Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study

Francisco Gomes de Oliveira Neto, Jennifer Horkoff, Eric Knauss, Rashidah Kasauli, Grischa Liebel
Chalmers and the University of Gothenburg
Gothenburg, Sweden
{gomesf,jenho,knauss,rashida,grischa}@chalmers.se

***Abstract*—As agile methods become more pervasive, agile practices are applied to more large-scale systems with a scope that goes beyond pure software. The expansion of agile in these contexts provides benefits, but creates new challenges. Widespread use of agile has changed the way we must think about practices both in Requirements Engineering (RE) and in System Testing (ST). Our experience shows that many challenges in the application of large-scale agile development relate to either RE or ST, and in particular to the alignment between these areas. In this paper we present large-scale agile-related challenges from a multiple case study which relate to REST alignment. We map our challenges to an existing framework for REST alignment, and make an initial attempt to suggest agile RE practices from the literature which may alleviate these challenges. Our results show that the interviewed companies need to first adopt more agile RE practices to enhance REST alignment and then leverage agile testing. Future work will look more towards evaluating these best practices.**

***Index Terms*—requirements engineering, system testing alignment, large-scale agile system development**

I. INTRODUCTION

Because of their benefits, agile methods, such as Scrum, have been extending their boundaries, both in terms of size, with application to increasingly large-scale projects, and in terms of scope, moving beyond pure software projects to more system-oriented projects [1, 2]. Incorporating agile practices and principles into other domains of development, besides software, has shown promise to different companies. Nonetheless, by moving towards a large-scale context with an increased scope and complexity, the adoption of agile methods produces several challenges for both processes of requirements engineering (RE) [1, 2, 3] and system testing (ST) [4, 5].

There are several trade-offs in RE for agile development with respect to the requirements artifacts (e.g., system requirements, non-functional requirements, system models) and the relation of those artifacts to customer value. Considering user stories, for instance, on one hand there is simplicity, but also limitations of effectively communicating requirements to off-site stakeholders. Similarly, the agile methods' emphasis on customer value can produce understated non-functional requirements [6]. Other challenges relate to longer lead times due to dependencies with hardware availability and interfacing with system engineering practices [6]. Such a combination of different disciplines and domains is a common source of challenges, since each contributes its own expertise and ways of communicating or performing activities.

In turn, different levels of testing in the V-model (e.g., unit, integration and system) provides distinct challenges towards product development. Lower level artifacts provide feasible support to automated techniques since they are closer to the System Under Test (SUT) implementation, whereas, high level artifacts provide comprehensible information to different roles in the product development chain [7]. Thus, when adopting agile methods for testing in large-scale companies, one needs to even out the distinct ways that different experts communicate [5] in order to enable exchange of information and reach alignment between requirements and the related test artifacts (e.g., test cases, test reports, test executions) [8].

Others have explored the relationship between agile methods in testing [5] and RE [9, 4], as well as their alignment [4, 8]. Similarly, there are studies that investigate large-scale testing [10] and RE [2, 3]. Existing studies on combining large-scale agile RE list several benefits to system development such as reduced development lead time and increased planning efficiency [6].

Therefore, we believe that by exploring further the relationship between RE and test artifacts one can continue to foster agility within large-scale companies. Particularly, we aim to answer:
RQ1: What are the RE-related challenges pertaining to large-scale agile testing in systems development?
RQ2: What are main challenges hindering alignment between requirements engineering (RE) and system testing (ST)?
RQ3: What are existing requirements engineering practices to improve alignment between RE and ST?

We explore these research questions in a multiple case study with three companies based on focus groups with topic matter experts. We begin by investigating challenges stated by each company during our data collection to address RQ1. Then, we use that data to identify RE and System Testing (REST) alignment challenges for each company (RQ2) based on the list of challenges presented in Bjarnason et al. [4], followed by an indication of practices to address those challenges. Since our exploration wants to foster agility within large-scale RE and testing, we use practices for agile RE, presented by Inayat et al. [9], to address each alignment challenge and answer RQ3. This paper comprises two main contributions:

- We identify a variety of challenges related to both RE and ST, and propose solutions to help stakeholders address those challenges.
- We offer strong indication that existing results on RE and Software Testing alignment can be generalized to large-scale agile system development. The challenges [4], and practices [9] identified in related work were also applicable to large-scale agile system development in our investigated companies.

Our conclusions show that the investigated companies face similar challenges regarding large-scale agile system development and, in order to leverage agile testing, the companies need to improve requirements management, prioritization and continuous planning. In other words, by encouraging agile RE practices, companies can also leverage their large-scale agile testing because requirements information can then be used to support automated techniques for both the product and the development teams. Even though our findings are limited to the companies involved, we use our experience in industrial cases to give insights on these questions, framing future work.

Section II presents existing research comprising large-scale and agile methods pertaining to both RE and testing, followed by methodology in Section III. We discuss our results about REST alignment (Section IV) and expose the limitations in our study (Section V). Finally we draw conclusions and describe future work in Section VI.

II. Related Work

A. REST alignment frameworks

REST alignment is the process of adjusting RE and testing activities to coordinate and optimize product development [8]. Often perceived as very similar concepts, we refer to alignment and traceability as two different instruments, where traceability is related to the connection of artifacts through trace links, and alignment includes coordination of practices, artifacts, and roles [8, 4]. Therefore, traceability becomes one element of alignment to document the relation between, for instance, requirements specification and test cases. There are several benefits in investing on the alignment between RE and testing, such as improved requirements coverage, risk management Gorschek and Davis [11], and better communication between stakeholders Kukkanen et al. [12]. Often the investigation leads to a set of challenges and practices to improve REST alignment.

Bjarnason et al. [4] perform a systematic literature review (SLR) to collect the set of proposed REST alignment challenges reported in literature. Unterkalmsteiner et al. [8] advances the contributions towards REST-alignment into a taxonomy and a process linking the respective areas through dyads and network constructs that can identify opportunities to improve REST alignment. At this stage in our case studies, we could not collect enough details on each company's tests, requirements artifacts and tacit knowledge to build a dyad network and benefit from Unterkalmsteiner et al. taxonomy. Instead, we begin by identifying specific challenges and hope to apply the taxonomy in future work as more data collection is performed with practitioners.

Regarding practices to improve REST alignment, Uusitalo et al. [13] interview six companies, where they present alignment practices aiming to improve communication and interactions between different testing and RE roles, whereas Kukkanen et al. [12] proposes a set of practices to align RE and ST such as use of metrics and traceability with tool support. In our analysis, we use agile RE practices presented in an SLR by Inayat et al. [9] to address REST alignment challenges. Their list of practices rather focus on supporting agile RE instead of REST alignment, thus providing a new viewpoint to existing analysis, especially considering large organizations. Below, we briefly describe their 17 practices (in italics), but refer to their SLR for a more detailed description.

Face-to-face communication refers to communication between clients and team members, emphasizing minimization of documentation. *Customer involvement and interaction* involves identification of appropriate customer contacts or representatives to extract requirements. *User stories* are the agile form of requirements specification, while *iterative requirements* refer to an emergence of the requirements over time via frequent interactions with stakeholders. During each agile iteration, *requirements are prioritized*, often by customers with a focus on risk and value.

Change management is a broad category in traditional RE, but in the agile context refers to adding or dropping features and refining requirements via frequent client communication. *Cross-functional teams* combine team members such as testers, designers and product owners, helping to alleviate communication gaps between roles. *Prototyping* allows for quick feedback for high-priority requirements, while *testing before coding* means writing tests before starting to implement functionality in code, related to test-driven development. *Requirements modeling* in an agile setting is more lightweight compared to traditionally RE modeling, focusing on easy-to-read sketching like models, such as goal-sketching. *Requirements management* in an agile setting refers to maintaining a product backlog, feature list, or index cards, while *review meetings* continuously review the status of the backlog, and acceptance tests provide a pass/fails results for individual user stories. *Code refactoring* has the typical software meaning, particularly important in light of changing requirements. *Shared conceptualizations* referred to the shared understanding of agile teams, fostered by frequent communication and necessary due to reduced documentation. *Pairing for requirements analysis* refers to a stakeholder performing multiple roles, similar to cross-functional teams. *Retrospective* meetings review the work and find new requirements after an iteration, while *continuous planning* refers to the general agile way of working, continually re-planning and adjusting to new requirements.

B. Large-scale agile RE and testing

We use the term *large-scale agile* to refer to large organizations adopting or implementing agile methods in their system development process [6]. Separately, the terms large-scale and agile can be conflicting; for instance, agile methods try to reduce complexity which is often an inherent aspect of a large-scale organization. Research pertaining to both terms has shown to be beneficial when addressing them independently [1, 9, 3, 2, 10, 5] and combined [6].

When considering agile methods, usually researchers relate improvements in terms of a comparing to a "traditional" approach, such as agile RE versus traditional RE. The same applies to testing. Adopting agile RE and testing methods has produced benefits to companies, such as decreasing process overhead and become more flexibility to changes [6]. However, there are both organizational and process changes involved when becoming agile while coming from a traditional waterfall development.

From an RE perspective, the roles of requirements must change to emphasize customer value and avoid overwhelming specification documents. For instance, in their SLR, Inayat et al. [9] map challenges from traditional RE that can be addressed by their list of agile RE practices. They further provide a list of 8 challenges from agile RE itself that can also be addressed by agile RE practices. In our research we see that many of those practices can, in fact, help companies overcome large-scale agile REST-alignment, especially since Inayat et al. emphasizes both test and requirements artifacts in some of those practices, even though they are primarily focusing RE.

In turn, testing also has its own set of challenges and solutions pertaining to agile methods. System testing relies on interaction with the system under test (SUT) in order to provide verification and validation assessment to stakeholders [5]. Test cases represent that interaction, which, can be done within a continuum having automatic and manual activities as opposite ends, each with their own trade-offs. Even though automated techniques is one of the goals when adopting agile testing [5], most processes do not support it since requirements and development artifacts are scattered among, or not properly managed by, stakeholders.

Both RE and testing can be found in existing processes that scale agile, such as the Scaled Agile Framework (SAFe)[1] and the Large-Scale Scrum (LeSS)[2]. SAFe is composed of 4 to 5 levels (team, program, value stream, portfolio and foundation), each with its own purpose such that, combined, they support four core values (alignment, built-in quality, transparency and program execution) to promote a lean-agile mindset. In turn, LeSS is similar to Scrum, but instead, shifts the team's focus from "my part" to the product as a whole so that all teams act as a team itself, thus providing stakeholders with guidelines on how to coordinate activities within and across teams in order to foster sharing and cooperation. The participating companies do not use SAFe and LeSS rigorously, thus we did not consider either frameworks in this study, but plan to explore them in future work with the same companies.

TABLE I
OUR CASE STUDY PLANNING ACCORDING TO GUIDELINES PRESENTED BY RUNESON AND HÖST [14].

Objective	Explore
The context	RE and testing in large-scale agile system development
The cases	Telecommunication case (FG = 2 people) Manufacturing case (FG = 7 people) Automotive case (FG = 5 people)
Theory	Challenges in REST-alignment [4] and agile RE practices [9].
Research questions	RQ1, RQ2, RQ3 (see Section I)
Methods	Semi-structured interviews
Selection strategy	Large companies adopting agile methods

III. METHODOLOGY

We perform a case study with the objective of exploring and describing the RE challenges that companies face when moving towards large-scale agile. Through workshop with different companies where we perform interviews with focus groups, we collect data regarding their artifacts and how they are used. We then use that information to describe each case and point to particular aspects of their RE or testing processes.

We performed the workshops and interviews with three companies from different domains: i) telecommunication (Tele), ii) package manufacturing (Manu) and iii) automotive (Auto). All are large companies developing products and systems that include a software, hardware, and mechanical components. Prior to the workshops, each company was asked to prepare: i) one or more projects to discuss, ii) a set of artifacts to show and discuss during the workshop, and iii) a group of practitioners, that we refer here to as a focus group (FG) involved in RE and testing activities to participate in these interviews. Table I presents our case study's planning.

Therefore, we do a multiple case study, with three participating companies, where each company is studied within their corresponding case. We decide to consider only one unit of analysis per case because the information collected from the FG is self-contained, i.e., cannot be split into distinct units of analysis because we did not interview groups from different projects within each company, separately. Then we summarize our findings by analyzing cross-cutting concerns identified in all companies from an REST-alignment perspective.

We used semi-structured interviews where, in a three-hour workshop session, we established a dialogue with each focus group. Each workshop lasted for three hours and was hosted by the companies in their respective workplaces. We used an instrument with a series of forms and questions to guide the dialogue. Using a funneling principle [14], we begin by questioning them about their requirements artifacts and then we narrow the questions down to specific aspects of test activities.

[1] http://www.scaledagileframework.com/
[2] https://less.works/

Nonetheless, during those dialogues, the subjects were allowed to diverge from the questions in order to explain specific aspects of their company's apparatus and *modus operandi*. This step was meaningful to our interviews to collect, particularly, data on the tacit knowledge from stakeholders.

After preparation and planning of the workshops, we begin our data collection phase by iterating one instance of the same instrument per company and recording the audio from the interviews. The workshops happened in different weeks, according to the practitioners' availability. After conclusion of data collection, we started to analyze data: i) workshop audio files, ii) notes taken by the participants, iii) documents sent by practitioners that were presented by them as requirements or testing artifacts. We coded the data in order to identify the artifacts and the challenges reported by the companies.

Our analysis is based on two existing frameworks in literature: we identify REST alignment challenges presented and discussed in Bjarnason et al. [4], and we refer to Inayat et al. [9] list of agile RE practices. Both are briefly discussed in Section II but detailed in Section IV for the context of our case studies.

A. Description of RE and ST in each case

Most of the artifacts collected are closer to the requirements/ system level. The goal, for each case, is to present the challenges for each company with respect to the collected artifacts or explanations from practitioners (presented as quotations below) in the corresponding focus groups and help us answer RQ1.

B. Tele

Tele is experienced with software development, and its teams have been using agile methods for years, such that it has a process for their system development that complies with some agile practices. However, long lead times due to dependencies on hardware components still hinders agility in their development and testing.

Most of the testing activities rely on an artifact named *test report analysis*, created during the planning stage of a feature by test coordinators. There are guidelines and templates to create this artifact, but they are not always followed. In addition, updates on the features are not necessarily reflected on this artifact, hence reducing its REST alignment.

> *". . . there we have somewhat a challenge, one is developing faster and having some functionality and for some requirements, we need to keep changing . . . and potentially we can also break this report [test analysis report] for test for another product, when we are changing for other product."* — Tele

The reason is that stakeholders are sometimes encouraged to take ownership of features leading to a trade-off, where on one hand he or she reduces their documentation effort (i.e., they no longer need to create a thorough test artifact), and on the other hand the team becomes dependent of his/her involvement.

That aspect is less encouraged in their cross-function teams, where roles are not explicitly assigned and, ideally, anyone can assume the role of, for instance, a tester and create test cases. However, in practice, some developers still take ownership of the tester role and focus their time on test activities.

The test analysis report is used by development teams to create test cases at, predominantly, unit and integration level. Testers also do some exploratory testing at the system level, however, they comprise a smaller portion of their test set. Conversely, they are still hard to automate and require human interaction (i.e., manual execution). In turn, tests for non-functional requirements, such as regulatory requirements, are executed separately and independently of functional tests. That enables independent test reports of products that need to comply with similar non-functional requirements but, in fact, have distinct functional requirements.

> *"For the functional tests the pass/fail is the primary measure . . . then for the non-functional tests there are more specific reports."* — Tele

Additionally, there is a i) variety of toolkits to assist developers with unit testing and ii) a test infrastructure for continuous testing during integration builds where test cases can be scheduled differently as well (e.g., daily, weekly runs). One of the main factors enabling this infrastructure is traceability on code level. However, when reaching higher levels of abstraction, such as the feature, traceability becomes more difficult, due to the need for maintenance of requirements update and the many-to-many relationship between features and test cases.

> *"There are more specific tests that we call them multi-feature test . . . more typically relating to multiple features."* — Tele

C. Manu

This company aims to become more agile throughout its different teams, and stakeholders are still converging towards standardized usage of tools and templates for specification documents. Therefore, only some teams in the company use tools that support traceability. However, for the other teams, most of the information transfer across artifacts is done manually or visually by consulting distinct files.

The existing template documents are designed to ensure some alignment between test specification and execution. For instance, they use a *design verification plan and report* (DVP&R) artifact that has fields so that engineers specify test configuration and group them in order to optimize test setup during test execution, since the test sessions are costly.

> *"That was the idea when I showed you the matrix in the beginning . . . So this is preventing you from doing the same test several times by accident."* — Manu

In addition to the DVP&R, they use test specification templates that require system engineers to manually fill out numerous fields, such as: test responsibility, milestones, goal, resources, test steps, etc. In addition, the same document has fields to report on the corresponding test executions.

As a consequence, it is unfeasible to align the test specification with the system requirements, rather the DVP&R should bridge them. It is hard for testers to separate a test specification from its execution, such that, for instance during regression tests, the same test specification being executed several times can quickly produce a lot of information from test reports

D. Auto

One of the main challenges stated by the stakeholders is the variety of usage across teams of the artifacts for system requirements and how they relate to testing. As a consequence, Auto focuses on ensuring consistence of a process to handle requirements and testing within its team throughout its department, due to the variety of disciplines involved in the system development (e.g., electrical, mechanical and software engineering).

> *"Not everyone at our department works with the same process."* — Auto

As an example, one of the stakeholders belongs to a team that uses an intermediate documentation to bridge unit tests and design information, but was not able to effectively align those unit tests to the system requirements.

> *"We call it 'documentation' because it's not requirements, since they are not tested by our testing group. We refer to them when we unit test, so it's possible for us to trace back what we have done in our unit tests. . . . but to not confuse anyone else, we call it 'design documentation'."* — Auto

In order to foster consistent communication across teams, the practitioners suggest creation of interfaces between hardware and software teams, to provide both perspectives and enable cross-functional development across those two levels. In addition, they argue that their REST alignment is existing but limited (for example, towards regression testing), i.e., that it is effective, but more expensive/time-consuming than necessary. Currently, there is impact analysis when changing requirements, but the company wants to also improve efficiency in selecting and executing non-changed requirements as part of their regression test sessions.

> *"How to work with development verification together with regression testing, that is something that we should have; this change structure to steer up verification on new development. But we have to have a product structure to steer up regression testing, I think, that is one idea."* — Auto

Ultimately, we identify that stakeholders want to leverage alignment between requirements and tests, especially to improve automation. The challenges for them is, however, to determine and distinguish which information separates tests from requirements and design.

> *"Test cases could serve as means to discover what the system should do, since we do design documentation more to document what it can do, then the tests are the description of what it should do, and I think that's how we can live with design and tests, rather than requirements and design as we do today."* — Auto

E. Discussion

In summary, Tele has stated that its challenge is to better align three elements: requirements specification, test cases and the platform responsible for test execution and management to benefit from updates in requirements and automatic selection of test cases to optimize requirements coverage. Manu's challenges are mainly related to consistent usage of their tools and document templates, since the resulting improvement on traceability can address their issues with the amount of manual work involved in creating and managing test specifications. Auto's challenges are to bridge the agile software development and requirements management in order to be agile at higher levels of the product. One of their suggestions was to represent higher level information as interfaces to improve communication among different experts (e.g., software and mechanical engineers), and build infrastructures with baselines that should be constantly tested.

Regarding RQ1, the main RE-related challenge from companies is the downstream alignment between requirements and tests to foster automated execution and support the team in communicating better. Even though all companies are implementing agile methods, they are in different stages of that implementation, thus, each has specific challenges to overcome with respect to REST alignment detailed next in Section IV.

IV. Alignment challenges and agile RE practices

Here, we summarize the results by connecting each company's results to the challenges related to RE and System Testing (REST) alignment in order to answer RQ2 and RQ3. Table II contains the alignment challenges as presented by Bjarnason et al. [4] and the corresponding company that stated facing such challenges. Challenges *Ch3*, *Ch4*, *Ch6* and *Ch7* comprise, respectively, requirements specification quality, system testing quality, requirements abstract levels and traceability. Note that the unchecked cells in the Table II indicate that we cannot confirm, based on the collected data, whether that the corresponding company faces that particular challenge or not.

For each challenge listed, we present: *i*) a description of the challenge (for detailed description refer to Bjarnason et al. [4]), *ii*) the risk of not addressing the challenge, *iii*) the observations collected from the interviewed companies and *iv*) a brief discussion on how agile RE practices can help to address the corresponding challenge in order to improve REST alignment in our context.

Ch1: Sometimes, practitioners feel that goals are missing or unclear, which could result in requirements misunderstanding, lack of insight into and awareness of different perspectives. **The risk:** Questionable decisions and costly requirements changes at a late stage in the development cycle. **Results:** For our cases, this is one of the main challenges to scale RE in agile because agile methods emphasize reactivity and informal communication [6, 4] that are hard to guarantee across multidisciplinary teams within a large organization. **Agile RE practice(s):** Suggested practices for this challenge should encourage communication and interaction across teams. The companies already use *cross-functional teams* with satisfactory communication. We suggest to also strike a balance between *face-to-face communication* and *requirements modeling* since focusing on them alone is a hindrance for, respectively, large-scale RE and agile RE.

Ch2: At the product level, weak cooperation affects negatively the alignment, specially at team and organizational boundaries where cooperation between people is required. Ch1 and Ch2 are similar, but they distinguish themselves in items, such as goals and perspectives (Ch1), and how to cooperate in communicating those items (Ch2). **The risk:** Increased lead times, additional rework, and conflicts in resource allocation

TABLE II
RESULTS FROM DATA COLLECTION RELATING THE CHALLENGES IN REST ALIGNMENT [4] WITH AGILE RE PRACTICES [9].

Id	Challenges in REST alignment	Tele	Manu	Auto	Agile RE Practices
Ch1	Aligning goals and perspectives within an organization		X	X	Face-to-face communication, req. modelling
Ch2	Cooperating successfully		X	X	Cross-functional teams
Ch3.1	Defining clear and verifiable requirements	X	X	X	Change and req. management, acceptance tests
Ch3.2	Keeping requirements documents updated	X	X	X	
Ch4.1	Full test coverage	X		X	Req. prioritization, management and continuous planning
Ch4.2	Defining a good verification process		X		
Ch4.3	Verifying quality requirements		X	X	
Ch5	Maintaining alignment when requirements change	X	X	X	Req. management, continuous planning, shared conceptualizations
Ch6.1	Defining requirements at different abstraction level	X			Iterative requirements, req. management
Ch6.2	Coordinating requirements at different abstraction levels	X	X	X	
Ch7.1	Tracing between requirements and test cases	X	X	X	Continuous planning and review meetings
Ch7.2	Tracing between requirements abstraction levels	X	X	X	
Ch8	Time and resource availability	X	X		Req. prioritization
Ch9	Managing a large document space	X	X	X	Req. management, prioritization and continuous planning

between projects. **Results:** In our data collection, Manu and Auto expressed concerns in finding isolated resistances across some teams when transitioning towards agile methods. Although teams do collaborate, the adoption of tools, templates and processes requires on-boarding among stakeholders. **Agile RE practice(s):** Those companies have already started to use *cross-functional teams* to increase communication and interaction between stakeholders (especially in their development teams). By involving, as well, different roles and expertise in product development they are able to leverage knowledge sharing to facilitate on-boarding.

Ch3: Poor requirement specification leads to missing information, hindering developers and testers to write and test the software. **The risk:** Increased testing effort and the risk to misinterpret and fail to deliver customer value, which is essential for agile RE. **Results:** All case companies reported challenges in dealing with their requirements specification. The difficulty is to create testable requirements[3] that are easily updated and reflect changes to the test artifacts. **Agile RE practice(s):** A combination of practices apply here, such as *change management*, *requirements management* and *acceptance tests*. These enable stakeholders to cope with the dynamic nature of agile RE, since the effort in maintaining links between tests and requirements information can be delegated to tools instead of people.

Ch4: Similarly to Ch3, poor quality in test artifacts and activities (e.g., unclear or incomplete test artifacts or process definition) affects alignment to achieve full test coverage, verification of quality requirements, as well as compliance to a verification process. **The risk:** Increased cost and effort when dealing with late requirements changes, since testers may not know how to promptly verify the changes and assess full test coverage. **Results:** Test coverage and non-functional tests become particularly challenging, since the information can be scattered between artifacts and teams in the organization. **Agile RE practice(s):** Ideally, *testing before coding* should be feasible for most large-scale agile projects. However, dependencies of modules and sub-systems across teams can be a hindrance for test case maintenance, such that *requirements prioritization, management and continuous planning* can, instead, leverage V&V planning and reduce test effort, especially for regression test sessions.

Ch5: This is related to consistent updating of RE and ST artifacts, and being able to assess the impact of changes. That maintenance effort is important when requirements are actively used. Otherwise stakeholders may need to search for artifact information from other sources. **The risk:** Wasted effort in development or testing activities along with inconsistent deliverable, either because the developers are not aware of changes, thus do not implement them, or because the tests no longer validate what is actually described in the updated requirements. **Results:** Companies do not yet use techniques for REST alignment [4, 8], even though they want to, particularly, improve and automate regression and NFR tests. **Agile RE practice(s):** The suggestions here overlap, to some extent, with **Ch3** and **Ch4**, since stakeholders need diligence to adopt *requirements management*, *continuous planning* and *shared conceptualizations* [9] to bridge both ends of the V-model across the system's levels.

Ch6: While Ch5 addresses consistences between deliverable and artifacts, Ch6 includes the consistence of the information being represented in the requirements and test artifacts at different abstraction levels, in order to enhance downstream coverage in the V-model. **The risk:** Increased test effort and costs to validate specific requirements, and the difficulty in selecting tests for subsystems or components being developed by different teams. **Results:** As mentioned in Section III-A, all companies report on challenges about levels and decomposition of their requirements from customer requirements to system requirements and downstream through the V-model (unit tests). **Agile RE practice(s):** At each decomposition, the risk is to lose traceability and become overwhelmed with cluttered documentation, hence *iterative requirements* along with *requirements management* can help stakeholders alleviate the cluttering and gauge large-scale agile requirements engi-

[3] Our interpretation is to have requirements specifications that one can easily derive, and ideally execute, test cases from.

neering.

Ch7: Creation and maintenance of trace links between requirements and test cases is costly. This cost involved in both introducing and maintaining traceability often compensates for the costs of lacking traceability. **The risk:** Increased effort in assessing change impact. Then stakeholders will be tricked into guessing the connections between artifacts, again propagating inconsistencies in the process. **Results:** Traceability is a challenge for all interviewed companies, even though they have guidelines and tool support to provide trace links to their artifacts. The main part of the problem is adopting the corresponding tools and the habit of creating and maintaining the trace links in the tools. **Agile RE practice(s):** *Continuous planning* and *review meetings* can encourage diligence towards maintenance of trace links, since stakeholders would need to refer to their status during each retrospective [9].

Ch8: The alignment between the demanded functionality and quality levels expected of the products versus the amount of testing that can be performed. **The risk:** Unnecessary increase in costs or, in case resources for satisfactory test coverage are unavailable, the inability to properly verify and/or validate the product. **Results:** This was explicitly discussed by Tele and Manu. Particularly regarding costs for test execution, because testing sessions usually require allocation of expensive competences or hardware/mechanical components, thus increasing the testing costs. **Agile RE practice(s):** In order to apply existing test case selection techniques to cope with resource constraints, stakeholders need to do *requirements prioritization*, so that test cases can be automatically selected. Most automatic techniques assume, however, traceability between test and requirements since manual selection can be costly and error prone [15].

Ch9: If not managed properly, the information in requirements or test databases will become redundant and unnecessary. **The risk:** Inconsistent (or even conflicting) information that accumulates when that information is updated, hindering the creation of baseline versions (and ultimately affecting regression tests). For instance, outdated test cases may trigger failures related to obsolete requirements rather than actual faults in the system under test. **Results:** This was reported by all companies, because agile methods do not encourage large pieces of documented information [6, 4]. However, they companies might need this to comply with regulatory agencies and legacy requirements. **Agile RE practice(s):** Bjarnason et al. [4] recommend *user stories* to address the challenge of traditional requirements documentation along with *requirements modeling and management* to address the need for shared information across teams in large-scale companies. There is some connection to **Ch7** as well, since traceability tools can facilitate maintenance of this document space, even if smaller.

A. Summary and discussion

When analyzing the REST alignment challenges to answer RQ2, we saw that *the main challenges, for the companies, are* (Table II) : requirements quality Ch3.1, Ch3.2), alignment maintenance (Ch5), abstract levels (Ch6.2), traceability (Ch7.1, Ch7.2) and management of requirements documents (Ch9). Surprisingly, the major challenge is related to RE rather than test artifacts (Ch3.1, Ch3.2 and Ch 3.3), since most companies have to first move towards large-scale and consolidate their RE needs so that, then, they can benefit from that in their testing activities.

Answering RQ3, *the most recommended practices* for the companies are emphasizing requirements management, prioritization, and updates, along with continuous planning. Requirements management can be a rather general practice, but here we emphasize that using tools with traceability and control version support yielding two main benefits: the allow stakeholders to automate the maintenance of requirements information, and to reuse that information to leverage automated and agile testing [5].

V. Threats to validity

We relate our choice of theory and case (see Table I in Section III) towards threats to construct validity. In order to comply with existing constructs we refer to existing work that has been validated in previous studies, representing practices and challenges in large-scale agile system development. There is still a risk that the constructs are unsuitable due to the reliability of their original studies. However, we observe consistence in the identified challenges throughout our and the original studies.

Similarly, external validity depends on the extent that we can generalize our findings, given that the constructs come, also, from case studies that might have low external validity, since the cases' conclusions are limited and still specific to a company or project within that company where data was collected. In addition, all three companies are from Sweden. Nonetheless, our objective is exploratory at this stage, and we intend to expand towards generalization by including other frameworks (e.g., LeSS and SAFe) and, possibly, more practitioners in future studies.

The main internal validity threat relates to the interviews performed during the workshop, since we allow the focus groups to diverge slightly from the schedule and the semi-structure interview, in order to keep an open dialogue and allow description of tacit knowledge for practitioners in each company. We mitigate this threat by using consistent[4] instruments and methods throughout the different workshops and analysis. Ultimately, we cannot emphasize significantly control of data collection, since this is a qualitative study and we assume that each company has its own peculiarities, as a consequence of their disparate domains.

Regarding reliability, there are several limitations in our study, specially since data sharing is limited due to non-disclosure agreements with the participating companies. Conversely, two points improve reliability of our study. First, all authors in this paper were involved in this study, particularly

[4] Examples of instruments are the forms to collect artifact data, whereas analysis instruments are the frameworks used to identify challenges.

during data collection. In addition, we planned so that not all authors would attend all workshops, hence reducing the risks of bias when conducting the interviews, or coding the results. Second, we re-use existing frameworks from reliable studies in literature, thus providing other researchers aiming to reproduce our study to refer to those additional sources for any extra information needed on our constructs.

VI. Conclusions

We investigate REST alignment challenges [4] in three large companies aiming to move towards large-scale agile system development. Based on data collected from interviews with focus groups from each company, we also discuss how existing agile RE practices [9] can help stakeholders to overcome the challenges to improve REST alignment. Our results show that the companies need to first address many RE related challenges to consolidate their creation and maintenance of requirements artifacts, in order to enable REST alignment, eventually leading to better management of test artifact and automated test activities [5].

As limitations we highlight that we used one set of REST challenges and agile RE practices, when several frameworks could also be included in our study. Nonetheless, our conclusions are a first step for future work where different frameworks can now be included and compared. Even though the REST alignment challenges refer to alignment between RE and software tests, our contexts includes system testing and still several challenges were observed and reported by the companies.

Besides expanding the theory framework in our case study by including other lists of challenges and practices, we plan to analyze the data from an agile testing perspective focusing on agile testing challenges and practices reported by Crispin and Gregory [5]. In addition, we plan to collect more data from the companies to achieve more concrete RE practices that can be implemented and evaluated in terms of large-scale system development.

Acknowledgments

We thank all participants in this study for their great support, deep discussions, and clarifications. This work was supported by the Chalmers Software Center Project 27 on RE for Large-Scale Agile System Dev. and the SIDA BRIGHT project.

References

[1] 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*, 2016.

[2] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl, “The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson,” in *ACM / IEEE Int. Symposium on Empirical Softw. Engg. and Measurement*, 2013, pp. 348–356.

[3] T. Dingsøyr and N. B. Moe, “Research challenges in large-scale agile software development,” *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 5, pp. 38–39, 2013.

[4] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, “Challenges and practices in aligning requirements with verification and validation: A case study of six companies,” *Empirical Softw. Engg.*, vol. 19, no. 6, pp. 1809–1855, Dec. 2014.

[5] L. Crispin and J. Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams*, 1st ed. Addison-Wesley Professional, 2009.

[6] V. T. Heikkilä, M. Paasivaara, C. Lasssenius, D. Damian, and C. Engblom, “Managing the requirements flow from strategy to release in large-scale agile development: a case study at ericsson,” *Empirical Softw. Engg.*, pp. 1–45, 2017.

[7] M. Felderer and A. Herrmann, “Manual test case derivation from UML activity diagrams and state machines: A controlled experiment,” *Information and Software Technology*, vol. 61, pp. 1–15, 2015.

[8] M. Unterkalmsteiner, R. Feldt, and T. Gorschek, “A taxonomy for requirements engineering and software test alignment,” *ACM Trans. on Softw. Engin. and Methodolology*, vol. 23, no. 2, pp. 16:1–16:38, Apr. 2014.

[9] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, “A systematic literature review on agile requirements engineering practices and challenges,” *Computers in Human Behavior*, vol. 51, Part B, pp. 915 – 929, 2015.

[10] B. Robinson, M. D. Ernst, J. H. Perkins, V. Augustine, and N. Li, “Scaling up automated test generation: Automatically generating maintainable regression unit tests for programs,” in *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Nov 2011, pp. 23–32.

[11] T. Gorschek and A. M. Davis, “Requirements engineering: In search of the dependent variables,” *Inf. Softw. Technol.*, vol. 50, no. 1-2, pp. 67–75, 2008.

[12] J. Kukkanen, K. Väkeväinen, M. Kauppinen, and E. Uusitalo, “Applying a systematic approach to link requirements and testing: A case study,” in *16th Asia-Pacific Software Engineering Conference*, Dec 2009, pp. 482–488.

[13] E. Uusitalo, M. M. Komssi, M. Kauppinen, and A. M. Davis, “Linking requirements and testing in practice,” in *2008 16th IEEE International Requirements Engineering Conference*, Sept 2008, pp. 265–270.

[14] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical Softw. Engg.*, vol. 14, no. 2, pp. 131–164, 2008.

[15] F. G. de Oliveira Neto, R. Torkar, and P. D. Machado, “Full modification coverage through automatic similarity-based test case selection,” *Information and Software Technology*, vol. 80, pp. 124 – 137, 2016.