Interface Problems of Agile in a Non-agile Environment

Sven Theobald and Philipp Diebold^(✉)

Fraunhofer Institute for Experimental Software Engineering (IESE),
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{sven.theobald, philipp.diebold}@iese.fraunhofer.de

Abstract. Agile is the widespread software development approach. But many projects are still working with traditional methods. In addition, non-technical business units continue working in traditional ways. Thus, problems arise on the interface of agile and traditional due to their fundamental differences. To prevent potential problems, one must be aware of the existing interfaces and common pitfalls. Based on a literature search and workshops, we identified existing interfaces, collected and grouped problems. We present the identified problems and propose a matrix that facilitates classification of interface problems. This matrix can be used to identify and classify more problems as well as understanding and preventing problems on the interface of agile and traditional.

Keywords: Agile · Traditional · Plan-based · Non-agile environment
Hybrid organization · Interface · Problems · Challenges · Classification
Agile transition

1 Introduction

Agile is widespread, especially on team level [1]. But other studies show that complete agile enterprises are quite rare. This is mainly the case because at least the supporting functions, such as marketing or human resources, are still operating in their established ways. Thus, the agile parts of an organization do have at least those interfaces to these traditional environments. Alignment between agile and traditional approaches is challenging [2]. There is a lack of guidance for problems at those interfaces, although many practitioners confirmed that these problems exist. Understanding and addressing them is viable to make use of the benefits provided by agile, and to support agile initiatives to progress towards an agile organization. This work discusses which interfaces exist, presents the problem fields and proposes a problem classification.

2 Related Work

An interface problem is a (potential) problem or challenge occurring on the interface of agile and traditional approaches, e.g., caused by the conflicting underlying principles, culture, or processes. There exists literature reporting problems with agile, e.g., [1, 3], or [4]. Interface problems are already mentioned, but the borders to other organizational

internal or external entities surrounding the agile team are not explicitly considered. The following literature focused on interface problems.

In interviews with 21 agile practitioners, two problem categories resulted [5]: Increased IT landscape complexity, caused by concurrent development streams, separated layer development and different processes. The other one was lack of business involvement, caused by a centralized it department and a traditional project organization. Mitigation strategies were proposed to address the identified challenges.

Based on interviews with seven German practitioners, problems in agile projects within traditionally organized corporations were identified [6]. The study is limited to the viewpoint of agile teams, especially on global software engineering. Among the 8 identified problems, 3 had an non-agile environment: “wrong application of agile”, “lack of acceptance of agile principles” and “difficulties in translating agile [...] into their non-agile counterparts.” Further interface problems are mentioned, such as “fragmented documentation in agile” or “combined usage of paper and electronic media”.

With the aim of analyzing how agile and lean approaches are adopted at scale, a literature review was conducted focusing on challenges and success factors [7] considering interfaces to other organizational units. We identified 17 of their challenges as interface problems. The groups with the highest number of interface problems were “integrating non-development functions”, “hierarchical management and organizational boundaries” and “coordination challenges in multi-team environment”.

Kusters et al. [8] identified challenges and risks that appear in hybrid organizations. The focus was on project and program level issues that impact the coordination and cooperation. 22 issues were classified in six groups: Organization and structure, business processes and control, culture and management style, development and testing, involvement of stakeholders, documentation and communication.

Motivation has to be considered when interfacing with non-agile surroundings [9]. Agile team members might get frustrated by missing or unsynchronized feedback from traditional stakeholders, or when having to work in a traditional way after experienced agile projects. More interface problems were mentioned, such as a fixed budget and project plan at the project start or having to adapt to the surrounding stage-gate process.

Kuusinen et al. [10] already went one step further and identified strategies to mitigate such problems. The information was collected from a survey and a workshop. The themes were grouped in two categories, one for organizational themes and one for change themes. Most strategies are generic and can be applied at different interfaces.

3 Research Method

Our overall research goal is described in the GQM-goal-template [11] as following: Identify and analyze *common existing problems on the interface of agile in a non-agile environment* in order to *create a classification* in the context of *organizations developing software or systems* from the perspective of *research and practice*. Because this goal covers several different aspects, we needed to refine it into several research questions (RQs). For this reason, these are the RQs refining our study goal: **RQ1** - Which interfaces

of agile entities to a non-agile environment exist? **RQ2** - Which problems do appear on the identified interfaces? **RQ3** - How to classify and group these problems?

The research procedure contained the following steps: (1) We started with a brief research on the state of the art in existent literature. (2) Second, we performed a workshop with five IESE employees, all experienced in agile and some in systems engineering. (3) Third, we had the possibility to discuss this topic at the Automotive Agile PEP-Conference. We were able to collect further problems from practitioners and discuss most important problems to identify concrete solutions. (4) Afterwards, an initial classification was developed. (5) Finally, this classification was used to categorize the identified problems on specific interfaces according to the dimensions of the classification.

4 Results

The results are presented along the RQs. Thus, we first present the different interfaces, where problems might arise. Then we present an overview over the problems, identified and categorized into different problem fields. Finally, a matrix that supports classification of interface problems and solutions is presented and discussed.

4.1 Existing Interfaces (RQ1)

The first step was to identify all possible situations where agile approaches could interface with traditional development. The identified interfaces can be categorized into company-internal (I) and external (E) interfaces, see Table 1:

Table 1. Presenting the interfaces (name, external/internal interface, description/example)

Name	E/I	Description/Example
Project-Team	I	The coordination between agile and non-agile teams in large projects appears to be a challenge, e.g. the synchronization between agile software teams and hardware teams
Organization - Project	I	In larger organizations, projects collaborate with different organizational units, e.g. the line management demanding traditional reporting or synchronization with other organizational parts like human resources or operations. Agile projects have a different mind-set and information demands than traditional organizational units
To Customer	E	Successful collaboration with customers and other stakeholders is important, e.g. for requirements clarification or meetings. Working with traditional customers is challenging for an agile project
To Subcontractor	E	Traditional subcontractors slow down the agile process and complicate synchronization. Also, agile subcontractors of traditional teams might cause issue
To Reg. authority	E	The requirements posed by regulatory authorities have to be considered. Regulated organizations have to provide evidences for the fulfillment of regulations, leading to a focus on documentation

4.2 Existing Problem Fields (RQ2)

In total, 169 problems were assigned to various interfaces during our data collection. Different categories emerged when grouping similar problems into the so called problem fields. Example problems will be given either in the description of the problem fields or in the problem classification (cf. Sect. 4.3).

Project Planning. Traditional plans are specified upfront, e.g. in detailed milestone- or Gantt-charts and only updated when necessary. Thus, it is difficult to synchronize with agile plans which are only detailed for a short period of time and which are adapted each iteration. Traditional project managers perceive Agile as lacking the right amount of risk management, traceability, requirements management and quality assurance. Breaking down dependencies between agile and traditional teams is challenging.

Controlling, Reporting & Approval. Similarly, traditional project managers demand a different kind of reporting and expect different artifacts than delivered by agile. This leads to effort for a second report or the transformation of agile reports. The regular reprioritization and de-scoping done by agile teams is perceived as lack of control. There might be differences between the formal approval of delivered artifacts, especially concerning safety aspects in system development. KPIs might differ, e.g., individual performance is measured while often team productivity is measured in Agile.

Contracting & Budgeting. Contracting between a traditional customer and agile projects is a common challenge. The customer usually expects to have certainty of time, scope and budget at the beginning of the project. The approval of the product depends on the contractually defined features from the start of the project, even if the scope was changed to optimize business value. Traditional customers are often not able to provide the team with a representative who collaborates with the team on a daily basis and is knowledgeable enough to take over product responsibility.

Process Requirements. Especially regulated domains have to stick to standards and regulations prescribing how their process has to look like. Original Equipment Manufacturers demand certain development approaches by their suppliers. Regulations demand upfront planning, heavyweight documentation of requirements or proofs of risk management. Standards like Automotive SPICE or ISO 26262 are relevant in the automotive context, and it is still not clear how agile processes cover these standards.

Tooling & Infrastructure. Traditional and agile teams have different needs for tools and infrastructure. Some projects are forced to use the organizational tool chains, although these might not fit to agile. The agile way of using paper-based media like a whiteboard, physical task boards or burn charts is a problem in distributed projects.

Coordination & Integration. Dependencies and integration of products developed by several teams need to be synchronized. It is a challenge to coordinate agile iterations with the stage-gate process of traditional (system) development. The communication needs of agile teams are often not met by other non-agile business units (e.g., marketing, sales, operations, customer service) who are not used to direct involvement. Therefore, long reaction time slows down the agile teams. For distributed teams, especially over time zones, agile collaboration creates more challenges due to the

regular collaboration and feedback loops. From the technical point of view, dependencies in the architecture of several teams might become a problem, since agile architectures emerge during the project, while traditional architectures are extensively planned upfront and expected not to change. Also, dependencies between fast software and slower hardware development have to be tackled, as well as collaboration with dedicated quality assurance teams.

Staffing. The typical traditional organizational structure is functional, hence the knowledge is distributed in silos. An agile team contains all roles necessary to conduct the project, so there are no borders between different functional teams handing over products or artifacts to another silo. Traditional teams are overfilled with team members, each one working only partially on the project. Agile suggests that team members are dedicated to only one product, and teams are only as big as they have to be. There is a change in role profiles, since agile demands different skills from team members as well as from managers. Developers in agile teams are expected to be T-shaped, having expertise in a specific field while being able to collaborate with other disciplines.

4.3 Problem Classification (RQ3)

Using the concepts of interfaces and problem fields allows classifying existing problems along two dimensions: Which interface does the problem occur at? Which problem field does the problem belong to?

Some of the identified problems might occur on several interfaces and it might be possible that one problem fits to several problem fields, because some problem fields have borders to each other. Based on the resulting classification matrix, each problem (and solution) can be related to one (or more) cells of the following classification matrix (cf. Table 2). As an example for the classification, synchronization between an agile software team and a traditional system development project happens in the problem field Coordination & Integration on the interface Project – Team.

A comparison of the number of problems occurring on the different interfaces helps to assess which interfaces are more relevant, since they contain the most problems. Especially the interface Organization - Project contains a wide range of problems ($n = 64$), since multiple organizational units have to cooperate with a software team, e.g., human resources, sales, operations, marketing, etc. Thus, there are many instances of the organizational unit, therefore more possibilities to find problems. The high number of problems on the Project - Team interface ($n = 50$) is understandable, since this is where most organizations currently are [1]. Some pilot teams are already agile, but have to work with traditional teams to deliver a common product. Although the interface To Customer is very specific, surprisingly many problems were found ($n = 28$). This shows that many customers still lack the understanding of agile and have to learn how to collaborate with agile teams. There are also quite many problems on the interface To Subcontractor ($n = 27$). None were identified yet on the interface To Regulatory Authorities.

Regarding the problem fields, a comparison of the amount of problems in each problem field might show which field is causing the most problems. Coordination & Integration is the biggest problem field ($n = 39$), potentially because it is very broad

and many things can be classified as the very general term collaboration. Contract & Budgeting (n = 31) and Project Planning (n = 27) were also two problem fields with a high amount of problems, followed by Staffing (=23) and Controlling, Reporting & Approval (n = 20). Process Requirements (n = 15) and Tooling & Infrastructure (n = 14) were the problem fields with the smallest number of problems.

Some combinations of problem fields and interfaces are more common. That means that certain problem fields are more likely to cause problems on a certain interface. Coordination & Integration problems happen most often on the interface Project – Team. Staffing issues are most likely to occur on the interface Organization & Project, often with the human resource department representing the organization. Contracting problems occur mainly on the interface To Customer or To Subcontractor. Controlling is challenging on the interface Organization – Project. Further, Project Planning is relevant for both Organization - Project and Project – Team interfaces. Tooling problems are most likely to occur on the interface between a Team and the overall Project.

Table 2. Matrix presenting the two dimensions and the total number of identified problems (from literature, from IESE, from conference workshop).

Problem fields	Interfaces				
	Organization-Project	Project - Team	To Customer/ Stakeholders	To Reg. Authority	To Subcontractor
Project Planning	11 (8, 3, 0)	10 (6, 2, 2)	4 (1, 2, 1)		2 (0, 2, 0)
Controlling, Reporting & Approval	10 (8, 1, 1)	5 (3, 2, 0)	4 (2, 2, 0)		1 (0, 1, 0)
Contract & Budgeting	7 (0, 2, 5)	1 (1, 0, 0)	12 (3, 3, 6)		11 (1, 1, 9)
Process Requirements	8 (3, 1, 4)	4 (3, 1, 0)	2 (0, 2, 0)		1 (0, 1, 0)
Tooling & Infrastructure	3 (2, 1, 0)	6 (2, 2, 2)	1 (0, 1, 0)		4 (0, 2, 2)
Coordination & Integration	9 (8, 1, 0)	21 (11, 4, 6)	4 (2, 2, 0)		5 (1, 1, 3)
Staffing	16 (11, 3, 2)	3 (0, 3, 0)	1 (0, 1, 0)		3 (0, 3, 0)

4.4 Limitations

Except for the data from literature, data is mostly representing the automotive domain. IESE experiences are based on agile projects often conducted with automotive companies or suppliers and the participants of the workshop were mainly from this domain. This poses a threat to the generalization of results to other domains. The literature review was not conducted as a systematic literature review, and did only consider a few of the first sources that were identified. This led to a limited amount of identified

problems, which limits the validity of the classification matrix and the possibility to analyze which problem fields or interfaces cause the most problems.

5 Conclusion and Future Work

Although general problems with agile are covered by research, there is few related work investigating problems on the interface of agile and traditional approaches. We collected those interface problems with a literature review and within workshops, categorized them into problem fields, and finally suggested a classification matrix. The results support practitioners in considering and mitigating potential problems, and help researchers align their research efforts on the most common and important problems. Of course, the classification matrix has to be refined, evaluated and improved with the help of further data. This data will be identified in interviews with practitioners or from reviewing literature.

References

1. VersionOne: 11th Annual State of Agile Report. <https://www.versionone.com/>
2. Thamhain, H.J.: Can we manage agile in traditional project environments? In: 2014 Portland International Conference on Management of Engineering & Technology (PICMET), pp. 2497–2505. IEEE, July 2014
3. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. *Commun. ACM* **48**(5), 72–78 (2005)
4. Gregory, P., Barroca, L., Sharp, H., Deshpande, A., Taylor, K.: The challenges that challenge: engaging with agile practitioners' concerns. *IST* **77**, 92–104 (2016)
5. Van Waardenburg, G., Van Vliet, H.: When agile meets the enterprise. *IST* **55**(12), 2154–2171 (2013)
6. Richter, I., Raith, F., Weber, M.: Problems in agile global software engineering projects especially within traditionally organised corporations. In: Proceedings of International C* Conference on Computer Science & Software Engineering, pp. 33–43. ACM, July 2016
7. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *JSS* **119**, 87–108 (2016)
8. Kusters, R., van de Leur, Y., Rutten, W., Trienekens, J.: When agile meets waterfall - investigating risks and problems on the interface between agile and traditional software development in a hybrid development organization, pp. 271–278 (2017)
9. Gren, L., Torkar, R., Feldt, R.: Work motivational challenges regarding the interface between agile teams and a non-agile surrounding organization: a case study. In: Agile Conference, 2014, pp. 11–15. IEEE, July 2014
10. Kuusinen, K., Gregory, P., Sharp, H., Barroca, L.: Strategies for doing agile in a non-agile environment. In: Proceedings of ESEM, p. 5. ACM, September 2016
11. Basili, V., Caldiera, G., Rombach, D.: The goal question metric approach. In: Encyclopedia of Software Engineering, vol. 2, pp. 528–532 (1994)

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.