



Agile Autonomous Teams in Complex Organizations

Marius Mikalsen^{1,2(✉)}, Magne Næsje², Erik André Reime²,
and Anniken Solem^{1,2}

¹ SINTEF Digital, Trondheim, Norway
marius.mikalsen@sintef.no

² Norwegian University of Science and Technology, Trondheim, Norway

Abstract. In complex organizations, the effective functioning of autonomous teams is challenged by the need to coordinate and align work with multiple experts and other units in the organization. We report on the challenges experienced in an agile program consisting of cross-functional teams set up with resources from both the IT and business development side of the organization, while team members simultaneously remain in their line organization. Through an empirical case study of the agile program, we find that the production structure (i.e. the distribution of operational tasks) and the control structure (i.e. managing activities related to the operational task) influence agile team autonomy. We contribute by pushing past describing dependencies in terms of coordination challenges and mechanisms. To do this, we use modern sociotechnical theory to discuss how a production structure with many dependencies cause challenges and how a misaligned control structure is time-consuming and reduces team autonomy.

Keywords: Agile · Autonomous teams · Complex organizations · Case study · Modern sociotechnical theory

1 Introduction

With increased digitalization, organizations face rapid changes in customer demands, changing markets, and continuous technological advancements. It forces established, traditional non-agile, complex organizations to make their software development more agile. By complex, we depict an organization with many units, with dependencies amongst them. The number of units and the strength of the dependencies determines the complexity. Traditionally, bureaucracy, plans, and hierarchies have addressed complexity. Agile ways of working are different, and extant research has not sufficiently addressed the issue of introducing agile autonomous teams within complex organizations [1, 2]. Introducing agile teams in a complex organization will cause challenging dependencies on condition that the surrounding organization remains plan based and hierarchical [3]. However, our knowledge of how such dependencies influence the autonomy of agile teams is limited, hence our RQ: *How is complexity in organizations influencing the autonomy of agile teams?*

We answer this question through an empirical case study of an agile program (henceforth referred to as AP) consisting of autonomous agile teams within a bank. The teams in AP are cross-functional, being staffed with personnel from both business and IT units. The number of dependencies between people, tasks, knowledge, technical assets, and other resources makes this a complex organization.

To analyze our findings, we use modern sociotechnical theory (henceforth referred to as MST). MST is concerned with meeting dynamic business environments by designing flexible organizations. MST problematizes structural issues of production and control, and in this case, how these influence agile autonomous teams.

2 Theoretical Background

Agile and Autonomous Teams

Team autonomy and diversity is reported to be key in achieving agility [4]. The process of forming and implementing teams with high autonomy, as well as the effective functioning of such teams, are not yet adequately addressed and understood in the context of complex organizations [5]. To understand autonomous teams in complex environments, it is crucial to understand the organizational context surrounding the team as it is an important determinant of effectiveness [6], effecting the potential for autonomy. Thus, there is a need for new knowledge on how to organize for and utilize autonomous teams, in order to attain better performance, productivity, innovation and value creation, hence increase competitiveness.

Agile autonomous Teams in Complex Organizations

While agile and autonomous teams have shown success in smaller projects, introducing such teams in complex organizations is known to be challenging [1]. Complex organizations are more challenging because there are many dependencies, which traditionally have been controlled through plans, hierarchies and standardization [2]. Dependencies exist between agile teams, and from agile teams to the rest of the organization [7]. As agile and autonomous teams are introduced within complex organizations, it may be problematic to scale agile coordination and communication practices from the team level, as such a change requires a change in organizational structure and processes [8].

When introducing agile teams in an organization that is otherwise traditionally organized, this can imply that the teams are being restricted by the rest of the organization operating in plan and waterfall mode [8]. This has led some suggest that it is necessary that all departments “transform” if one is to gain full benefit from agile [3]. It seems that new insight on how to deal with complex interactions and dependencies is required [7, 9], as discussed next.

Modern Sociotechnical Theory (MST) as a Lens on Dependencies

The motivation behind modern sociotechnical theory (MST) is meeting dynamic business environments (e.g. rapidly changing user behaviors and technology) by designing flexible organizations. A flexible organization is achieved by creating complex jobs within simple organizations [10]. This approach is similar to the thinking

behind agile ways of working and autonomous teams. Bureaucratic organizations, due to their focus on maximum division of labor and central control over core work processes, do not fare well in dynamic environments [11].

Originating from a tradition focusing on how organizations are structured, MST puts emphasis on how structural complexity (e.g., the degree of team dependency towards other teams and functional units) in organizations influences team autonomy and production [12]. The theory describes how disturbances (i.e. unplanned events) originate from dependencies that can be traced to the configuration of the organizational structure.

MST divides the organizational structure into production and control structure [10]. The production structure describes the distribution of operational tasks across actors, and their relation (e.g. a business developer and a software developer cooperating to develop a new feature). The control structure describes the distribution of regulatory tasks, that is, managing activities related to the operational task (e.g. a PO managing the above-mentioned development). A job entails a set of operational and regulatory tasks.

When the handling of disturbances to production involves actors outside the task (such as a manager), it is considered an external regulation. When the handling is within the task without interfering with the environment, it is considered an internal regulation. At the core of MST is a two-step structural redesign [11]. The first step is to create a production structure with few dependencies that minimize disturbances. Second, the goal of the control structure is to decentralize decision making-authority, creating an amplified regulatory potential (i.e. ability to perform regulatory tasks) that is aligned with operational tasks (e.g. enabling teams' opportunity to handle remaining disturbances). MST has concepts for understanding organizational structures, dependencies, and disturbances. Thus, it can be used to deepen our understanding of agile autonomous teams established in complex organizations.

3 Case and Method

Case Background

This study is a part of a longitudinal interpretative case study [13] of agile autonomous teams set in a Norwegian bank (dubbed NorBank for anonymity), with more than 2,000 employees. NorBank initiated in 2017 an agile program (AP) consisting of five cross-functional autonomous teams organized in line with agile principles, with the goal of developing improved software for their business-to-business solutions in the insurance market. The teams consist of resources from both the IT and business development side of the organization. A product owner (PO), who is responsible for realizing the team's delivery and goals, as well as managing the customer relationship, leads each team. Importantly, and which is often the case in complex organizations, members of the teams remained organized in a line organization (such as the IT and business side) and partake in several projects simultaneously. Therefore, the teams are staffed with part-time resources, particularly from the IT side. AP is managed by a group consisting of relevant IT and business managers from NorBank units which AP interfaces with. The

POs and a business architect also participate in this management group. NorBank has a set of legacy systems to which the solutions developed by the AP depends.

Data Collection and Analysis

The empirical data was collected in four rounds, using semi-structured interviews, observations of retrospectives carried out in AP and feedback sessions (Table 1). In order to qualitatively analyze our empirical findings, we used the stepwise deductive-inductive method (SDI) [14]. The inductive purpose of the SDI-method is to move from raw data, through categories, to concepts or theories. This is done by first encoding (i.e. writing words or phrases that describe paragraphs or even smaller sections of the data material) the data in a manner that retains the details of the original material. The next step is to systematize the codes that are relevant to the research question into categories. Finally, concepts are developed by applying relevant theory to the categories.

To increase the validity of our research, two of the researchers that created the codes first carried out the data coding individually. The data from the first round of the data collection resulted in 383 codes, and the second round gave 231 codes. The “NVivo 12 Pro”-software was used in the coding process. After the coding, the researchers created inductive categories based on the codes. This resulted in 21 categories for the first round and 12 categories for the second. By applying the MST lens, the findings were organized into two main categories, production structure and control structure. Combined, these categories provide insight on the dependencies occurring with agile teams in complex organizations. The purpose of the two last rounds of data collection was to saturate the categories. Feedback sessions were used to present the intermediary results to the case.

Table 1. Collected data

Data sources	Participants
25 interviews	Head of AP, IT managers, business managers, business architect, POs, IT developers, IT analysts and business developers
4 observations	Management group meeting and retrospective (2), PO meeting and retrospective, and a team retrospective
2 feedback sessions	Head of AP and business manager, AP’s management group

4 Findings

Production Structure: AP has dependencies to its surrounding non-agile organization, thus resulting in a complex production structure. The dependencies are in relation to the IT and business units in NorBank, as well as personnel working on core and business systems on which AP systems rely.

AP’s dependency towards the IT-side of the organization proved particularly challenging, as AP uses shared IT-resources. While IT-resources work in AP, they also have tasks in other projects and in their own units. When one PO is asked about the effects of

having shared resources, the answer is: *“Having resources that aren’t 100% (dedicated) is clearly a challenge. [...] If they have line tasks, then their mind is on that and not the project (AP), it is something we have to work with.”* A team member in AP also reflects this diverging focus: *“I work 50%, 50%, 40% (distributed between AP, department and another project). [...] It’s chaotic working with so much at the same time. [...] You don’t feel like you get to do your job satisfactorily.”* AP’s dependency towards the IT-departments through shared resources gives rise to structurally conditioned disturbances. A PO illustrates this disturbance: *“Often there is something that, ‘we just have to fix tomorrow’ or customer issue or that kind of thing that is prioritized over everything else. The consequence is that you get fewer resources in the flows (AP teams).”*

AP develops software solutions meant for business-to-business units of NorBank, such as sales and customer service. The business side knows the customer, and is sometimes the customer of the solutions developed by AP. AP needs to mobilize resources from the business side to understand the software’s functional requirements. A PO describes this process: *“When we start developing a new module, we assess who is to be included in a workgroup (from the business side), lead by one of our (the teams’) business developers. We have to fetch those who are on the phone with customers.”* A manager from the business side indicates that such work is not always prioritized: *“Many (POs) wish that more (from the business side) were hands-on and involved. [...] But we are not there that it is prioritized.”*

There are also dependencies to personnel with knowledge and responsibility for the core and business systems. These systems constitute the underlying IT-infrastructure, on which business systems are built. A PO indicates how this can be time-consuming: *“This takes time. [...] We, together with [a core team], develop [a new system], but you also have those who work with the business systems. [...] They have completely different prioritizations.”*

In sum, AP’s production structure has many dependencies to the surrounding organization, which as shown have diverging prioritizations. This creates disturbances that influence the AP-teams’ ability to be autonomous and agile. Thus, the production structure gives rise to challenges regarding who controls the resources.

Control Structure: AP’s control structure functions so that AP teams, through their respective POs, have a regulatory potential that is internally high, and externally low, as illustrated below.

The POs have a high regulatory potential internally through trust from the management group. Due to the dedication of internal decision-making authority to POs, the AP-teams have adapted their work methods and a coordination layer within the AP has been removed. An IT-manager describes trust as an important factor: *“I think it’s an important factor for the teams and those leading the teams that we [the management group] give them trust, in regards to us being confident that they work with the right stuff.”* One of the POs also expresses this trust: *“I feel that we are very autonomous in the team. If we want to do it this way or that way, it’s just up to us to decide it.”*

Externally, on the other hand, the POs are responsible for handling several dependencies. When asked about the POs negotiation power and understanding of the surrounding organization, a manager from the business side answers: *“It is a very important factor. [...] You are supposed to deliver something to many (stakeholders).”*

Or, a lot of people that care. The maneuvering of the stakeholder map, in that way, is demanding.” The low external regulatory potential is indicated by the fact that the POs themselves do not have control over the resource allocation from the IT side, the priorities of the business side, and resources in core and business systems. A PO illustrates the lack of control over resource allocation from the IT side: *“The only thing I can’t do anything about is the team members and how many percent they are allocated to the team. That is, in a way, beyond my control.”* Another PO describes how this takes time: *“The project took a lot of time last year before we had any progress because we were staffed wrong.”* An extra dimension to the team challenge with shared resources, is that the employees are highly autonomous in their work prioritization, as one IT-analyst confirms when asked about who decides where work should be done: *“There is nothing given at all. [...] I decide myself.”* The above illustrates how POs need to handle both IT-managers and –resources in AP’s dependency to IT units.

A PO describes how the dependency towards the business side and their prioritization cause disturbances hindering the AP-teams’ effectiveness: *“If you meet a service-minded person, like we did when needing [type of insurance], it is done ‘in seconds’. And if you meet someone different, like we did [another type of insurance], it takes like a month.”*

The dependency on core and business systems personnel is also a hindrance, according to one PO: *“What also limits our effectiveness is unstable test systems. [...] the test systems were down for 1,5 week. [...] I felt that no one listened when I brought it up.”*

In sum, the above indicates how the control structure does not give enough external regulatory potential to enable handling of disturbances. Thus, the control structure does not align with the complex production structure.

5 Discussion

Extant research shows that there are challenges related to dependencies between agile autonomous teams and a surrounding complex organization [1, 7]. However, empirical insight on the particularities of these challenges is scarce. Driven by our research question - how is complexity in organizations influencing the autonomy of agile teams – we have reported findings from an empirical case study of a bank that has established an agile program consisting of autonomous cross-functional teams within a complex organizational setting. Our findings indicate how production structure and control structure influence the autonomy of teams. We find how the AP has a high degree of internal regulatory potential (i.e. shaping the inner life of the program), while the external regulatory potential is lower (e.g. regulating access to resources). These findings contribute to the existing literature on agile autonomous teams in complex organizations as we discuss.

First, in terms of production structure, we find that not all production resources are embedded in the AP, which leads to a reliance on shared resources. This creates external dependencies, which we find are time-consuming to handle and hence reduce the autonomy of teams. This adds to the findings from [7], as a form of negotiation that takes place between autonomous teams and the surrounding organization. While having full-time resources dedicated to teams is common in many organizations, we expect that having shared production resources with many dependencies is more common in complex organizations adopting agile methods. MST [10] suggests reducing such dependencies by organizational redesign moving more resources that are full-time into the autonomous teams. Such redesign of organizational structures is also what Dikert et al. [3] suggest. Changing the surrounding organizational structure to fit agile, is somewhat different from Barlow et al. [2], who suggest tailoring methods depending on the size of the project and the nature of the dependencies. We cannot conclude on what is the best course of action, if there exists one, but illustrate how there are different ways of approaching the challenges of dependencies.

Second, we find that AP has a high level of internal regulatory potential by having sufficient levels of control of the inner workings of the agile program, such as how work is organized, what tasks that are prioritized etc. However, the dependence on resources that are outside of AP, such as shared IT resources, lowers the external regulatory potential. Low external regulatory potential, our findings show, is time-consuming and limits the autonomy of the teams in AP. This is in line with other research on how plan based and waterfall managed surroundings is a restricting factor upon interfacing agile teams [8]. Some suggest that it is necessary that all departments transform towards agile if one is to gain the full benefits [3]. An additional insight from MST [10] is that the degree to which external dependencies influence autonomous teams' production capacity is dependent on the level of control the teams have in terms of regulating disturbances. With shared resources, control, i.e. the ability to regulate potential disturbances, may well be, at least partially, outside the agile autonomous teams' control, and potentially influence production capacity.

Finally, there are also some practical implications we can draw from our analysis. First, according to MST [10], creating such agile programs should start by creating a production structure with as few dependencies as possible. One way to do this is to, as far as possible without negative consequences to the rest of the organization, staff the agile programs with full-time resources. Second, if the organization cannot provide full-time resources into agile programs, it would be beneficial if the units having resources in the program also use agile ways of working. Then the shift from working in the agile program and the units will be less time-consuming. Third, resources working on core systems often become bottlenecks and create time-consuming dependencies. One way to remedy this would be to aim to give a sufficient level of resources experience and training in such systems. This involves giving developers enough time to develop in-depth knowledge of core systems.

6 Conclusion and Recommendations for Future Work

In this paper, we have analyzed the establishment of agile autonomous teams in a complex organization. A perspective on the alignment of production and control structure relating to dependencies and disturbances can provide new insights on agile team autonomy in complex organizations. First, it would be interesting to investigate the alignment of production and control structures in organizations that have done full-scale agile transformations, that is, moved towards fully autonomous teams. Such studies could be used for comparison towards cases where only parts are transformed, as reported in this case. Second, we have only begun to describe the role of production and control structures relating to agile autonomous teams in complex organizations, so there is a need for more in-depth studies to flesh out this phenomenon. Third, in our analysis, part-time resources is a prime source of disturbances. However, it will be necessary to investigate other sources of disturbances, such as technical dependencies, and see how they can be controlled. Finally, it seems relevant to investigate the level of control necessary for teams in order to regulate disturbances and be able to act autonomously.

References

1. Kruchten, P.: Contextualizing agile software development. *J. Softw. Evol. Process* **25**(4), 351–361 (2013)
2. Barlow, J.B., et al.: Overview and guidance on agile development in large organizations. *Commun. Assoc. Inf. Syst.* **29**, 25–44 (2011)
3. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* **119**, 87–108 (2016)
4. Lee, G., Xia, W.: Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Q.* **34**(1), 87–114 (2010)
5. Mathieu, J., Maynard, M.T., Rapp, T., Gilson, L.: Team effectiveness 1997–2007: a review of recent advancements and a glimpse into the future. *J. Manag.* **34**(3), 410–476 (2008)
6. Hackman, J.R.: The design of work teams. In: Lorch, J. (ed.) *Handbook of Organizational Behavior*, pp. 315–342. Prentice Hall, Englewood Cliffs (1987)
7. Mikalsen, M., Moe, N.B., Stray, V., Nyrud, H.: Agile digital transformation: a case study of interdependencies. In: *Thirty Ninth International Conference on Information Systems*, San Francisco (2018)
8. Khan, M.R., Fernandez, W.D., Jiang, J.J.: Is there such a thing as agile IT program management? In: *International Research Workshop on IT Project Management* (2016)
9. Rolland, K.H., Fitzgerald, B., Dingsøyr, T., Stol, K.-J.: Problematizing agile in the large: alternative assumptions for large-scale agile development. In: *37th International Conference on Information systems (ICIS)*, Dublin (2016)
10. de Sitter, U., Den Hertog, J.F., Dankbaar, B.: From complex organizations with simple jobs to simple organizations with complex jobs. *Hum. Relat.* **50**(5), 497–534 (1997)
11. van Amelsvoort, P., van Hootegem, G.: Towards a total workplace innovation concept based on sociotechnical systems design. In: Oeij, P., Rus, D., Pot, F.D. (eds.) *Workplace Innovation*. AHSW, pp. 281–299. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56333-6_17

12. Vriens, D., Achterbergh, J.: Cybernetically sound organizational structures I: de Sitter's design theory. *Kybernetes* **40**(3/4), 405–424 (2011)
13. Klein, H.K., Myers, M.D.: A set of principles for conducting and evaluating interpretative field studies in information systems. *MIS Q.* **23**(1), 67–88 (1999)
14. Tjora, A.H.: *Kvalitative forskningsmetoder i praksis*. Gyldendal akademisk, Oslo (2012)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

