2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement

The impact of agile principles and practices on large-scale software development projects

A multiple-case study of two projects at Ericsson

Lina Lagerberg, Tor Skude, Pär Emanuelsson and Kristian Sandahl (corresponding author)
Department of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden
linalagerberg@gmail.com, tor@skude.se,
{par.emanuelsson, kristian.sandahl}@liu.se

Daniel Ståhl
Ericsson AB
Datalinjen 3
SE-581 12 Linköping, Sweden
daniel.stahl@ericsson.com

Abstract— **BACKGROUND: Agile software development methods have a number of reported benefits on productivity, project visibility, software quality and other areas. There are also negative effects reported. However, the base of empirical evidence to the claimed effects needs more empirical studies. *AIM:* The purpose of the research was to contribute with empirical evidence on the impact of using agile principles and practices in large-scale, industrial software development. Research was focused on impacts within seven areas: Internal software documentation, Knowledge sharing, Project visibility, Pressure and stress, Coordination effectiveness, and Productivity. METHOD: Research was carried out as a multiple-case study on two contemporary, large-scale software development projects with different levels of agile adoption at Ericsson. Empirical data was collected through a survey of project members. RESULTS AND CONCLUSIONS: Intentional implementation of agile principles and practices were found to: correlate with a more balanced use of internal software documentation, contribute to knowledge sharing, correlate with increased project visibility and coordination effectiveness, reduce the need for other types of coordination mechanisms, and possibly increase productivity. No correlation with increase in pressure and stress were found.**

Keywords—agile software development, large-scale software development, multiple-case study, survey, empirical software engineering

I. INTRODUCTION

Agile software development (ASD) methods are often advertised as a contrast to the traditional, plan-driven approach to software development [1] and the reported and argued benefits are numerous. ASD methods are claimed to increase software quality [2], improve communication [3] as well as coordination [4] and increase productivity [5], to name just a few. However, the lesser emphasis on documentation could lead to inferior knowledge management in the long term [6] and the focus on constant delivery of results could increase the pressure on the software professionals [4] [7]. More empirical studies on the effects of ASD methods are asked for by several researchers, especially on which effects could be expected in a large-scale, industrial setting [1] [8].

The contribution of this research is empirical evidence on the impact of using agile methods in large-scale software development at Ericsson. The design of study comprises two projects. The first project, named Project A, is a mature project using a classical waterfall method [9], which has been improved over many years, now also including a few agile practices. The second project, named Project B, is recently started with the clear intention of being an agile project. The projects are particularly interesting for comparison since Product B is planned to replace Product A in future. To the best of our knowledge this study is the only one that compares two simultaneously running projects.

Project A and Project B are large commercial projects with 420 and 120 members respectively. The projects have a large number of teams (15 and 14 respectively). The developers are mature, with 56% and 37% having more than one year ASD experience, respectively. Dybå and Dingsøyr [1] made a systematic review of experience based agile project studies. From 1996 studies they filtered out 36 studies that were experience based, out of which only four were investigating projects with mature developers (more than one year experience of ASD).

The data collection was made through a survey of project members with a web-based questionnaire focusing on effects from ASD found in a systematic literature review [10]:

1) *Less and insufficient documentation.*
2) *Facilitated knowledge sharing.*
3) *Increased project visibility.*
4) *Increased pressure and stress.*
5) *Effective coordination.*
6) *Increased productivity.*

This article is organized as follows: Section II gives the definition of agile principles and practices used in the study, Section III describes the study context and methodology, Section IV presents the results, which are related to earlier studies in Section V. In Section VI the results are summarized and discussed. Section VII concludes the article.

978-0-7695-5056-5/13 $26.00 © 2013 IEEE
DOI 10.1109/ESEM.2013.53

II. Agile Principles and Practices

The Agile Manifesto [11], created in 2001, lists a set of values upon which ASD relies. Along with these values, there is also a set of principles. Principles are "*domain-specific guidelines for life*" [12, p. 15], showing how the values can be applied in different areas. Thirdly there are practices, which are even more concrete, specific applications of the values and principles. This research was focused on the effects of agile practices.

A. Defintion of studied agile practices

We studied the agile practices in use in the two projects, where effects were reported in earlier research [10]. The studied practices were defined as follows:

Daily meetings - Agile development teams should have daily, short meetings to keep everyone up to date with the current progress and what others are doing [13].

Demo - At the end of an iteration, a working product incorporating the completed work should be demonstrated to the Product Owner and other stakeholders [14].

Empowerment – refers to agile development teams being autonomous, self-organizing and self-managing [14].

Information radiators – are highly visible physical displays of updated information about the software development project [13]. A Scrum task board is an example of an information radiator.

Iteration planning – is done to decide what should be done in the coming iteration. Often includes preparing a sprint backlog, breaking down requirements to smaller work items, estimation of effort, as well as planning of what features should be included in coming releases [14].

Iterative and incremental development – is when software development is done in iterations, often called sprints. In each iteration, visible functionality – increments - is added to a working piece of software [13].

Open office space – is when development teams are located together in an open environment that enables face-to-face communication [15].

Retrospectives – are meetings held at the end of an iteration to reflect on what went well and what could be improved for the next iteration [13].

Sign up – the practice of sign up means that teams members sign themselves up for tasks, instead of being assigned [13].

Task board – is a place where the progresses of tasks are visualized [13].

Whole team – means that an ASD team consists of people with a broad variety of skills, including analysts, developers and testers [15].

III. Research Design

To investigate what impact the use of agile principles and practices have in Project A and Project B, we chose to regard each project as a separate case and conduct a holistic multiple-case study [16]. This methodology was chosen since it gives the opportunity to collect both quantitative and qualitative data [16] and gain a complex, detailed understanding of how agile principles and practices work in a natural large-scale, industrial setting [17], as opposed to e.g. an experiment where the issue under study is separated from its context [17]. The two studied projects were chosen to show different perspectives of a development process: one plan-based and one agile approach with different amounts of implemented agile practices. The study was performed during the fall and winter of 2012 and 2013.

The study is an explorative study in the sense that there is no predefined hypothesis to be tested; rather the goal is to contribute as much knowledge as possible within the selected areas of study. This has to be balanced against the time we can expect the informants to spend on the survey.

A. Research context

1) Company

Ericsson is a global telecommunications company that develops, produces and sells telecommunications equipment, services and software. Software revenue accounted for 24% of the total revenue in 2011, making Ericsson the world's fifth largest software company [18].

2) The studied projects:

a) Project A – the plan-based project

Project A is responsible for designing, developing and maintaining new versions of Product A – a product that supports telecommunication networks and have been in development for more than ten years. It has a hierarchical project organization and runs on a biannual release schedule, following a traditional waterfall model [9], with clearly defined and separate roles for separate departments. Extensive, document-intense handovers are necessary between the departments. That being said, steps have been taken to address perceived challenges in integration, dependency management and flexibility, caused by said model by implementing certain agile practices (see Table II). For instance, several teams have

TABLE I CHARACTERISTICS OF PROJECT A AND PROJECT B

	Project A	**Project B**
Product age	More than 10 years	1 year
#People	420 (+480 external consultants)	120
#Teams	15	14
#People/#Teams	28,0	8,57
Development organization	Separate subprojects for System Analysis, Design and Integration & Verification	Cross-functional teams. Separate supporting teams.
Development team's responsibility	Horizontal: architecture layers	Vertical: features
Process model	Plan-based, waterfall	Scrum
Iteration length	1 month	3 weeks
Product release frequency	9 months	3 weeks (potentially)
Integration frequency	Once per month	Continuously
Architecture	Component-based	Service-oriented

begun working iteratively, some use Scrum, and continuous integration and test automation have been implemented in parts of the project. However, these efforts are typically partial; leading us to consider Project A to be a traditional, plan-based software development project.

b) Project B – the agile project

The purpose of Project B is to design, develop and deploy a product, Product B, which in the long run can replace Product A. The project was launched in the beginning of 2012 and has not yet released any product to customers. Project B uses an agile project management method inspired by Scrum, consisting of a flat organization where everyone reports to the same agile manager. The development process is inspired by feature driven development (FDD) and the entire project is executed in iterations. Cross-functional teams are responsible for all phases in the development process, but supporting teams handle the project management, continuous analysis, Continuous Integration (CI) framework execution etc.

Every cross-functional team includes system analysts, designers and testers, as well as a Scrum Master and a Product Owner. The teams in Project B are responsible for features, compared to product/architecture layers in Project A. To enable the team to deliver business value at the end of each sprint, each team requires vertical competence across all layers of the system architecture and also across the design, implementation and testing areas. This requires a competence shift compared with the specialized functional departments traditionally used within Ericsson. Project B has implemented a large number of agile practices (see Table II) and we consider it an agile software development project.

In parallel with the new product, Project B designs and implements an extensive end-to-end CI framework. The idea is that the integration should be done automatically upon code commit, the tests should be automatically executed and feedback on build and test results should be given quickly to the developers. If the tests on all levels pass, the deployment should be made automatically.

We consider that Project A and Project B differ sufficiently in their ways of working and are sufficiently similar in other aspects that the impact of using agile principles and practices can be studied by comparing the two projects. Regarding similarities, they both operate in the same corporate context and the people working in them don't differ appreciably in education level or experience. Additionally they develop a product with the same functional requirements, which is to be sold to similar customers. Finally, the two projects have the same technical preconditions. They are both large-scale projects with hundreds of software professionals and more than ten different teams.

TABLE II ADOPTION OF AGILE PRACTICES IN PROJECT A AND PROJECT B.

	Project A	Project B
Product backlog	Partial	Full
Collective ownership	Partial	Partial
Continuous integration	Partial	Full
Daily meetings	Partial	Full
Demos	Partial	Full
Empowerment	Not in use	Full
Iteration planning	Partial	Full
Iterative and incremental development	Partial	Full
On-site customer	Not in use	Not in use
Open office space	Full	Full
Pair programming	Not in use	Not in use
Refactoring	Not in use	Full
Retrospectives	Partial	Full
Sign up	Partial	Full
Simple design	Not in use	Not in use
Task board	Partial	Full
Whole team	Not in use	Full

B. Research methodology

1) Literature review

The first steps of a systematic literature review [19] was performed in order to collect propositions to what impact the use of agile principles and practices could have in Project A and Project B [10]. The search phrase *"agile software development" AND (effect* OR impact*)* was used in the Inspec database. Results were limited to articles written in English, Swedish and Spanish and published between 2005 and 2012. The initial result set included 281 hits. After removing duplicates, 229 hits were left. The title and abstract of these 229 hits were investigated and 77 articles were selected for further investigation. After reading first the conclusions and then the full articles, any articles that were not based on empirical studies in an industrial setting were culled. This resulted in 17 articles. Finally 5 articles referenced in the 17 selected articles were included, which yielded a set of 22 articles used for data extraction. The results of the literature review were synthesized in a conceptual framework, consisting of possible effects of agile principles and practices within six different areas in Project A and Project B. We chose to focus on reported effects of concrete practices, to enable for transferability and comparability between different projects. The 14 articles most relevant to the results presented in this article are presented in Table IX.

2) Data collection

Empirical data were collected through a survey of project members; the 1st, 2nd, and 5th authors had further full access to internal documents concerning the two projects.

The survey invitations were distributed via e-mail to a sample of 120 members of Project A and all 120 members of Project B, on two separate occasions during fall 2012. The surveys were open for one week (Project A) and two weeks (Project B) and hosted on a paid version of SurveyMonkey. The survey of Project A was responded by 34 persons, this is a response rate of 28%, which corresponds to 8.1% of the project members. The survey of Project B was responded by 52 project members, which corresponds to 43% of the total project members.

The survey questionnaire included 30 questions in seven different sections: Background questions and experience of software development, Experiences of ASD in Project A/B, Knowledge sharing in Project A/B, Software documentation in

TABLE III SURVEY RESULTS ON THE AMOUNT OF INTERNAL SOFTWARE DOCUMENTATION. RANGE OF VALUES 1-7. (SIGNIFICANT DIFFERENCES IN ITALLICS)

	Project A		Project B	
	μ	σ	μ	σ
Overall amount	*3.95*	1.40	*3.78*	1.10
Importance of internal documents	4.92	1.64	5.60	1.18
Change requests	4.46	1.29	3.88	0.90
Code comments	3.25	1.81	4.00	1.03
Design specifications	3.67	1.68	3.66	0.97
Process documentation	*4.64*	1.71	*3.79*	1.08
Requirement specifications	4.59	1.47	3.37	1.30
Technical documentation of code	3.16	1.57	4.22	0.89
Trouble reports	4.87	1.49	4.00	0.65
Work packages	4.30	1.52	3.86	0.89

Project A/B, Project visibility in Project A/B, Work environment (Pressure and stress) in Project A/B and Coordination in Project A/B. The questionnaire also included an open-ended question that gave the possibility to comment on the ways of working in Project A/B. Participation in the survey was voluntary and the responses were collected anonymously. The full survey and a summary of the results are available in the appendix of [10] for free at:

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-89658

3) Data analysis

The data from all sources were combined and analyzed using the pattern-matching logic [16], where the theoretical patterns synthesized in the conceptual framework were matched with the patterns visible in the gathered data. The survey was analyzed with help of the software IBM SPSS 20. The presence of the expected effects of ASD in Project A and Project B was investigated by studying the mean [μ] and standard deviation [σ] of the questions that corresponded to an effect. The interval scale [20] used for these questions ranged between one and seven, where the endpoints represented 'extreme values'. Accordingly a mean value above four was considered an indication that the investigated effect was present in the project. To investigate differences between the two projects, the non-parametric Mann-Whitney U-test for independent samples was used, since the majority of the variables couldn't be considered normally distributed in a Shapiro-Wilk test. The Mann-Whitney null hypothesis of equal means was rejected for an asymptotic significance (p) smaller than $\alpha=0.05$ and thus indicating a statistically significant difference in means between the two projects.

C. Validity of the study

A low response rate can threaten the validity of a survey [21]. We can note that even though participation in our survey was voluntary, it yielded an acceptable response rate, especially in Project B. The respondents also represented different functional roles and levels of experience in a satisfactory manner. Another possible threat to the validity is that the survey questions don't capture the real perceptions of the survey respondents [20]. To mitigate this risk we let persons with domain knowledge, including the project management of Project B, review the questions and the methodology and also piloted the questionnaire beforehand.

To improve the validity, the study results was presented to Project A and Project B for validation and they were given the opportunity to present their views not only on the data, but also on our analysis.

IV. RESULTS

In this section we present the survey results of agile principles and practices found in Project A and Project B within the six studied areas. In each area we also formulate the findings and discuss possible explanations. We believe that readability is increased by focusing on one practice at the time, rather than splitting the content in one survey result and one discussion section.

A. Internal software documentation

The members from both Project A and Project B perceived the "overall amount of internal software documentation" in their project as just right (see Table III). There was however, a significant difference between the two projects regarding several software documentation items. The internal software documentation items in Project B were on average perceived as very close to just right, while they in Project A were perceived as slightly too much. We can therefore see in Project B has a better, more balanced use of internal software documentation, rather than an insufficient use, as reported by [6]. There could be several explanations why we didn't find the same pattern as [6]. A first one is that the findings of [6] simply aren't valid for all agile projects, or for large-scale agile projects. It's also possible that Project B isn't fully following agile principles and practices in this regard. Assumable general Ericsson policies and directives affect the software documentation produced in Project B.

The survey respondents from Project B didn't perceive internal documentation as significantly more important, nor believe that it can be replaced with face-to-face conversation to a larger extent than the survey respondents from Project A. We therefore believe that internal software documentation is important, no matter the ways of working.

Surprisingly enough the survey respondents from Project A spent less time on documentation than the respondents from Project B. In Project A, 58.3% of the respondents spent 15 minutes or less daily on producing internal software documentation, which is line with agile projects [6]. However, only 43.5% spent 15 minutes or less in Project B. Again, this may be explained by the large scale of both projects and general company documentation policies. It can also be a question of attitude. Members of Project B are working with the intention of being agile, which is generally conceived as a positive, more modern way working. Thus they might be less apt to complain.

B. Knowledge sharing

The abstract concept of knowledge sharing was measured from three aspects: to what extent the project members help each other with their project work, to what extent they feel that their project members can help them and how much they have learned about other functional areas from team members. The survey results on knowledge sharing are found in Table IV.

There was no statistically significant difference in how much the project members got help or to what extent they felt that their team member have the necessary capabilities to help them. It was clear that knowledge was shared within both projects. There was however, a statistically significant difference between the two projects in the extent agile practices contribute to knowledge sharing, which supports the findings of [4] [22] [23] [24]. The members of Project B felt that the iteration planning meetings give insights about the project to a higher extent than the members of Project A. The members of Project B also felt that they get insights from the retrospective meetings and the demos to a much higher extent.

In Project B there was further a significantly larger knowledge sharing between different functional roles, compared to Project A. The members from Project B had gained more insights from other functional areas than the members from Project A and had also performed a greater number of activities from different functional areas. We can interpret this that the practice of whole team is more widely spread in Project B, compared to Project A, and that this has had positive effects on the knowledge sharing. This would be in line with the findings of [4] and [24], who reported that the practice of whole team widens the developer's knowledge and make team members able to perform each other's activities.

C. Project visibility

Based on the survey data (presented in Table V), there is a considerable difference of the perceived project visibility between the two projects. The members of Project A don't consider themselves aware of the statuses of the other project teams, while the members of Project B do. The same statistically significant difference was present for the entire project's status; Project A members didn't consider themselves aware, while the Project B members did. This supports the findings of [2], [3] and [25]. The shorter development cycles and the constant delivery of value are believed to have facilitated for the project members to grasp the status of the project.

When the survey respondents were asked outright what practices they considered important for their awareness of the project's status, continuous integration and iteration planning received the highest scores, while open office space, demos and task boards received lower scores, although still considered important in both projects. Both projects also considered themselves aware of the status of their own project team. Our results thereby confirm the findings of [3], that project visibility is increased by several agile practices. They also confirm the findings of [25], that it is possible to achieve increased project visibility in large-scale projects.

TABLE IV SURVEY RESULTS ON KNOWLEDGE SHARING RANGE OF VALUES 1-7. (SIGNIFICANT DIFFERENCES IN ITALLICS)

	Project A		Project B	
	μ	σ	μ	σ
Get help	4.77	1.45	4.98	1.27
Team members possess necessary capabilities	5.38	1.42	5.53	1.47
Number of activities performed	*1.67*	1.30	*2.44*	1.62
I have gained insights from other functional areas than my own	*4.27*	1.86	*5.46*	1.68
The sprint planning meetings give me insights about the project I wouldn't get otherwise	*4,33*	1,65	*5,40*	1,40
The retrospective meetings give me insights about the project I wouldn't get otherwise	*3,70*	1,69	*5,32*	1,67
The demos give me insights about the project I wouldn't get otherwise	*4,32*	1,84	*5,22*	1,64

TABLE V SURVEY RESULTS ON PROJECT VISIBILITY. RANGE OF VALUES 1-7. (SIGNIFICANT DIFFERENCES IN ITALLICS)

	Project A		Project B	
Awareness of	μ	σ	μ	σ
Team status	5.60	1.29	6.04	1.21
Other teams' status	*3.32*	1.70	*4.27*	1.94
Project status	*3.64*	1.87	*4.98*	1.64

D. Pressure and stress

The average levels of negative stress and pressure stated by the survey respondents from Project B are all slightly higher than the average levels stated by members from Project A (see Table VI). However, the difference was not statistically significant. The survey respondents from both projects felt moderately stressed and pressured by their project work. The negative stress caused by *demos* and *sign-up*, as reported by [4] and [23], was present to a moderate extent inside both projects. Especially the *demos* are causing negative stress for the members of Project B. The negative stress caused by a pressure to report progress on the *daily meetings*, reported by [7], does not seem to be present to a large extent neither in Project A nor in Project B.

In summary, Project B with more agile principles and practices doesn't increase the overall level of stress and pressure to a large extent, but some agile practices are perceived as stressful. Also, stress is something very individual and there was a high dispersion in the answers that these effects vary from person to person.

TABLE VI SURVEY RESULTS ON PRESSURE AND STRESS. RANGE OF VALUES 1-7. (SIGNIFICANT DIFFERENCES IN ITALLICS)

	Project A		Project B	
Level of stress caused by	μ	σ	μ	σ
Project work in general	*3.59*	1.62	*3.93*	1.72
Demos	3.82	1.91	3.93	1.84
Sign-up	3.19	1.42	3.89	1.79
Daily meetings	2.94	1.95	3.32	1.99

E. Coordination effectiveness

The level of coordination effectiveness in the two projects was asked for explicitly in the survey and calculated from a set of survey questions, constructed from the model of coordination effectiveness proposed by [26]. The survey results on coordination effectiveness are shown in Table VII. From the survey it was clear the level of coordination effectiveness in both projects was high. Finally we saw that the members from Project B felt more aware of the other teams, than the members from Project A. We consequently conclude that Project B has achieved a higher both inter-team and intra-team coordination effectiveness than Project A with less overhead. This supports the findings of [4].

We believe that the most important contributor to the higher coordination effectiveness is the use of whole teams in Project B and the fact that these teams are responsible for features instead of architecture layers, as this has considerably reduced both the handovers between functional departments and within them, compared to Project A. The design of the open office space in Project B does also increase coordination effectiveness. While the members in both projects sit together in an open office space, the cross-functional teams sit together in Project B, and as such every team member is in the proximity of most persons he or she needs to coordinate with. This is in line with the concept of proximity that [4] described as an important coordination mechanism. Additionally it can be seen that the daily meetings form an important coordination mechanism in both projects, as proposed by [4] and [22]. Regarding information radiators, or more exactly task boards, they seem to be increasing coordination effectiveness inside the team, but not between teams. No significant difference in this was found between the projects and neither in Project B, where task boards are more consistently used by all teams did the project members feel that it helped them to know what the other teams are doing to a large extent.

TABLE VII SURVEY RESULTS ON COORDINATION. RANGE OF VALUES 1-7. EFFECTIVENESS. (SIGNIFICANT DIFFERENCES IN ITALLICS)

	Project A		**Project B**	
Coordination effectiveness	μ	σ	μ	σ
Perceived	4.80	1.73	5.14	1.71
Calculated	*4.15*	1.63	*5.25*	1.03
Daily meeting help in knowing what team members are doing	*6.17*	*1.10*	*5.64*	*1.42*
Team task board help in knowing what team members are doing	*5.33*	1.80	*5.17*	1.76
Other teams' task boards help in knowing what they are doing	*3.93*	1.79	*4.18*	2.09

F. Productivity

Respondents from both projects mentioned productivity in the open-ended survey question. They thought that "time is lost on planning" when working in an agile way. It is true that the recurrent meetings associated with agile ways of working, e.g. daily meetings, retrospectives and iteration planning meetings take quite a lot of time. However, if the meetings were decreased, the many positive effects of the meetings we found both in earlier research and in our own study, among them improved project visibility and knowledge sharing, might be lost. More research on the effects of meetings is needed to investigate this argumentation.

G. Additional results

A positive impact of several agile principles and practices was reported by the participants of the survey in Project A. This indicates that a partial implementation can give positive results. However, we see that the full implementation of the agile practices across the entire project in Project B gives stronger results. In summary, the way agile principles and practices are implemented in Project A, but especially in Project B, was found to have many positive effects according to our informants' perceptions. We believe this provides evidence that agile ways of working can be beneficial in large-scale implementations.

TABLE VIII OUR FINDINGS IN RELATION TO RELATED WORK.

ID	Documen-tation	Knowledge sharing	Project visibility	Coordination effectiveness	Productivity	Stress	Major method	Major data source	Population / team size
Our work	Balanced	Facilitated	Increased	Increased.	Possibly Increased	Not increased	Multiple-case study.	Survey. 86 respondents Ericsson. Metrics	420+120
S1		Facilitated	Increased				Single-case study	12 semi-structured interviews. 1344 defect reports.	6
S2	Insufficient.	Insufficient.	Increased				Multiple-case study.	Semi-structured and group interviews F-secure.	6+6
S3		Facilitated		Increased.		Increased.	Multiple-case study.	11 semi-structured interviews. 3 companies.	6+10+5
S4	Reduced. Insufficient	Insufficient					Survey.	Survey. 79 respondents, 8 teams, 13 countries.	Unkown
S5						Increased	Single-case study.	20 semi-structured interviews.	Unknown
S6		Facilitated.		Increased.			Case study.	Observations.	Unknown
S7	Reduced.	Facilitated				Increased	Multiple-case study.	25 interviews.	10+9+8
S8	Reduced.	Increased		Insufficient	No increase		Singl- case study	33 interviews Ericsson	43+53+17
S9			Increased				Survey.	500 respondents Nokia.	> 1000
S10							Survey.	129 respondents.	Unknown
S11					Increased		Experiment/ case study.	eXPERT metrics. Rila Solutions.	4+4
S12		Facilitated.			Increaseed		Single-case study	XP metrics (Defect reports, LOC). Interviews. Sabre Airlines Solutions.	6+10
S13			Increased.		Increased		Multi-method: field interviews, survey, case study	36 semi-structured interviews. 565 survey respondents. Metrics (on-time and on-budget completion, software functionality).	Survey: 1750.
S14					Increased.		Experience report.	Internal surveys.	150 teams

V. RELATED WORK

The number of studies on ASD conducted in a large-scale industrial setting is limited [1] [8]. We therefore chose to include also studies conducted in small projects in our theoretical basis and the set of the related work [10]. In Table IX we show the earlier studies included in the theoretical basis most closely related to the results presented in previous section. Table VIII compares our findings with the findings of the earlier studies within the six studied areas, as well as within three study characteristics. In general, our findings confirm the effects observed in other industrial studies. However, we can note deviations within some areas. S2 and S4 reported that agile projects may produce an insufficient amount of internal software documentation, while we observed that the agile ways of working in Project B correlated to a more balanced use of internal software documentation. S3, S5 and S7 all reported on an increased level of pressure and stress when working in an agile way, but we saw no increase in the level of pressure and stress when comparing Project A and Project B. S8 and S9 are similar to our study conducted in a large-scale, industrial setting. S8 is conducted at Ericsson, but found no increase in productivity when changing from a plan-driven to an agile way of working.

VI. DISCUSSION

A. Summary of findings

This study investigated the impact of using agile principles and practices in two large-scale software development projects at Ericsson – the largely plan-based project Project A and the entirely agile project Project B – within six different areas:

Internal software documentation, Knowledge sharing, Project visibility, Pressure and stress, Coordination effectiveness, and Productivity.

The found effects in the studied projects were:

1) *Balanced use of internal software documentation.*
2) *Facilitated knowledge sharing.*
3) *Increased visibility of the status of other teams and the entire project.*
4) *Effective coordination, with less overhead.*
5) *Possibly increased productivity.*

It was further found that: (i) Internal software documentation is important also in agile software development projects, and cannot fully be replaced with face-to-face communication. (ii) It's possible to make a partial implementation of agile principles and practices, and still receive a positive impact. (iii) It's feasible to implement agile principles and practices in large-scale software development projects. Agile principles and practices were not found to increase negative pressure and stress within the studied projects, although some agile practices were perceived as stressful.

TABLE IX THE MOST STRONGLY CONTRIBUTING STUDIES INCLUDED IN THE THEORETICAL FRAMEWORK

ID	Study reference
S1	Transition from a Plan-Driven Process to Scrum - A longitudinal Case Study on Software Quality. Li, Jingyue, Moe, Nils B and Dybå, Tore. Bolzano-Bozen, Italy : u.n., 2010. International Symposium on Empirical Software Engineering and Measurement.
S2	The impact of agile practices on communication in software development. Pikkarainen, Minna, o.a., o.a. 3, 2008, Empirical Software Engineerging, Vol. 13, ss. 303-337.
S3	Coordination in co-located agile software development projects. Strode, Diane E, o.a., o.a. 6, 2012, The Journal of Systems and Software, Vol. 85, ss. 1222-1238.
S4	Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Software Development Teams. Stettina, Christoph Johann and Hijstek, Werner. Pisa, Italy : u.n., 2011. Special Interest Group on Design of Communication.
S5	Escalation of Committ: A Longitudinal Case Study of Daily Meetings. Stray, Viktoria Gulliksen, Moe, Nils Brede and Dybå, Tore. 2012, Lecture Notes in Business Information Processing, Vol. 111, ss. 153-167.
S6	Workspace Environment for Collaboration in Small Software Development Organization. Mishra, Deepti and Mishra, Alok. 2008, Lecture Notes in Computer Science, Vol. 5220, ss. 196-203.
S7	Agile Practices: The Impact on Trust in Software Project Teams. McHugh, Orla, Conboy, Kieran and Lang, Michael. 3, 2012, IEEE Software, Vol. 29, ss. 71-76.
S8	The effect of moving from a plan-driven to an incremental software development approach with agile practices. Petersen, Kai and Wohlin, Claes. 6, 2010, Empirical Software Engineering, Vol. 15, ss. 654-693.
S9	Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Laanti, Maarit, Salo, Outi and Abrahamsson, Pekka. 2011, Information and Software Technology, Vol. 53, ss. 276-290.
S10	The Impact of Organization, Project and Governance Variables on Software Quality and Project Sucess. Abbas, Noura, Gravell, Andrew M and Wills, Gary B. 2010b. Agile Conference.
S11	Analyses of an agile methodology implementation. Ilieva, Sylvia, Ivanov, Penko and Stefanova, Eliza. 2004. Euromicro Conference.
S12	Exploring extreme programming in context: an industrial case study. Layman, Lucas, Williams, Laurie and Cunningham, Lynn. Salt Lake City : u.n., 2004. Agile Development Conference, 2004.
S13	Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. Lee, Gwanhoo and Xia, Weidong. 1, 2010, MIS Quarterly, Vol. 34, ss. 87-114.
S14	Rolling out Agile in a Large Enterprise. Benefield, Gabrielle. 2008. Hawaii International Conference on System Sciences.

B. Contributions to industry and research

Our study provides more empirical evidence on the impact of ASD in large-scale, industrial software development projects, as asked for by e.g. [1] [8]. We had access to two large-scale, industrial software development projects during half a year and have been able to show that the use of agile principles and practices has had a positive impact in both of them. This could serve as a positive case for other industrial organizations planning to implement the same ways of working. We have also shown that a survey is a good source of data in a case study.

C. Limitations of the study

We studied what impact the ways of working, namely an agile way of working, have within a software development project. It is unknown, but possible that other factors may have the same impact. The agile teams in Project A and all teams in Project B have implemented a wide range of agile practices. It was not always possible to separate which agile practice/practices have contributed to a certain effect. In many cases it is considered that the important contributor is the entire combination of agile practices.

As for all case studies, transferability of the results beyond the borders of the case is troublesome. In [16], Yin argues that transferability should be discussed from the theoretical foundations upon which the study was constructed. We gathered suggested effects from other industrial studies and believe that when we confirmed the same effects, other industrial organizations with similar conditions should be able to achieve the same effects. Nevertheless all expected effects weren't visible in the projects we studied and more industrial case studies in similar and different contexts are needed to fully understand how ASD works in large-scale, industrial software development projects.

VII. CONCLUSIONS

We have been able to perform a multiple-case study of two large-scale software development projects with a web-based survey responded by experienced informants as the major source of data. Since many factors were similar between the projects a comparative analysis was possible. The study at hand is one of few studies in a large-scale agile software development context.

VIII. REFERENCES

[1] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology,* vol. 50, pp. 833-859, 2008.

[2] J. Li, N. B. Moe and T. Dybå, "Transition from a Plan-Driven Process to Scrum - A longitudinal Case Study on Software Quality," in *International Symposium on Empirical Software Engineering and Measurement*, Bolzano-Bozen, Italy, 2010.

[3] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineerging,* vol. 13, no. 3, pp. 303-337, 2008.

[4] D. E. Strode, S. L. Huff, B. Hope and S. Link, "Coordination in co-located agile software development projects," *The Journal of Systems and Software,* vol. 85, no. 6, pp. 1222-1238, 2012.

[5] J. Sutherland, C. Ruseng Jakobsen and K. Johnson, "Scrum and CMMI Level 5: The Magic Potion for Code Warriors," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, Waikoloa, 2008.

[6] C. J. Stettina and W. Hijstek, "Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Software Development Teams," in *Special Interest Group on Design of Communication*, Pisa, Italy, 2011.

[7] V. G. Stray, N. B. Moe and T. Dybå, "Escalation of Commitment: A Longitudinal Case Study of Daily Meetings," *Lecture Notes in Business Information Processing,* vol. 111, pp. 153-167, 2012.

[8] P. H. Breivold, D. Sundmark, P. Wallin and S. Larsson, "What Does Research Say About Agile and Architecture?," in *Fifth International Conference on Software Engineering Advances*, 2010.

[9] S. L. Pfleeger and J. M. Atlee, Software Engineering Theory and Practice, Upper Saddle River, New Jersey: Pearson Education Inc., 2006.

[10] L. Lagerberg and T. Skude, "The impact of agile principles and practices in large-scale software development projects - a multiple-case study of two software development projects at Ericsson," Linköping University, Linköping, 2013.

[11] "The Agile Manifesto," 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 25 September 2012].

[12] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd ed., Boston: Addison-Wesley, 2005.

[13] "Guide to Agile Practices," Agile Alliance, 2011. [Online]. Available: http://guide.agilealliance.org/. [Accessed 25 September 2012].

[14] K. Schwaber, Agile Project Management with Scrum, Redmond, Washington: Microsoft Press, 2004.

[15] F. J. Abrantes and H. G. Travassos, "Common Agile Practices in Software Processes," in *International Symposium on Empirical Software Engineering and Measurement*, 2011.

[16] R. K. Yin, Case Study Research, vol. 5, Thousand Oaks: Sage Publications, Inc., 2009, pp. 3-5, 18, 132-133.

[17] J. W. Creswell, Qualitative Inquiry & Research Design, Thousand Oaks, California: Sage Publications Inc, 2007, pp. 40, 73-74.

[18] M. van Kooten, "Global Software Top 100 - Edition 2011," 23 August 2011. [Online]. Available: http://www.softwaretop100.org/global-software-top-100-edition-2011. [Accessed 19 December 2012].

[19] B. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[20] P. Lekvall and C. Wahlbin, Information för marknadsföringsbeslut, Göteborg: IHM Publishing, 2001.

[21] A. N. Manzo and J. M. Burke, "Increasing Response Rates in Web-based/Internet Surveys," in *Handbook of Survey Methodology for the Social Sciences*, New York, Springer Science+Business Media, 2012.

[22] D. Mishra and A. Mishra, "Workspace Environment for Collaboration in Small Software Development Organization," *Lecture Notes in Computer Science,* vol. 5220, pp. 196-203, 2008.

[23] O. McHugh, K. Conboy and M. Lang, "Agile Practices: The Impact on Trust in Software Project Teams," *IEEE Software,* vol. 29, no. 3, pp. 71-76, 2012.

[24] K. Petersen and C. Wohlin, "The effect of moving from a plan-driven to an incremental software development approach with agile practices," *Empirical Software Engineering,* vol. 15, no. 6, pp. 654-693, 2010.

[25] M. Laanti, O. Salo and P. Abrahamsson, "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation," *Information and Software Technology,* vol. 53, pp. 276-290, 2011.

[26] D. E. Strode, B. Hope, S. Huff and S. Link, "Coordination Effectiveness In An Agile Software Development Context," in *Pacific Asia Conference on Information Systems 2011 Proceedings*, 2011.

[27] G. Lee and W. Xia, "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility," *MIS Quarterly,* vol. 34, no. 1, pp. 87-114, 2010.

[28] L. Layman, L. Williams and L. Cunningham, "Exploring extreme programming in context: an industrial case study," in *Agile Development Conference, 2004*, Salt Lake City, 2004.

[29] S. Ilieva, P. Ivanov and E. Stefanova, "Analyses of an agile methodology implementation," in *Euromicro Conference*, 2004.

[30] N. Abbas, A. M. Gravell and G. B. Wills, "Using Factor Analysis to Generate Clusters of Agile Practices," in *Agile Conference*, 2010a.

[31] N. Abbas, A. M. Gravell and G. B. Wills, "The Impact of Organization, Project and Governance Variables on Software Quality and Project Sucess," in *Agile Conference*, 2010b.

[32] G. Benefield, "Rolling out Agile in a Large Enterprise," in *Hawaii International Conference on System Sciences*, 2008.