BerkeleyHaas
Haas School of Business
University of California Berkeley

When Agile Harms Learning and Innovation:
(And What Can Be Done About It)

California Management Review
2020, Vol. 63(1) 61–80
© The Regents of the
University of California 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0008125620948265
journals.sagepub.com/home/cmr
SAGE

Maria Carmela Annosi[1], Nicolai Foss[2], and Antonella Martini[3]

SUMMARY
Originally developed for software development, Agile approaches are increasingly adopted by organizations that seek flexibility in the face of rapid change. However, little attention has been paid to the potential negative consequences of the implementation of Agile in large-scale settings. This article presents the results of a multi-site study of a multinational telecom company over five years during its implementation of Agile practices in the context of large-scale software development. The article points to six potential pitfalls of implementing such practices that may negatively influence individual learning, ideation, and exploitation capabilities. It offers advice on how to avoid these consequences in large, established firms.

KEYWORDS: agile, large firms, organizational learning, ideation, exploitation

Managers are increasingly under pressure to build organizational flexibility in increasingly complex and dynamic environment. In response, new, adaptable organizational forms and practices are being deployed. Thus, companies such as CISCO, 3M, Microsoft, GE, Ericsson, SAAB, and SAP seek to increase organizational flexibility by adopting new organizational forms; migrating from vertical to lateral, organization-wide communication; moving from traditional hierarchy toward networks; and substituting from formal cooperation and coordination toward more spontaneous forms of interaction. Almost all of them have adopted Agile Scrum practices, which are widely viewed as reliable means of speeding up

[1]Wageningen University & Research, Wageningen, The Netherlands
[2]Copenhagen Business School, Frederiksberg, Denmark
[3]University of Pisa, Pisa, Italy

project development and execution. However, a desire to boost innovation is also often a key driver of organizations' adoption of Agile approaches.

Formally initiated in 2001 with the *Manifesto for Agile Software Development*,[1] the Agile methodology (henceforth, "Agile") is a set of recommendations for a more adaptive and efficient approach to software production that now enjoys widespread use beyond its original software development context. Thus, Agile is often seen as a general project management toolbox that contains methods and practices—such as backlogs, scrum boards, workflows, sprints, stand-up meetings, and velocity charts—and has implications for project- or team-level choices, task-management configurations, modes of team collaboration, and metrics and reporting.

Although Agile was initially designed for use in small, team-based software-development projects,[2] they are being increasingly adopted in large-scale settings[3] in large organizations.[4] While the practitioner literature acknowledges that Agile can be difficult to implement in larger settings,[5] very little research exists on this issue.[6] However, in large-scale settings, the implementation of Agile presents distinct challenges. For example, while the main focus in Agile is on intra-team interaction issues, inter-team coordination issues naturally emerge in large settings. The problem is that in small Agile settings, everyone involved is expected to oversee everything, while in larger projects with many agile teams, not everyone will understand the big picture. Such challenges have thus far been neglected in the literature, and little is known about their occurrence or size. This is potentially problematic, because of increasingly widespread adoption of Agile in large-scale settings, including British Telecom (14,000 employees), SAP AG (18,000 employees), Yahoo! (150 teams), KeyCorp (1,500 employees), and Ericsson (95,000 employees).[7]

In the following, we examine a concrete case of large-scale adoption of Agile. Over five years we followed the implementation, use, and consequences of Agile Scrum methods in a major multinational telecommunications company. At the time Agile was implemented, the overall strategy of the company was to bolster leadership in its core businesses by stepping up learning and innovation efforts. Implementing Agile across R&D units in the company was expected to reduce R&D costs and improve efficiency, but also to help with reaching innovation goals, notably with respect to continuous upgrading of core products.

Our findings suggest that the innovation goals were not achieved. The pursuit of operational efficiency with Agile Scrum practices harmed individual learning, reducing the ability to continuously renew the product portfolio and upgrade the company's existing products.[8] As team members increasingly lost track of product code, they experienced increasing difficulties with respect to implementing customer requirements. We discuss the ways in which senior managers can manage the potentially conflicting demands of pursuing efficiency while facilitating learning and ideation under Agile.

A Case of Large-Scale Agile Transformation and Implementation

Implementing Agile Methods in Company E

To gain insights into the large-scale implementation and outcomes of Agile, we closely observed over a five-year period (2012-2017) the implementation, execution, and consequences of Agile Scrum practices in large-scale software development settings in a major multinational corporation operating in the telecommunications sector, Company E. We interviewed more than 100 informants at the team level, middle management, and top management levels and added analysis of documents and archival records from the involved R&D units and relevant intra- and inter-unit community groups to corroborate and augment evidence from our interviews and observations.[9]

At the time of Company E's transition to Agile, the pace of change in the telecommunications market was increasing, as the telecom, IT, and media industries were converging into a broader ICT industry. The company decided to step up investments in its core product areas, involving an increased focus on software production and professional services. Company E also introduced programs to improve its marketing success by better satisfying customer needs. The adoption of Agile was also seen as a means to this end.

Our research particularly focused on agile implementation in eight large R&D units of Company E, which involved thousands of programmers and software developers, hundreds of middle managers,[10] and several senior managers. These R&D units were dedicated to the core products of Company E and all of them had transitioned to post-bureaucratic structures, abolishing hierarchies and embracing a flatter organizational model based on self-managing teams. (The firm also had R&D units devoted to more radical R&D in which Agile was not implemented.)

Agile was implemented because senior management wanted to better adapt to customer needs, increase the speed of software deliveries, and reduce the costs of software implementation. To support Agile organizationally, senior management changed the organizational design to allow for cross-functional teams and flatter structures that were believed to increase the speed of decision making. Thus, in the eight R&D units, team members with different backgrounds and experiences were asked to work together in a cross-functional manner. Internal product experts, who were embedded in cross-functional teams and participated in software development with their peers, were only occasionally involved in other teams' product-design choices, whereas prior to the transition to Agile, their involvement had been mandatory and their suggestions were always implemented. Moreover, team members were explicitly told that they should not invest too much time in updating the project documentation, as the agile product-development process minimized handovers by allocating the design and development of pieces of software to the same team.

To compensate for this "lightweight" approach, team members were told to keep product code simple, and, hence, easily understandable. Similarly, they were instructed to respect the software architecture of their products as much as possible, as the intent was to use the system architectures to enhance inter-team coordination. To make team members better software developers, senior managers introduced the craftsmanship system, which was essentially a new approach to peer-to-peer training that engineers working in the agile teams needed to acquire.

To coordinate activities across teams, stakeholders of agile teams (such as product owners) divided work packages into smaller, independent functional items, which allowed them to allocate the items to different teams during each iteration cycle and assemble the developed parts at the end of each iteration. Management supported the Agile transformation process through a clear and consistent communication strategy that emphasized the main advantages of the application of Agile, including (1) an improved customer interface, which would lead to more frequent releases and more responsiveness; (2) an increased motivation for team members, as they expanded their competences beyond their core expertise; and (3) an improved IT infrastructure for test automation with the potential to improve product quality.

In some ways, the large-scale Agile transformation in Company E was successful. In some cases, R&D units succeeding in reducing end-to-end lead time by 60% and maintenance costs by 40%. However, the transformation also had consequences that were neither anticipated nor desired by Company E's management.

Agile Challenges in Company E

Prior to the transition to Agile, Company E had emphasized the reuse of software assets, such that the development of new functionalities made systematic use of a core set of software assets, which contributed to realize economies of scale and scope. However, this also meant that newly developed software had to take into account a variety of existing assets that reflected past requirements and pre-defined architectural designs which necessitated a high degree of familiarity with the existing code. However, during the three-week iteration cycles, developers could only gain a temporary and weak understanding of the relevant part of the code. At the same time, the likelihood that this knowledge would be reused during the next iteration was remote.

The software products developed by Company E consisted of a mix of components produced in-house and components purchased from outside developers. With the transition to Agile, those components that had originally been assigned to few, specific teams were now assigned to multiple teams for parallel development, the aim being to reduce time to market. However, the coordination of software development across teams became costly and difficult after the transition to Agile because of a lack of effective knowledge-integration mechanisms. Interactions aimed at coordinating interfaces among components emerged spontaneously, but in an uncoordinated manner.

Thus, the initial intention to maintain a modular product design with a high degree of interdependency between components and documented standardized product interfaces was difficult to realize because a hierarchy of technical authority and effective inter-team coordination mechanisms were lacking. Given the parallel and decentralized development of product components, the information structures and standardized component interfaces were inadequately maintained in the design of new product solutions. Formal mechanisms that connected employees across teams (e.g., inter-team meetings and demonstrations) were insufficient to convey the complexity of the emerging product knowledge, which was volatile (renewed every three weeks) and often highly tacit.

Agile development stresses the relevance of "architectural" knowledge through the development of new products based on modular designs. However, the incomplete information structure surrounding the evolving product architecture had profound implications for the development process in Company E. In addition, the pre-existing code consisted of monolithic blocks of code that were difficult to navigate and read. In response, teams attempted to develop products by experimenting with new component designs and alternative arrangements of code components.

The existence of a large-scale legacy code, made up of millions of lines of code developed in a traditional manner as a monolithic block, made it difficult to update the software programs, especially given the lack of knowledge about the software system. Notably, the new knowledge-management strategies (i.e., decentralized software production and a lack of central coordination mechanisms for diffusing tacit knowledge) were not accompanied by the widespread emergence of modular product design, which could give rise to fewer software components that were easy to assemble. Customer requirements were modularized and decomposed into smaller chunks, and solutions were reassembled at the system level through the automated integration of software deliveries. However, due to the absence of knowledge-recombination mechanisms that could collect and organize the temporary and tacit knowledge developed within teams during each iteration cycle, team members could not regularly update themselves on the development of their products. Demonstrations designed to provide an overview of the work done by a single team at the end of each iteration cycle were rarely attended by members of other teams, as they were too busy with their own deliveries. Team members therefore found it difficult to identify the relevant impacts of the new customer requirements on their legacy products and to design product updates.

Team members also faced substantial uncertainty about whom to contact for advice concerning customer requirements. In essence, team members became users who interacted with a system that was evolving through an ecosystem of developers, and these users often had interchangeable roles. Most of the knowledge creation occurred within teams through the individual interactions but with some difficulty to be spread outside their teams. The integration of newly created knowledge emerging from teams into organizational knowledge requires an ability to systematically integrate the different sources of knowledge. However, Company E lacked such a systematic ability. Multi-team meetings were supposed to function as knowledge-transfer mechanisms, but they failed to convey complex and contextual knowledge. Such knowledge transfers may require stronger ties that are characterized by closer and frequent personal interactions among team members. They may also demand the use of more formalized routines or self-contained task teams. None of these were in place.

Still, managers felt they needed to identify processes that would support an agile approach in a large-scale context. Middle managers played a key role in facilitating the agile implementation, but they often struggled to make sense of the changes themselves. Charged with implementing such changes, they experienced intense confusion, as they thought of Agile as too vague to provide guidance for the managerial tasks associated with properly reshaping the intra-organizational boundaries. Managers were also aware that there were no standard large-scale implementations of Agile that they could imitate, and that they therefore had to improvise solutions. Therefore, they relied to a large extent on the existing coordination mechanisms that the company used to manage its dependencies. These dependencies were inherent in the structure of their products (e.g., components of a system may interact, thereby constraining the kinds of changes that can be made to a single component), in the decomposition of organizational goals into activities, and in the assignment of activities to actors and resources spread across different departments or units. As a result, decision-making and goal-setting procedures remained highly centralized. The way in which regular feedback loops and reporting between teams and managers was implemented enabled managers to retain tight control.

However, managers realized that some decentralization of decision making could facilitate the execution of project tasks. Thus, they canceled the previous multi-layered networks of knowledge consisting of product experts used for consultation by employees, and of product committees in charge of approving technical documents and of giving high level directions on proposed technical solutions. Before the Agile transformation, these networks were able to enrich individual tacit knowledge with explicit knowledge and practical judgment. Through this combination of knowledge, individuals could have better support in the realization of their exploitation activities and create robust dynamic organizations that enabled environmental adaptation and creation simultaneously.

Unanticipated Consequences for Learning and Innovation

Quite contrary to initial positive expectations regarding the implementation of Agile, management of Company E increasingly had to realize that its adoption had massive, unanticipated, and far from beneficial consequences. Thus, innovation initiatives increasingly failed, the number of new patents and product improvement ideas declined, and team members found it increasingly difficult to implement customer requirements. Agile—which was expected to greatly enhance individual initiative, improvisation, and collaboration—seemed to reduce individual learning and ideation activities. The company's focus on the collection of new product improvement ideas also declined, as management resources were spent on implementing the transition to Agile, ensuring the quality of existing products, increasing the level of productivity, and reducing the time to market. Although the R&D units did not define specific key performance indicators (KPIs) on ideation performance, ideation was monitored through metrics such as the number of patents and ideas for product improvements, and these numbers showed that most of the observed R&D units exhibited declining ideation performance.

Moreover, R&D managers consistently observed that the application of Agile reduced individual employees' motivation to learn. Thus, team members seldom went beyond learning what was strictly needed to implement a requested software update. Similarly, many team members felt that the transformation to Agile had negatively affected their colleagues' interest in learning. In addition, R&D managers reported high levels of stress within the Agile teams, which further reduced the exploration of new technical solutions. According to these managers, such stress was due to the difficulty of keeping pace with the speed of the software development. Prior to the transition to Agile, the organization worked with delivery cycles lasting 18 to 24 months; but after the transition, software deliveries were scheduled every three weeks, while software integration and related integration testing activity took place every day. In addition to the perceived pressure, team members pointed to several other factors that seemed to inhibit their learning as well as their understanding of their current products and creativity (especially the absence of detailed product documentation). There was also a lack of clear and shared architectural knowledge among teams. Although senior management began to recognize the challenges, they were unable to point to specific contextual factors at the individual, team, or organizational levels that might have contributed to the inertia in individual learning or to the decline in organizational ideation capability.

Agile: Organizational Challenges and Pitfalls

Analysis of our interview data points to three categories of unanticipated organizational dynamics stemming from the agile transformation in Company E that had negative effects on individual learning and ideation behavior. First, the data revealed high coordination costs due to relevant information being increasingly dispersed across different teams and the lack of effective integration mechanisms, which inhibited knowledge flows between teams. Therefore, due to the lack of knowledge integration mechanisms at the organizational level, it was difficult to keep technological and customer competences updated and to let individuals acquire new skills. Second, the data pointed to inherent tensions caused by the difficulties of combining learning with more exploitative activities (i.e., continuous deliveries). Finally, the data revealed the emergence of separate collective identities and goals: managers stressed the relevance of learning and generating new product ideas to let their existing products evolve and improve, while teams were entirely focused on project deliveries.

The interview data helped us to further break down these unanticipated dynamics into six key specific challenges, or pitfalls, that can lead to negative consequences for individual learning, creative behavior, and individual exploitation capability.[11]

Six Pitfalls

#1: Learning and Ideation Behaviors Are Not Legitimized by the Team

The performance systems used to measure and reward team members along with the scheduling of their activities generated high levels of interdependence among team members. In addition, during daily scrum meetings, team members were asked to observe each other's progress and highlight any obstacle or constraint within the team in order to minimize the risk of not accomplishing team tasks.

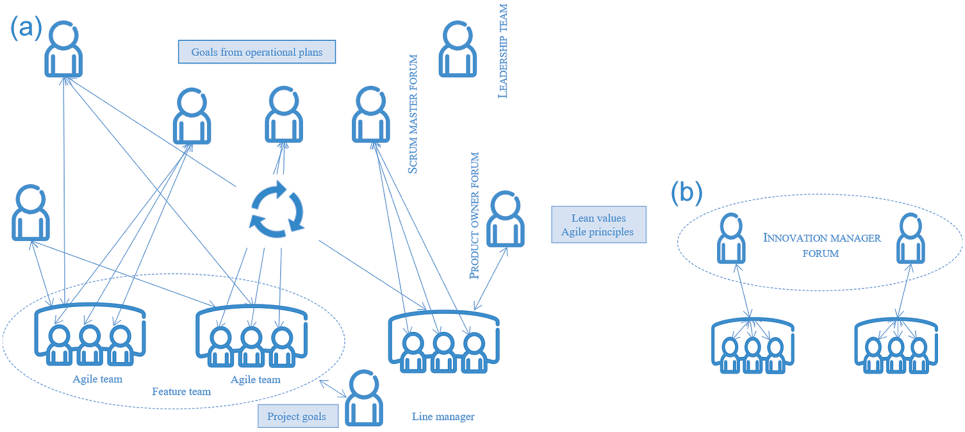
When team members are highly interdependent, they typically develop a shared perspective that allows them to coordinate actions in response to internal and external contingencies. High levels of interdependence also tend to create team norms (i.e., standards of approved behaviors inside a team) and team identities (i.e., rules of action for decision making that help resolve ambiguities). These become ways of controlling individual behaviors and cognition. However, our data point to a divergence between the goals set by managers in the observed R&D units and the professional norms developed by team members. Management continuously stressed the importance of pursuing learning and ideation behaviors to secure the constant evolution of the existing products, but team identities and norms did not include these behaviors as important.

The delegation of project tasks to teams with deliveries every third or fourth week, along with the presence of numerous feedback and reporting mechanisms, served to center team's attention on short-term project-development activities, shaping the team's identity so that it was completely focused on the need to deliver new product releases. As a result, team members did not devote time to generate new product improvement ideas or reserve time to explore the software functionalities embedded in the software systems. Thus they did not gain a better understanding of the required changes before implementation, they avoided engaging with communities of practice[12] activities, and they dropped the monthly learning days organized by their units.

#2: Organizational Structure Drifting toward the Prioritization of Urgent Issues above Learning and Innovation

Team members were under constant pressure to deliver and the presence of short feedback loops within each work iteration increased the time pressure and the debilitating effects of stress that accompanied it. These loops were mainly defined at the boundary between the team and the organizational environment and involved frequent interactions between team members and members occupying (agile) organizational roles (i.e., product owners and middle managers). The loops served to inform teams about decisions of relevance for projects and to inform those in agile

FIGURE 1. Agile feedback loops: (a) #2 the prevalent effect of agile feedback loops for project-development activities; (b) agile feedback loops for the news product ideas.



organizational roles about teams' progress. However, the latter often exerted pressure on the teams and reinforced certain counterproductive behaviors.

Figure 1a and 1b shows two competing feedback processes implemented by the R&D units: one serving to introduce new project-development activities; the other aiming at generating new product and technology improvement ideas. The arrows in Figure 1a indicate the direct interfaces between the various agile roles and the teams, while Figure 1b shows the interactions of the innovation manager with the entire organization. As shown in Figure 1a, new product-development activities resulted in several interactions with different agile organizational roles at the team and individual levels. However, the organizational structure for innovation projects (Figure 1b) reduced individual social interactions with innovation stakeholders, reducing the chance that a team member would encounter an innovation champion.

This organizational setup undermined the organizational ideation capacity, as it reduced the opportunities for innovation champions to present the generation of new product ideas as an important value. The absence of interactions between the two feedback networks created conditions for ideation initiatives to use the same organizational resources as development projects. However, the innovation network—which was based on a closed-loop structure that covered the innovation managers and individuals in teams—was unable to create opportunities for team members to spend time on generating new product improvement ideas. Shorter feedback loops and well-defined performance goals for new development projects consumed the resources and captured the team members' attention and kept it on the accomplishment of new development goals instead of generation of new product improvement ideas.

#3: Reduced Knowledge Accumulation

Team members reported that the cross-functional and cross-product nature of Agile team activities dragged them into unrelated activities, such as working on different software systems or covering different roles. As a result, they forgot their knowledge related to specific software products. Attempts to learn and absorb the information needed to implement new software functionalities represented such a volume of new knowledge to that they taxed team members' absorptive capacities. These difficulties were magnified when new information about variations in requirements arrived before the foundation of the software system was well understood. As a result, team members were unable to transform their experiences into meaningful learning, which they found frustrating. They consequently became slower and less productive than they had been before Agile was implemented.

Overall, teams were pushed by organizational stakeholders to stretch their competences and apply them across several organizational domains, which often resulted in team members working more in a cross-functional and cross-product manner within their team. The reduced differentiation of roles within teams (due to cross-functionality) and teams' commitment toward the specific product releases undermined both the value and existence of team members' product expertise within the organization. Consequently, given their reduced background knowledge, team members felt unable to understand the nuances of their jobs or to handle the complexity of their activities.

#4: Reduced Knowledge Integration

The Agile Scrum framework also failed to add new knowledge to the organization's knowledge stock. Multi-team meetings were introduced more as a way to coordinate activities among teams than to transfer knowledge across them. For instance, few people could attend demonstrations owing to their high workloads and it was impossible to attend every demonstration given the high number of teams.

An inherent challenge is that knowledge that is produced in complex product development within Agile teams may be highly context dependent and may remain tacit—particularly when, as in the case of Company E, there is minimal focus on product documentation. Given the absence of organizational routines to incorporate new knowledge into daily operations and to facilitate access to the constantly evolving organizational knowledge, team members found it difficult to acquire new knowledge and even reported declining stocks of job-relevant knowledge after the transition to Agile.

Unless the information system is effectively integrated into a firm's operational strategy, decentralized operations will not lead to goal achievement.[13] Managers provided team members with direction and focus by regularly communicating a topic of organizational importance. Team members were invited to offer their perspectives on the relevant topic, which they were then expected to back up with documentation. However, as senior managers were too far removed from the company's operations and the teams, they often failed to define the right topics. In addition, initiatives such as the creation of communities of practice, the appointment of product experts, and "learning days" remained isolated instances of information diffusion rather than effective instruments for knowledge development and sharing. Therefore, evolving/emerging knowledge was not integrated into day-to-day team operations because of the lack of effective integration mechanisms.

#5: Loss of Know-Who Knowledge

As a result of the decentralized development of knowledge within teams, individuals found it difficult to easily identify who might have the knowledge they needed. This led to an erosion of "know-who" knowledge. Significant investments were made in promoting teamwork and collaboration among team members. However, no one was appointed to play an active role in initiating the change and improving cooperation among teams. Thus, no role was established to break up pre-existing norms, restructure individual behaviors, deal with teams' isolation, systematically help team members develop innovative ideas, or determine contingency rewards that would motivate team members to more actively clarify problems and propose novel solutions. Therefore, Company E was increasingly suffering from an inability to utilize the vast amount of knowledge possessed by the individual teams.

Moreover, given the absence of formal support networks, team members increasingly turned to friendship and trust-based networks rather than to advice networks that could better direct them to relevant sources of knowledge, and which research suggests that are extremely important for innovation.[14]

#6: Lack of Individual Self-Efficacy

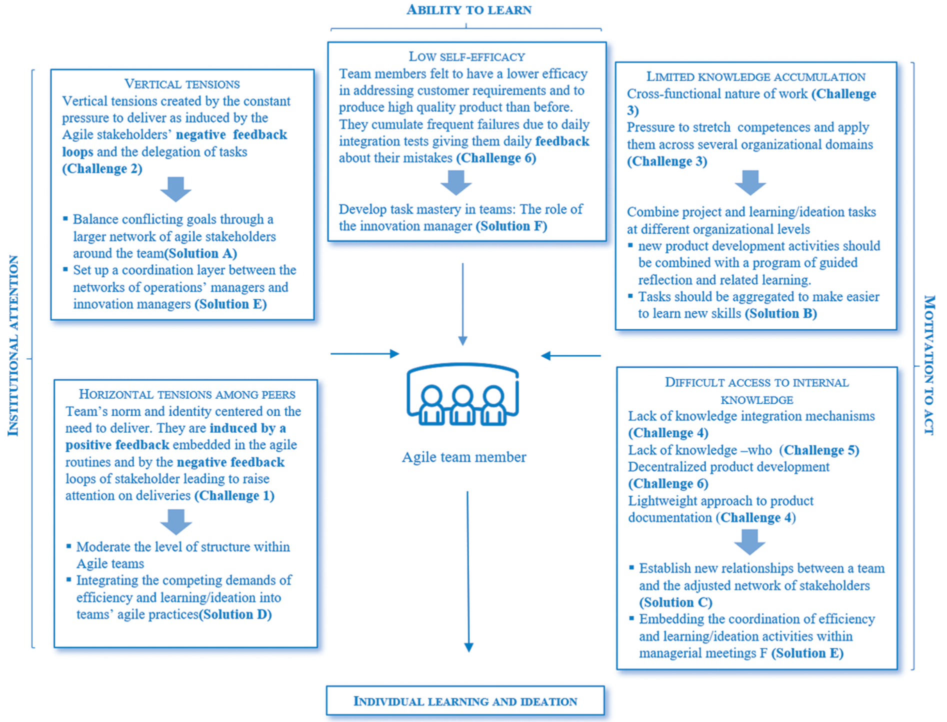
Individual team members perceived a lower level of personal efficacy related to generating new product ideas, developing high-quality software products, and executing changes in the program code. Moreover, team members experienced more frequent failures in developing new product functionalities due to the daily results of the automatic integration tests, which confirmed that, given their limited experience and knowledge of the products, they were unable to master their products as well as in the past. In addition, they felt they lost the ability to modify and control the environment given the more volatile knowledge that the organization developed along with the shorter product-release cycles. In line with the central role played by efficacy expectations in reactions to tasks and in motivations for task engagement, individual team members showed cognitive inertia in terms of learning and ideation—they waited for others to solve their problems.

Meeting Agile Management Challenges

Diagnostic Tools for Addressing Agile Challenges and Pitfalls

We now offer a systematic approach to reducing the negative effects of Agile implementation on individual learning and ideation capabilities, effects that threaten upgrading existing products and innovating new ones. To understand the implications of Agile, we collected data not only from the R&D units

FIGURE 2. Theoretical framework showing the mechanisms of influence on individual behaviors and solutions.



ABILITY TO LEARN

LOW SELF-EFFICACY
Team members felt to have a lower efficacy in addressing customer requirements and to produce high quality product than before. They cumulate frequent failures due to daily integration tests giving them daily **feedback** about their mistakes **(Challenge 6)**

Develop task mastery in teams: The role of the innovation manager **(Solution F)**

VERTICAL TENSIONS
Vertical tensions created by the constant pressure to deliver as induced by the Agile stakeholders' **negative feedback loops** and the delegation of tasks **(Challenge 2)**

- Balance conflicting goals through a larger network of agile stakeholders around the team**(Solution A)**
- Set up a coordination layer between the networks of operations' managers and innovation managers **(Solution E)**

LIMITED KNOWLEDGE ACCUMULATION
Cross-functional nature of work **(Challenge 3)**
Pressure to stretch competences and apply them across several organizational domains **(Challenge 3)**

Combine project and learning/ideation tasks at different organizational levels
- new product development activities should be combined with a program of guided reflection and related learning.
- Tasks should be aggregated to make easier to learn new skills **(Solution B)**

INSTITUTIONAL ATTENTION

MOTIVATION TO ACT

Agile team member

HORIZONTAL TENSIONS AMONG PEERS
Team's norm and identity centered on the need to deliver. They are **induced by a positive feedback** embedded in the agile routines and by the **negative feedback** loops of stakeholder leading to raise attention on deliveries **(Challenge 1)**

- Moderate the level of structure within Agile teams
- Integrating the competing demands of efficiency and learning/ideation into teams' agile practices**(Solution D)**

DIFFICULT ACCESS TO INTERNAL KNOWLEDGE
Lack of knowledge integration mechanisms **(Challenge 4)**
Lack of knowledge –who **(Challenge 5)**
Decentralized product development **(Challenge 6)**
Lightweight approach to product documentation **(Challenge 4)**

- Establish new relationships between a team and the adjusted network of stakeholders **(Solution C)**
- Embedding the coordination of efficiency and learning/ideation activities within managerial meetings F **(Solution E)**

INDIVIDUAL LEARNING AND IDEATION

that experienced problems with learning and ideation after the implementation of Agile, but also from one unit that did not fully implement Agile and reported fewer problems. This helped us develop two analytical tools to help managers create conditions for individual learning, ideation, and exploitation in a large-scale Agile setting. We subsequently tested these tools in Company E and in a multinational insurance company.

The tools are based on our understanding of how the implementation of Agile can influence individual behavior in relation to learning and ideation. To understand how individual behaviors are related to organizational influence, we analyzed individuals' connections with the organizational environment, the feedback they received, and the routines they followed. We also explored individuals' knowledge stock after the transition to Agile by asking team members about their daily operations and the cognitive means they used to accomplish them. Moreover, we asked team members about their emotional reactions to the Agile transformation.

Based on these data, we developed a framework (see Figure 2) of the main influencing mechanisms to help companies evaluate workplace sentiments after implementing Agile. It can also be used to examine the relationships between an organizational design based on Agile and individual behaviors related to individual proactiveness to learn and engage in ideation.

Many of the relevant factors in the framework have already been described in connection with the above potential pitfalls of Agile. We now organize them to make their direct influence on individual behavior toward learning and ideation activities more visible. Our findings strongly suggest that individual behaviors toward learning and ideation activities are influenced by the interaction of two types of feedback: positive (or self-reinforcing) feedback (which happens because of Agile Scrum routines) and negative feedback (which occurs from constant interactions with Agile stakeholders). We also observe that declining learning and ideation performance within self-managing teams arise from the decreasing ability of team members to absorb new competences, from difficulties in accessing relevant knowledge in the company due to the absence of knowledge-transfer mechanisms, and from low perceived self-efficacy due to the negative impact of past failures induced by the difficulty of implementing new customer requirements.

Our tools enable the identification of the feedback processes (see pitfalls 1 and 2) and other dimensions of the agile work environment that inhibit knowledge accumulation and access (see pitfalls 3, 4, and 5). They also highlight the dynamics leading to inertia in individual learning and ideation (see pitfall 6).

Diagnosis: Identify Conditions that Inhibit Learning and Ideation Activities

The questionnaire in Table 1 highlights key factors that influence the individual proactiveness to ideate and learn, which in turn affects the capacity to exploit existing products.

Analyzing answers to the questions in Table 1 can help management assess how inertial individual learning and creativity behaviors are. Specifically, managers respond to each question using a scale ranging from 1 ("strongly disagree") to 5 ("strongly agree"). The higher the score, the better the conditions for individuals within teams. Mean scores are computed for each of the five categories. These means serve as inputs for a more sophisticated diagnostic tool, depicted in Figure 3, which allows for an assessment of the nature and severity of the learning and innovation problems in an agile organization.

Addressing the Challenges

The results of the diagnostic questionnaire and the diagnostic tool in Figure 3 can be discussed with those members of senior management who can trigger changes that will improve learning and innovation in an agile context. Here are four key questions that senior management should ask to contextualize the data that emerge during the process of adopting an agile approach.

Is the level of satisfaction among team members sufficient? The first step is to analyze the scores related to team members' individual efficacy (questions 13 and 14). If the level of satisfaction with their recent work performance and work experience is problematic, then the assessment continues to step 2. Otherwise,

TABLE 1. Diagnostic Questionnaire.

	Questions
Access to internal knowledge	1. Do members of your agile teams have the possibility to work systematically with people outside their teams or the organization? 2. Do members of your agile teams have the organizational support needed to exploit the product and inject fresh knowledge from outside the team?
Degree of knowledge accumulation	3. Do members of your agile teams work on a limited set of software functionalities that are easy to understand? 4. Do members of your agile teams work across products that are not very complex (e.g., count the lines of code, look for the presence of legacy system and the availability of architecture documentation)? [This also affects teams' internal tensions]
Internal tensions within teams	5. Do members of your agile teams use a diversified set of performance goals by including, for instance, project and innovation goals when measuring their performance? 6. Do members of your agile teams have recurrent and formal routines for reflecting on what they have learned about the products they have just developed?
External tensions on teams	7. Are members of your agile teams interacting with different stakeholders (e.g., project development and innovation projects)? 8. Do those stakeholders (see 7) often interact via formal or informal meetings to discuss project or team performance? 9. Are members of your agile teams engaged in short feedback loops? Are they balanced between project deliveries and innovation/creativity activities? 10. Does the backlog contain items that indicate when it is necessary to innovate the product? 11. Are members of your agile teams constantly interacting with at least two specific actors (e.g., scrum master and innovation manager) who foster a balance in team activities focused on team projects and creativity (e.g., better quality in the developed product, faster delivery speed, number of new product ideas)? 12. Does the career development of the members of your agile teams depend on evaluations of their project-development and creativity performance?
Individual efficacy	13. Do members of your agile teams perform well in term of project results? 14. Are members of your agile teams familiar with the product they need to exploit?

the assessment should consider other aspects of the work situation in order to reveal other problems.

Is it difficult to access information? To determine whether the implementation of Agile is a possible cause of the negative working experience and the negative performance of team members documented in step 1, it is necessary to examine scores related to employees' access to information, and to consider whether the division of roles and tasks within the team and the level of support provided by the rest of the organization (questions 1, 2, 3, 4, 13, and 14) are possible causes of the problems documented in step 1. If this is the case, it is necessary to proceed to step 3. If the score is adequate, other reasons for the difficulties team members encounter in relation to learning and innovation activities must be considered.

FIGURE 3. The Agile diagnostic tool.

Levels of individual learning & innovation inertia	*Low*	**Limited access to internal knowledge** (if each question scores 1 or 2) **Low individual efficacy** (if each question scores 1 or 2)
	Medium	**Low individual efficacy** (if each question scores 1 or 2) **Low degree of knowledge accumulation** (if each question scores 1 or 2) **Limited access to internal knowledge** (if each question scores 1 or 2)
	High	**Low individual efficacy** (if each question scores 1 or 2) **Low degree of knowledge accumulation** (if each question scores 1 or 2) **Limited access to internal knowledge** (if each question scores 1 or 2) **Unbalanced internal tensions within teams** (if each question scores 1 or 2) **Unbalanced external pressure on individuals** (if at least 3 out of 6 questions score 1 or 2)

What aspects of the work environment may cause to learning and ideation inertia? This question leads to an examination of three core elements in the work environment: the position of agile stakeholders with respect to teams, the specific interests of agile stakeholders in various teams' work, and the scope of the routines enacted by team members in their daily working lives (questions 5-10 and 11-12 in Table 1). This allows for identification of the strengths and weaknesses of the current organizational structure. To define a benchmark for comparing and diagnosing problems, managers should consider the ideal organizational structure in an agile setting as one capable of reducing vertical tensions (e.g., performance feedbacks from managers) and horizontal tensions (e.g., peer pressure) by combining efficiency and learning goals. Such a structure would

- embody team and individual goals (combining efficiency and learning) that will improve both the team's project performance and each member's learning competence;
- include a regular update on each team member's short-term competence growth (within the reporting system implemented in scrum);
- limit the frequency of stand-up meetings and reports on the realization of project tasks;
- ensure that the group of agile stakeholders surrounding a team (e.g., scrum master, product owner, middle manager) includes someone knowledgeable enough to drive that team's competence growth, and to take responsibility for the execution of their learning and ideation in each iteration cycle; and
- distribute the sprint effort between operative and learning activities and create space at the end of the sprint for experimental learning through the introduction of an initiator responsible for team learning—the initiator should focus on combining and integrating knowledge by working at a higher level of abstraction than the level on which the team is working, and by interpreting knowledge with the support of the team.

FIGURE 4. The Agile change tool.

Levels of individual learning & innovation inertia	*Low*	Implement intervention action (C), (E), and (F)
	Medium	Implement intervention actions (B) (C), (E), and (F)
	High	Implement intervention actions (A), (B), (C), (D), (E), and (F)

If employees working in such an ideal Agile environment exhibit low learning and ideation performance, the problem probably has little to do with the nature or design of the work environment. Instead, it is most likely related to the relationships between the team and the external environment.

Figure 4 offers suggestions for the actions that should be implemented depending on the level of inertia. Figure 2 instead illustrates the links between the identified learning and ideation problems and the specific solutions we are proposing. As such, Figure 2 translates our theory of maintaining multiple inconsistent and conflicting demands between the need to be efficient and to proactively learn and ideate into concrete action steps that will enable firms implementing Agile on a large scale to balance those conflicting demands. Proposed action steps refer to the routines and processes through which large-scale organizations can not only coordinate, seize, and integrate different and conflicting efforts, but also recombine resources and assets across differentiated demands after agile methods are implemented.

Improving Learning and Innovation in the Agile Context

Finally, we highlight six solutions that can help firms that use Agile balance efficiency in product development activities and ideation performance. Each solution entails specific actions that will improve the quality of team members' working lives as well as their individual proactive behavior in relation to learning and ideation (see Figure 2).

Solution A: Balance conflicting goals through a larger network of agile stakeholders around the team. The principle underlying the construction of a larger network of agile stakeholders is that successful culture-shaping can create conditions for individual proactive behavior in relation to the learning and ideation activities. Three steps are needed to create such a network. First, basic work tasks that achieve both the new product-development project goals and the learning/ideation goals must be identified. Second, these tasks should be grouped into a set of structures, including team goals and feedback loops, that explicitly combine efficiency with learning and ideation. Third, each member of the team/stakeholder/manager network must share a set of personal goals pointing at the overall team's performances but keep a clear responsibility on the team's achievement of either an efficiency goal or a learning/ideation goal. In this way, the organization can focus on integrating and balancing employees' efforts between efficiency and learning/ideation without making the mistake of utilizing resources only for the fine-tuned efficiency activities related to the development projects.

Solution B: Combine project and learning/ideation tasks at different organizational levels. The fragmentation of team tasks and team members' difficulty to connect them to build a bigger picture may create a perception of unrewarding work. To make the work more meaningful, new product-development activities are integrated into a program of guided reflection and related learning under the supervision of a team's innovation managers. Once team tasks are less fragmented and disconnected from the previous work iteration—while at each work iteration they are aggregated to form a large work module—it will be easier for team members to learn new skills and gain more relevant work experience. This implies a need to appoint an innovation manager who can engage agile team members in topics of relevance for the large work module, thereby boosting their motivation to learn.

Solution C: Establish new relationships between each team and the adjusted network of stakeholders. The distribution of work among different teams leads to a reduced awareness of the ultimate user of the final products. Agile stakeholders need to encourage and enable team members to collect relevant information from clients. An enlarged set of agile stakeholders means that there will be more feedback on the team's work, which might lead to an improvement in team members' skills. Moreover, successful direct engagement with clients (or customer representatives) can improve relationships with stakeholders.

Solution D: Integrating the competing demands of efficiency and learning/ideation into teams' agile practices. The competing demands of efficiency and learning/ideation can also be addressed by simplification cycling, such as reducing the number of daily stand-up meetings and the number of team reports. Daily stand-up meetings activate team norms and controls, but they restrict the ability to select new opportunities to learn and ideate and limit the repertoire of actions that individuals can adopt for different knowledge work processes. We therefore recommend adding a new role that can expose team members to a model of the desired learning behavior as well as a credible learning strategy that will enhance the team's learning process. In addition, we propose that

- Efficiency and learning/ideation requirements should be a combined into a set of team and individual goals. This will improve the team's performance and learning, which will support daily operations.
- Frequent self-evaluations of the team's progress in acquiring competences are important.

- Stand-up meetings should occur no more than twice a week, and the flow of feedback between stakeholders and managers should be reduced.
- The sprint effort should be distributed between operational, learning, and ideation activities. A period of time should be provided at the end of the sprint to enable experimental learning. The introduction of a role responsible for team learning can help team members achieve higher levels of learning.

Solution E: Embedding the coordination of efficiency and learning/ideation activities within managerial meetings. Senior managers need to pay attention to both the short- and long-term objectives of the businesses and strike a balance between improving efficiency and capturing emergent innovation opportunities. The integration of operations and ideation may be facilitated by setting up a coordination layer between the networks of operations managers and the networks of innovation managers. This might allow for the combination of efficiency activities with real-time adjustments of activities in response to innovation challenges and opportunities.

Solution F: Develop task mastery in teams: The role of the innovation manager. When teams have numerous negative experiences during task execution, their performance is negatively affected because repeated failures lower collective efficacy, especially if those failures happen early in the course of events. To handle this problem, members must experience sufficient success when using their newly gained knowledge. This is best achieved when newly acquired skills are first used in situations that are likely to produce good results. As individuals gain confidence in handling relatively easy situations, they can gradually take on more difficult problems. Thus, the innovation manager should guide team members toward skill mastery in topics of interest for the projects. Even after teams understand the new skills, they still need guidance as well as opportunities to perfect them. Initially, they need to test their newly acquired skills in simulated situations in which they do not need to be afraid of making mistakes or of appearing inadequate. This is best achieved by role playing focused on the types of situations they must manage in their work environments and through the provision of instructive feedback. Modeling and guided performance under simulated conditions are well-suited for competence creation.

Conclusion

Agile methods have been hailed as a superior way to quickly increase organizations' abilities to adapt to changing customer requirements. They are increasingly used outside the original context of software development. However, the widespread enthusiasm for Agile might have overshadowed some of the potential costs and pitfalls, especially when Agile is implemented in large-scale software-development settings. Our five-year study of agile implementation in a major multinational telecommunications company suggests that, contrary to intentions, Agile can be harmful to organizational learning and organizational exploitation capabilities if its implementation does not take into account the organization’s needs for formal and informal integration mechanisms that combine diverse sources of information and efforts, distribute project development and ideation activities across teams, and resolve tensions. Our study points to the importance of the innovation manager, who may help facilitate open discussions about conflicting demands and aspirations within teams. We also point to the relevance of organizational integration mechanisms, such as cross-functional interfaces, that can facilitate the flow of knowledge between the different activities planned for teams without interrupting their internal software-production processes. Moreover, our findings suggest that guided training sessions to reduce negative experience which can influence perceived self-efficacy. Finally, our findings suggest that in a context of dense social relationships, such as agile teams, introducing longer feedback loops between teams and their stakeholders can be more effective at creating space for learning than encouraging team members to decide how they can best divide their time among conflicting demands.

As our research focused on a single company, albeit one with many units and hundreds of teams, questions of generalizability may be raised. However, the key purpose of this research was to provide an example of how the use of agile in a large-scale setting may unexpectedly lead to an erosion of learning and problem-solving capabilities among development teams with negative consequences for the firm-level innovation capability. As this possibility has not previously been recognized in the literature, we have concentrated on analyzing this single case in order to gain initial insights into what exactly went wrong. It is reasonable to assume that other major companies might also implement Agile in a way that produces unforeseen and undesirable results. For such companies, the Company E experience recounted here might be instructive. The same is true for the organizational solutions we have devised to diagnose and address problems associated with agile implementation.

Author Biographies

Maria Carmela Annosi is an Assistant Professor of Organization Behavior and Innovation Management at the Wageningen University & Research. (email: maria.annosi@wur.nl)

Nicolai Foss is a Professor of Strategy at the Department of Strategy and Innovation at the Copenhagen Business School. (email: njf.si@cbs.dk)

Antonella Martini is a Professor of Strategy at the School of Engineering, University of Pisa. (email: antonella.martini@unipi.it)

Notes

1. http://agilemanifesto.org/
2. B. W. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed* (Boston, MA: Addison-Wesley, 2004).
3. M. Paasivaara, C. Lassenius, V. Heikkila, K. Dikert, and C. Engblom, “Integrating Global Sites into the Lean and Agile Transformation at Ericsson,” *IEEE 8th International Conference Global Software Engineering*, IEEE Xplore, 2013, doi:10.1109/ICGSE.2013.25, pp. 134-143; M.

Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study," *IEEE 9th International Conference on Global Software Engineering*, IEEE Xplore, 2014, doi:10.1109/ICGSE.2014.22, pp. 16-25; T. Dingsøyr and N. B. Moe, "Towards Principles of Large-Scale Agile Development," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014*, ed. T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen (Cham, Switzerland: Springer, 2014), pp. 1-8.

4. See VersionOne, Inc., 2018, *12th Annual State of Agile Report*, https://stateofagile.com/#ufh-i-613553652-12th-annual-state-of-agile-report/7027494.
5. T. Dybå and T. Dingsøyr, "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology*, 50/9-10 (August 2008): 833-859.
6. K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and Success Factors for Large-scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software*, 119 (September 2016): 87-108.
7. M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-Scale Agile Transformation at Ericsson: A Case Study," *Empirical Software Engineering*, 23/5 (October 2018): 2550-2596.
8. This can be seen as an example of "exploitation" driving out "exploration." See J. G. March, "Exploration and Exploitation in Organizational Learning," *Organization Science*, 2/1 (March 1991): 71-87.
9. Parts of the research are reported in M. C. Annosi, N. J. Foss, F. Brunetta, and M. Magnusson, "The Interaction of Control Systems and Stakeholder Networks in Shaping the Identities of Self-Managed Teams," *Organization Studies*, 38/5 (May 2017): 619-645; M. C. Annosi, M. Magnusson, A. Martini, and F. P. Appio, "Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development," *Creativity and Innovation Management*, 25/4 (December 2016): 515-535; M. C. Annosi, A. Martini, F. Brunetta, and L. Marchegiani, "Learning in an Agile Setting: A Multilevel Study on the Evolution of Organizational Routines," *Journal of Business Research*, 110 (March 2020): 554-566.
10. The Agile transformation shaped the definition of organizational roles within Company E. The middle manager (i.e., "line manager") had the following responsibilities after the Agile transformation: (1) act as a change agent for the teams she or he supervised; (2) secure the right team composition in terms of focus and competences; (3) secure a proper work environment for the team (e.g., seating, IT); (4) establish external interfaces between his or her teams and people working with other telecommunication nodes/products; (5) secure availability of external team members (from other locations or external suppliers); (6) provide important information, like team membership, start date, next features in pipeline, and strategy; and (7) organize sprint 0 together with the scrum master and product owner.
11. Specifically, the challenges/pitfall are related to the four unanticipated organizational dynamics in the following manner: (1) the coordination costs dynamics is broken down into challenges 4, 6, and 7; (2) the exploration/exploitation dynamics relates to challenge 1; (3) the isolated teams dynamics is broken down into challenges 3, 6, and 8; and (4) the separate identities dynamics relate to challenges 2 and 5.
12. Communities of practice are groups of team members belonging to different teams informally bound together by shared expertise and a passion for a specific software-development practice.
13. See, in particular, R. Simons, *Control in an Age of Empowerment* (Boston, MA: Harvard Business Review Press, 2008).
14. This is documented in I. Nonaka, G. von Krogh, and S. Voelpel, "Organizational Knowledge Creation Theory: Evolutionary Paths and Future Advances," *Organization Studies*, 27/8 (August 2006): 1179-1208.