competency in the job market.

6. Discussion

In the next subsections, we discuss the sustainability of Scrum after a successful implementation, the challenges and ambiguities related

to self-organizing, and finally open socio-technical systems.

6.1. Scrum sustainability challenges

During Scrum implementation, strong commitment and an appropriate budget were provided, which are recognized as best practices in agile change management [4]. Another best practice relates to the importance of leaders in facilitating change [4]. Although these three factors had a big influence, the leadership factor was especially significant. Even though the leader (the senior director) had *baggage from the past*, which is far from being a success factor [4], and distrust is a common problem during agile implementations [4], he was able to overcome these challenges through the development of a virtuous circle of trust (figure 2), which still existed. Showing patience and trust and shielding Scrum teams were important ingredients in the emergence of this virtuous circle, which was much appreciated and valued by the teams. Interestingly, Moe et al. [23] also observed that team members felt more protected against external noise than before Scrum implementation. However, the sustainability of agile Scrum was still considered to be fragile even though the Scrum teams were now mature and had delivered projects successfully; agile methodologies contrast with the traditional project management approaches such as the waterfall method [33] that were in use in the rest of the organization. Thus, it was believed that a change of leader could lead to the abandonment of agile methods in Zebra.

The organization's HR-related governance processes were also part of the challenge. Interestingly, in their study on reward systems, Sun and Schmidt [36] found that salary compensation was still based on individual performance and determined by the direct supervisor in all the organizations they examined, regardless of their levels of agile methodology. This was also the case in the Zebra unit. However, from a team perspective, reward structures that emphasize individual achievement represent an incentive misalignment [18]; teams' whose reward structures are aligned with the level of task interdependence should perform better than teams with incentive misalignments [38]. Consequently, transitioning from individual work to self-organized teams requires a reorientation not only of the developers but also of governance processes. Making such changes takes time and resources, but is considered to be a prerequisite for the success and sustainability of any kind of agile method based on self-organization [22].

6.2. Challenges for self-organized teams

At the root of self-organizing is the self-coordination of work teams. According to Okhuysen and Bechky [28], there are three integrating conditions for coordination: accountability, predictability, and common understanding. In the Zebra Scrum teams, these conditions had the following characteristics: (1) Accountability: teams were accountable and empowered for their sprint deliveries. (2) Common understanding was facilitated through the prescriptive Scrum method, within which roles and responsibilities are defined. The various formal meetings also facilitated the development of a common understanding within, and between, Scrum teams. Their common physical working area was also a positive factor for this condition. (3) Predictability was identified as an element to be improved; the effort estimates for sprint deliveries were a target for improvement, although current deliveries seemed satisfactory. An important parameter was the overtime that was sometimes done to deliver according to the teams' commitments. However, this overtime was neither paid nor recorded. Interestingly, in agile systems, frequent delivery and working software are the primary concerns of the control mechanisms [36]; therefore, predictability becomes even more important for managers, especially in a context where overtime is done informally on a voluntary basis.

In addition, coordination is under persistent attack by the regular dynamics of organizations; thus, individuals and groups must constantly recreate the integrating conditions for coordination in order to jointly execute their work [28]. This implies being sensitive to the internal risks identified; informal overtime is one of them. Another risk is the informal hierarchy (and associated informal governance processes) that may emerge over time. For example, some studies have observed that team members with more experience dominate decision-making [7], which was the case in Zebra. Such dominance may also be associated with status differences, which can erode the integrating conditions and cause coordination breakdowns [28]. Because accountability requires an acknowledgement of mutual responsibilities, status differences that prevent such acknowledgement limit its development [28]. Moreover, when working interdependently, low-status individuals will ask fewer questions and give less feedback. This situation leads to less sharing of knowledge, limiting common understanding in the group [28].

Additionally, in Zebra, the ambiguity about the meaning of self-organizing resulted in unresolved conflicts. Team members explicitly stated that they were not managers and that conflict resolution was the

8

managers' responsibility; thus, retro meetings were not to be used to facilitate the resolution of these conflicts. Team members were not all positive about the retrospective meetings: "*Retro … it does not go into very human subjects like that [conflict]*" and "*I ha… I hate it [retro meeting] to death. I do not see any value, I participate, but I don't really put myself forward. And I tell [the Scrum Master] … this week again, he had us play a little game, and I said: 'look, remember that I participated'.*" *(Team member).* Negative feedback was also reported in the study by McHugh et al. [19], where most interviewees attributed little value to these meetings.

Interestingly, agile methods do not discuss interpersonal concerns such as conflict resolution, apart from providing arenas for making decisions and processes for negotiating conflicts, such as the practice of planning poker to estimate the effort that projects will require; thus, though agile principles offer little advice about cohesion, there are concrete practices that support it [5]. However, as was reported previously, the current literature on Agile Scrum is ambiguous about the meaning of self-organizing. Nevertheless, according to Schwaber and Beedle [32], the team is accorded full authority to do whatever it decides is necessary to achieve the goal. Obviously, the teams perceived a limit on this authority or did not want it at all. Conversely, the directors' perception was that self-organizing and self-managing were similar. These considerations suggest that research needs to be done on the inner workings of teams and their relationship with the rest of the organization, especially for mature and self-organizing teams [13]. Governance processes should also be investigated, which would also include clarifying the various concepts used (self-organizing, self-managing, etc.). For the latter purpose, the literature on socio-technical systems should be of interest.

6.3. Open socio-technical systems

From a socio-technical systems (STS) perspective, researchers interested in self-managed groups can capitalize on a rich heritage that goes back to work done in the 1950s [37]. Given the existence of a significant scientific literature, it is surprising to find few studies in the agile and Scrum literature that explicitly refer to it; among the few are Hoda et al. [13] and Whitworth and Biddle [39]. However, theses few studies do not discuss important concepts such as the design principles that can be used to organize groups and which have important implications for governance and management concerns.

Open Systems Theory (the most recent version of STS), as defined by Emery [9], proposes that the structures of an organization (or team) can be designed according to one of three design principles: (1) DP1: a hierarchical design in which higher levels design and control the work of lower levels; (2) DP2: a self-managing design in which work is largely designed and controlled by those doing it; or (3) *Laissez-faire*: a structure in which responsibility and accountability are unclear or incoherent. Organizations that are designed according to DP2 have fewer negative effects and higher worker intrinsic motivation than their DP1 or Laissez-faire counterparts [11]. Therefore, using these design principles as a conceptual lens, it could be said that the Zebra teams correspond to the *Laissez-faire* design because the basis of performance appraisal and HR concerns are unclear. Consequently, striving only to achieve self-organizing according to *their language* (and possibly that of the Scrum literature) could be a dead end that promotes problematic team design. Some of performance and HR issues that the teams experienced were not surprising because this literature predicts such dynamics. Thus, more research must be done to bridge the Agile (and Scrum) and STS literature.

7. Conclusion

This study has contributed to a better understanding of project governance in Scrum projects and of their relationship with the organization by examining a rich example of this phenomenon [30]. A second contribution is the finding of ambiguity regarding the meaning of self-organizing (versus self-managing), which can provoke unresolved issues and conflicts, especially related to HR governance processes. Interestingly, this ambiguity is not specific to this study but also characterizes the current literature. Further studies should investigate such ambiguities and the associated challenges. The third contribution is the proposal that the rich literature on self-managed groups (socio-technical systems) should be used to deepen our knowledge of agile and Scrum methods, which should also enhance our understanding of their potential sustainability and of governance adaptation in organizations.

8. References

[1] T. Ahola, I. Ruuska, K. Artto, and J. Kujala, "What is project governance and what are its origins?," Int. J. Proj. Manag., 32(8), pp. 1321–1332, Nov. 2014.

[2] M. Aubry, H. Sicotte, N. Drouin, H. Vidot-Delerue, and C. Besner, "Organisational project management as a function within the organisation," Int. J. Manag. Proj. Bus., 5(2), pp. 180–194, 2012.

9

[3] C. M. Burke and M. J. Morley, "On temporary organizations: A review, synthesis and research agenda," Hum. Relations, 69(6), pp. 1235–1258, 2016.

[4] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," J. Syst. Softw., 119, pp. 87–108, 2016.

[5] T. Dingsøyr, T. E. Faegri, T. Dybå, B. Haugset, and Y. Lindsjørn, "Team performance in software development," IEEE Softw., 33, pp. 106–110, 2016.

[6] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," J. Syst. Softw., 85(6), pp. 1213–1221, 2012.

[7] M. L. Drury-Grogan, K. Conboy, and T. Acton, "Examining decision characteristics & challenges for agile software development," J. Syst. Softw., 131, pp. 248–265, 2017.

[8] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," Inf. Softw. Technol., 50, pp. 833–859, 2008.

[9] Emery, M., "The current version of Emery's open systems theory," Syst. Pract. Act. Res., 13(5), pp. 623–643, 2000.

[10] J. Geraldi, H. Maylor, and T. Williams, "Now, let's make it really complex (complicated): A systematic review of the complexities of projects," Int. J. Oper. Prod. Manag., 31(9), pp. 966–990, 2011.

[11] D. W. de Guerre, M. Emery, P. Aughton, and A. Trull, "Structure underlies other organizational determinants of mental health," Syst. Pract. Act. Res., 21, pp. 359–379, 2008.

[12] L. K. Hansen, P. Kræmmergaard, and L. Mathiassen, "IT project portfolio governance practice: An investigation into work design problems," J. Inf. Technol. Case Appl. Res., 19(2), pp. 81–101, 2017.

[13] R. Hoda, J. Noble, and S. Marshall, "Self-organizing roles on agile software development teams," IEEE Trans. Softw. Eng., 39(3), pp. 422–444, 2013.

[14] Y. Kim, "Analyzing Scrum agile software development with development process, social factor, and project management lenses," In AMCIS 2007 Proceedings, 2007.

[15] A. Langley, C. Smallman, H. Tsoukas, and A. H. Van den Ven, "Process studies of change in organization and management: Unveiling temporality, activity, and flow," Acad. Manag. J., 56(1), pp. 1–13, 2013.

[16] G. Lee and W. Xia, "Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility," MIS Q., 34, pp. 87–114, 2010.

[17] I. Lehto and K. Rautiainen, "Software development governance challenges of a middle-sized company in agile transition," in ICSE Workshop on Software Development Governance, pp. 36–39, 2009.

[18] L. M. Maruping, V. Venkatesh, and R. Agarwal, "A control theory perspective on agile methodology use and changing user requirements," Inf. Syst. Res., 20(3), pp. 377–399, 2009.

[19] O. McHugh, K. Conboy, and M. Lang, "Agile practices: The impact on trust in software project teams," IEEE Softw., 29(3), pp. 71–76, 2014.

[20] C. Midler, "'Projectification' of the firm: The Renault case," Scand. J. Manag., 11(4), pp. 363–375, 1995.

[21] D. Miller, "Disruptive texts: Case narratives as research inspirations," Acad. Manag. Rev., 42(1), pp. 154–164, 2017.

[22] N. B. Moe, T. Dingsøyr, and T. Dybå, "A teamwork model for understanding an agile team: A case study of a Scrum project," Inf. Softw. Technol., 52(5), pp. 480–491, 2010.

[23] N. B. Moe, T. Dingsøyr, and T. Dybå, "Understanding self-organizing teams in agile software development," Proc. Aust. Softw. Eng. Conf. ASWEC, no. 3, pp. 76–85, 2008.

[24] R. Müller, L. Zhai, A. Wang, and J. Shao, "A framework for governance of projects: Governmentality, governance structure and projectification," Int. J. Proj. Manag., 34(6), pp. 957–969, 2016.

[25] R. Müller and L. Lecoeuvre, "Operationalizing governance categories of projects," Int. J. Proj. Manag., 32(8), pp. 1346–1357, 2014.

[26] R. Müller, S. Pemsel, and J. Shao, "Organizational enablers for governance and governmentality of projects: A literature review," Int. J. Proj. Manag., 32(8), pp. 1309–1320, 2014.

[27] OECD, OECD Principles of Corporate Governance. Paris: OECD, 2004.

[28] G. A. Okhuysen and B. A. Bechky, "Coordination in organizations: An integrative perspective," Acad. Manag. Ann., 3(1), pp. 463–502, 2009.

[29] M. O'Neil, "Labour out of control: The political economy of capitalist and ethical organizations," Organ. Stud., 36(12), pp. 1627–1647, 2015.

[30] M. Q. Patton, Qualitative Research & Evaluation Methods, 3rd ed. London: Sage Publications Ltd, 2002.

[31] S. Pemsel, A. Wiewiora, R. Müller, M. Aubry, and K. Brown, "A conceptualization of knowledge governance in project-based organizations," Int. J. Proj. Manag., 32(8), pp. 1411–1422, 2014.

[32] K. Schwaber and M. Beedle, Agile Software Development with Scrum. Upper Saddle River, NJ: Prentice Hall, 2002.

[33] P. Serrador and J. K. Pinto, "Does Agile work? A quantitative analysis of agile project success," Int. J. Proj. Manag., 33(5), pp. 1040–1051, 2015.

[34] M. Simard, M. Aubry, and D. Laberge, "The utopia of order versus chaos: A conceptual framework for governance, organizational design and governmentality in projects," Int. J. Proj. Manag., 36(3), pp. 460–473, 2018.

[35] G. Stoker, "Governance as theory: Five propositions," Int. Soc. Sci. J., 50, no. 155, pp. 17–28, 1998.

[36] W. Sun and C. Schmidt, "Control mechanisms and agile methodology use: Data from the industry," J. Comput. Inf. Syst., 58(3), pp. 234–243, 2018.

[37] E. Trist, "The evolution of socio-technical systems," Occasional Paper, No. 2, 1981.

[38] R. Wageman, "Interdependence and group effectiveness," Admin. Sci. Quart., 40, pp. 145–180, 1995.

[39] E. Whitworth and R. Biddle, "The social nature of agile teams," in Proceedings of the AGILE 2007, Washington, DC, pp. 26–36, 2007.

[40] G. M. Winch, "Three domains of project organising," Int. J. Proj. Manag., 32(5), pp. 721–731, Jul. 2014.

[41] R. K. Yin, Case Study Research: Design and Methods, 5th ed. Thousand Oaks, CA: Sage Publications Inc., 2014.