e-Informatica Software Engineering Journal, Volume 13, Issue 1, 2019, pages: 7–35, DOI 10.5277/e-Inf190101

Usage, Retention, and Abandonment of Agile Practices: A Survey and Interviews Results

Indira Nurdiani*, Jürgen Börstler**, Samuel Fricker***, Kai Petersen****

*Department of Software Engineering/DTU Compute – Software and Process Engineering Section, Blekinge Institute of Technology, Sweden/Technical University of Denmark, Denmark

**Department of Software Engineering, Blekinge Institute of Technology, Sweden

***Institute for Interactive Technologies, Fachhochschule Nordwestschweiz, Switzerland

****Chair for Software Engineering, University of Applied Sciences Flensburg, Germany

indira.nurdiani@bth.se/innu@dtu.dk, jurgen.borstler@bth.se, samuel.fricker@fhnw.ch, kai.petersen@hs-flensburg.de

Abstract

Background: A number of Agile maturity models (AMMs) have been proposed to guide software organizations in their adoption of Agile practices. Typically the AMMs suggest that higher maturity levels are reached by gradually adding more practices. However, recent research indicates that certain Agile practices, like test-driven development and continuous integration, are being abandoned. Little is known on the rationales for abandoning Agile practices.

Aim: We aim to identify which Agile practices are abandoned in industry, as well as the reasons for abandoning them.

Method: We conducted a web survey with 51 respondents and interviews with 11 industry practitioners with experience in Agile adoption to investigate why Agile practices are abandoned.

Results: Of the 17 Agile practices that were included in the survey, all have been abandoned at some point. Nevertheless, respondents who retained all practices as well as those who abandoned one or more practices, perceived their overall adoption of Agile practices as successful.

Conclusion: Going against the suggestions of the AMMs, i.e. abandoning Agile one or more practices, could still lead to successful outcomes. This finding indicates that introducing Agile practices gradually in a certain order, as the AMMs suggest, may not always be suitable in different contexts.

Keywords: Agile practices, Agile maturity models, survey

1. Introduction

The software industry is highly competitive. Agile methods, like Scrum and eXtreme Programming (XP), help to tackle the challenges of rapid changes in the environment of software organizations and help to reduce time to market, minimize development costs, and improve software quality [1]. Agile practices are the enactment of Agile principles [2].

A recent survey indicates that some practices like *test-driven development (TDD)*, *pair programming*, and *continuous integration* are being abandoned [3]. Abandoning Agile practices seems contradictory to common guidelines such as Agile maturity models (AMMs) [4–6] that prescribe which practices should be implemented and when according to certain maturity levels. According to the AMMs, the more mature an organization becomes, the more Agile practices are adopted. However, the indication of abandonment of practices could also be due to lack of guidance. Perhaps such practices were not introduced at the right time, given the maturity of the software development teams or organization, because Agile practices dependencies are not well known.

Submitted: 6 Feb 2018; Revised: 17 July 2018; Accepted: 17 July 2018; Available online: 30 September 2018

Table 1. Allocation of Agile practices to maturity levels in three AMMs

	Sidky et al. [6]	Patel & Ramachandran [5]	Nawrocki et al. [4]
Context	Agile practice adoption based on a measurement index	Agile practice adoption based on CMM(I)	Adoption of XP based on other maturity models
Level 1	On-site customer, collaborative planning, coding standard	—	—
Level 2	Tracking progress, continuous delivery	Tracking progress, on-site customer, planning game, TDD	Planning game, collaborating customer (on-site customer), user stories, metaphors
Level 3	F2F meeting, continuous integration, self-organizing team	Refactoring, pair programming, continuous integration, TDD, coding standard, collective ownership	Pair programming, coding standard, collective ownership, continuous integration
Level 4	Daily meeting (stand up meeting), user stories, frequent releases	Self organizing team, 40 h week	Simplicity (simple design), on-site customer
Level 5	TDD, pair programming	Focus on continuous improvement	—

Currently, we do not know why Agile practices are abandoned and how this impacts the overall success of Agile implementations. Without such information, we are unable to evaluate the suitability of AMMs in industry. As the first step towards evaluating the suitability of the AMMs is to better understand the usage and retention of Agile practices, and identify the rationales for abandoning Agile practices.

In this study, we aim to identify the rate of usage of Agile practices, their retention, and the rationales for their abandonment. To achieve our aim, we conducted a web survey and 11 interviews with industry practitioners with experience in Agile.

The remainder of the paper is structured as follows: Section 2 presents related work. Section 3 presents the research questions and survey design. Section 4 presents the results and analysis of the survey. Section 5 discusses the results and Section 6 summarizes and concludes the paper.

2. Background and related work

2.1. Background

According Schweigert et al. [7], there are approximately 40 AMMs proposed by academia and industry consultants. Many AMMs usually associate a number of Agile practices with a maturity level [7, 8]. Practices are introduced gradually. As a team or organization becomes more mature, more Agile practices are adopted [8]. An overview over three typical AMMs is provided in Table 1.

The idea of adding more Agile practices as a team or organizations becomes more mature seems contradictory to current empirical studies that show that Agile practices like *TDD*, *pair-programming*, and *continuous integration* are abandoned [3]. This raises a question regarding the suitability of AMMs for industry, particularly when the AMMs do not provide rationales for the mapping of Agile practices to maturity models. Critics of the AMMs indicate that the AMMs are not fit for industry use [9] and that their recommendations are contradictory [8, 10]. In this study, we aim to evaluate the suitability of AMMs by investigating the usage and abandonment of Agile practices in industry through a survey and a series of interviews.

2.2. Related work

Kurapati et al. [16] performed a survey to identify commonly used Agile practices at project and organization levels. Their results show that the most commonly used practices both at project

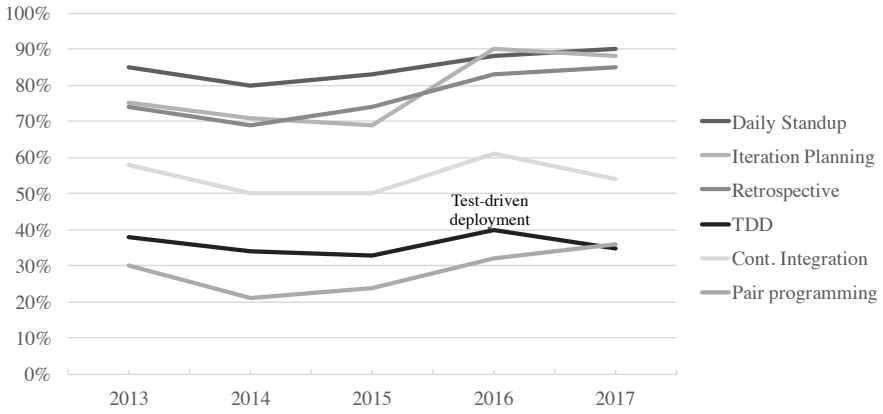


Figure 1. Fluctuation of Agile practice usage from Version One's State of Agile 2013–2017 [11–15]. The practices shown in the figure are for exemplification and the ones consistently reported across the annual surveys

and organization levels include *stand-up meeting, sprint and iteration, collective ownership*, and *tracking progress*. Less common practices both at project and organization levels include *simple design, TDD, pair-programming*, and *planning game*. One practice that is rarely practiced both at project and organization levels is *metaphor*. It is also interesting to highlight that the use of *metaphors* reported by Kurapati et al. turns out differently from Murphy et al.'s [17].

Kropp et al. [18] conducted a survey as part of Swiss Agile Study 2014. They distinguished three types of practices: technical, collaborative, and advanced practices. Technical practices include *refactoring*, *TDD*, and *coding standards*. Collaborative practices include *on-site customer, daily stand-up*, and *pair programming*. Advanced practices are *kanban pull-system, acceptance TDD*, and *Behaviour Driven Development (BDD)*. Their results show that more experienced practitioners implement considerably more practices compared to less experienced ones. Furthermore, less experienced practitioners implement primarily technical practices, meanwhile more experienced ones implement more collaborative practices. It is worth noting in this study *metaphors* is not included in the survey, unlike the previous survey by Kurapati et al. [16].

The two surveys described above, i.e. [16] and [18], report the results of Agile practice usage from one single calendar year. They do not capture whether the practices are continuously used or not.

Murphy et al. [17] reported results of five annual surveys internal to Microsoft over the course of six years. Their results show that practices like *code reviews*, *metaphors*, and *retrospective* are increasing in their adoption. Meanwhile, certain practices like *unit testing*, *TDD* and *pair programming* are decreasing in their adoption [17, Figure 4, p. 79].

VersionOne also conducts annual state of Agile surveys. We took the results from the annual survey over the past five years (2013 – 2017) and created a figure that presents the trend of the usage of some Agile practice [11–15] in Figure 1. The results of the annual surveys indicate that the use of Agile practices is fluctuating over the past five years, see Figure 1.

The surveys reported by Murphy et al. [17] and Version One [11–15] capture the increase and decrease of Agile practices usage over the years. However, the increase of some Agile practices from one year to the next does not indicate that those practices are being added, as suggested by AMMs. The decrease of some Agile practices does not indicate that those practices are being abandoned. It is possible that the respondents of the surveys from one year to the next are different. In the case of Murphy et al. [17] respondents who participated in one survey were not allowed to participate in the next survey. These surveys do not reflect the use of Agile practices in one context/team over time. Thus, the results cannot be used to assess the suitability of AMMs.

Solinski and Petersen [3] surveyed Agile practice adoption scenarios over time as practitioners transition from plan-driven development towards Agile. The survey identified Agile adoption scenarios which include an incremental adoption of practices, big-bang adoption – where plan driven practices are discarded and replaced by Agile practices, and complex tailored adoption processes. Their results also revealed that practices like *TDD* and *continuous integration* are being abandoned. However, their study did not focus on rationales for abandoning practices.

Indications of Agile practice abandonment is also reported by Ralph and Shportun [19]. Their case study revealed the abandonment of Scrum in distributed teams. One of the main factors associated with abandoning Scrum is the degradation of Scrum practices. Three Scrum practices that were difficult to implement due to distribution are *daily stand-up meeting, tracking progress using burn-down chart*, and *fixing sprint backlog*.

To summarize, current research indicates that some Agile practices are abandoned. However, current surveys have not yet focused on the rationales for abandoning Agile practices, or the time-frames from practice adoption to abandonment. Currently, we do not know how abandoning practices may influence the perceived overall success of implementing Agile methods. In this paper, we investigate why Agile practices are abandoned and whether or how this influences perceived success.

3. Research methodology

In this study, we aim to identify which Agile practices are being used and abandoned in the industry and the rationales for abandoning a practice to better understand practice adoption and the relevance of Agile maturity models.

RQ1. What is the rate of usage of Agile practices?

RQ2. Which Agile practices have been abandoned?

- RQ2.1. How long are practices in use before they are abandoned?
- RQ2.2. What are the rationales for abandoning these practices?

RQ3: What is the perceived success rate of Agile practices implementation?

- RQ3.1. Does the perceived success rate differ between respondents who retain practices versus respondents who abandon practices?
- RQ3.2. What are the used measures of success?

By "use" or "usage", we mean that an Agile practice is used or was in use at some point in time, while "abandoned" means that an Agile practice was used in the past, but is no longer used. To answer the research questions above, we conducted a survey and a series of follow-up interviews.

3.1. Survey

3.1.1. Sampling strategy

We distributed the survey to personal contacts and well-established professional groups in Agile software development on LinkedIn and Google Groups, i.e. convenience sampling. Distributing surveys over professional groups is a known way to distribute surveys as reported in [3, 16]. When using convenience sampling, which is a common strategy in software engineering surveys, it is important to describe the sample [20]. Following the guidelines from Linåker et al. [20], we define our sample as follows:

- Target audience: software industry practitioners who have experience in Agile practices adoption. Particularly, those who have experience in observing or experiencing when a practice is adopted and/or abandoned. In this survey, all practitioners from different industry domains, organization size and different levels of experienced are welcome to participate. However, this does not necessarily mean all responses will be considered (see Data Screening in Section 3.3).
- Unit of analysis: Agile practices which have been adopted and abandoned, their rationales, and perceived success rates.
- Source of sampling: professional groups or communities focused on Agile software devel-

opment. Personal contacts who are known to work with Agile software development.

3.1.2. Survey design

We followed the recommendations from Robson [21] in designing a self-administered web-based survey. The survey was developed using the tool SoSci Survey (https://www.soscisurvey.de).

We included interactive sliders as a visual aid to allow respondents to indicate the start and/or end of Agile practice usage, see Figure 2. The survey design is adapted from Solinski and Petersen [3], who also investigated time-frames of Agile practice usage.

Similar to past surveys, we included a selection of Agile practice. However, there is no commonly agreed set of Agile practices. Different surveys include different sets of Agile practices. For example, Rodriguez et al. [22] include 16 practices; Kurapati et al. [16] include 25 practices. In this survey, we adopted the list used by Solinski and Petersen [3], which includes 7 plan-driven practices and 14 Agile practices. We chose this list because their survey is quite recent and comprises a manageable number of practices. In their survey, Solinski and Petersen [3], merged some practices, such as *short iterations and frequent releases*. We also merged two practices, if the practices are closely related. To see if two practices are related, we cross referenced the definitions of Agile practices described by Petersen [23] and Williams [2]. However, we separated Solinski and Petersen's combined practice "technical excellence" into its original sub-practices *refactoring*, *simple design*, and *coding standards*.

At the beginning of the survey, we briefly described the aim of the survey to the respondent. To avoid bias, we did not mention that we are looking for practices which had been abandoned. We described that we are interested in understanding the order in introducing Agile practices. The survey itself comprises five main parts. The detailed survey questions are available in Appendix B.

Part 1A. Agile practice adoption (RQ1). Respondents could indicate practice usage as "used", "never used", or "don't know". See Figure 2 Part 1A (to the left). Definitions of practices are available by hovering the mouse over the information icon. The practices included in the survey and their definitions can be seen in Appendix A. In this survey, we did not inquire which Agile framework, e.g., Scrum, eXtreme Programming (XP), etc. was used. This was done to avoid confusion from the respondents because it is possible that practitioners combine practices from different frameworks or on occasions also include plan-driven or waterfall practices [3, 24].

Part 1B. Start and end of Agile practice (RQ2 and RQ2.1). Using interactive sliders, respondents could indicate the start- and stop-time for when a practice was in use as shown in Figure 2 part 1B. The time-frame for the sliders is between < 2006 and "Still in Use". When respondents indicated "never used" or "don't know" in Part 1A, the sliders are disabled. We used the interactive sliders to identify abandoned practices, so we did not bias respondents by explicitly asking for abandoned practices. Respondents could also leave optional comments or additional information regarding a practice.

Part 2. Perception and measures of success (RQ3). From Part 1B, we would be able to see which Agile practices were used, retained, and abandoned. The usage, retention, and abandonment of Agile practices represent a strategy for *Agile practice adoption*. We inquired the impacts of Agile practice adoption, as described in Part 1B, in terms of perceived success rate. *Success rate* is respondents' perceived degree of success of Agile practice adoption on their projects or teams. A Likert-type scale was used to indicate success rate, from very unsuccessful (1) to very successful (5). Respondents could also answer "don't know". Furthermore, we asked respondents to indicate how success was measured. We believe it is important to inquire what measures are used to indicate success, because different practitioners from different contexts may have different perceptions of success.

Part 3. Limitations and rationales (RQ2.2). We asked which challenges and limitations respondents experienced during Agile practice adoption according to Part 1B and, in particular, why practices were discontinued (if any).

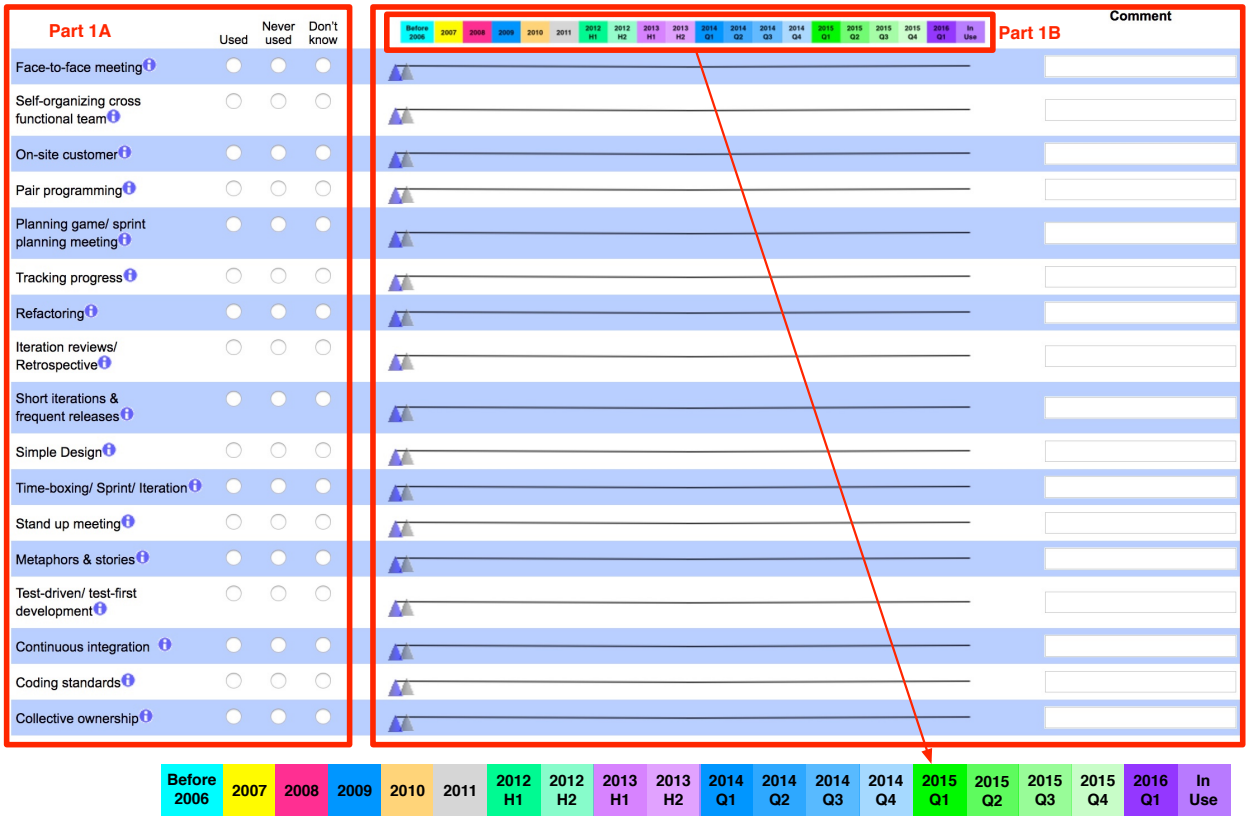


Figure 2. Interactive sliders

Part 4. Contexts. We asked respondents to provide information about their personal background and organizational context: (1) their role(s), (2) years of experience, (3) number of team members involved in software development, (4) team-setting (collocated or distributed), (5) how Agile practice adoption was decided (team-level or company), (6) industry domain(s), and (7) type(s) of software systems being developed (classification is adopted from [25]).

Part 5. Contact. We also asked the respondents to provide their names and email addresses, for follow-up interviews or to receive a copy of the survey results.

3.1.3. Survey pilot and execution

The sliders made the survey more complex and increased the risk that questions are not well understood. To mitigate these risks, we piloted the survey with five colleagues of the authors and five industry practitioners with experience in Agile software development.

Regarding the pilot, some industry practitioners felt that the definitions of some Agile practices were too specific and might not be applicable in their contexts. To address this issue, we reformulated the definitions. Two pilot respondents had difficulties to move the sliders. We resolved this problem by adding instructions on how to use the sliders. After addressing the feedback from the pilot, we deployed the survey, which was open between March–July 2016.

3.2. Interviews

3.2.1. Interviewees recruitment

Interviewees were recruited from the survey respondents who left contact information for further inquiries. Twelve invitations were sent out, and three confirmed for follow up interviews. We then recruited eight additional interviewees through personal industry contact and referrals. For the new recruits, we also asked them to fill in the survey prior to the interview to maintain

consistency and helped us to formulate interview questions.

In total, we had 11 interviewees. Our interviewees represent a wide variety of contexts. They came from various industry domains and geographic locations. More details on the interviewees can be found in Table 2.

3.2.2. Interview design

The goal of the interviews was to gather richer and better contextual information about the use and abandonment of Agile practices. In the interviews, we used semi-structured interviews. The interviews were done face-to-face whenever possible. Otherwise, the interviews were done over the telephone or video call. Prior to the interview, we sent each interviewee a summary of their answers from the survey. Each interview lasted 45–60 minutes and was recorded and transcribed. In the interviews, we inquired the following:

- Interviewee's roles and responsibilities, short description of the product being developed;
- The interviewee's survey answers were revisited and further clarified:
  - Why did you mark (enumerate Agile practice marked as "never used") as never used? (RQ1),
  - Why did you mark (enumerate Agile practice marked as "don't know") as don't know? (RQ1),
  - Could you please elaborate the reasons for abandoning (enumerate Agile practices which were no longer used from Part 1B)? (RQ2.2);
- Wrap-up. Inquire the interviewee's impressions on the interview.

3.3. Data analysis

Data screening. Prior to the analysis of the survey data, we carefully scrutinize each dataset to ensure their reliability. We checked each respondent's answers to each question. For example, we cross referenced the participants' experience (in years) and the time frame indicated in the sliders. We also checked the respondents' answers to the open-ended questions. We excluded a response if a respondent did not provide a comprehensible answer to one of the open-ended questions. We also excluded a response if a respondent indicated that most or all of the practices had been abandoned and did not specify that it was past experience.

Rate of Agile practice usage (RQ1). We used descriptive statistics to analyze the rate of Agile practice usage, i.e. practices that are marked as "used" by the survey respondents.

Agile practice abandonment (RQ2). For all practices that were indicated as "Used" (in Part 1A of the survey), we checked the slider position for "practice end". If this position did not indicate "In Use", we considered the practice as abandoned and calculated the timespan of use by means of the slider positions for start and end of use, respectively. We also calculated the *abandonment ratio* for each Agile practice to calculate the proportion of the number of times a practice is abandoned to the number of times a practice is used. To answer RQ2, we also included the results from the interviews. To analyze the interview transcripts, we used f4analyse tool (https://www.audiotranskription.de/english/f4-analyse) to help with coding steps. First, we performed line-by-line coding as an approach to open coding [26] on the interview transcripts. Open coding was followed by focused coding to identify common themes from the data. The result of focused coding can be seen, for example, in Table 5. The coding process was primarily done by the first author. To minimize bias, another co-author conducted post-hoc validation on the coding done by the first author.

Success rates and measures (RQ3). We analyzed the success rates of adopting Agile practices across domains, and retain vs. abandoned. We also used descriptive statistics to analyze the success rates. To cross-tabulate the industry domain and the success rates, we used the "Crosstab" feature in SPSS. To identify the measures of success from the survey, we employed qualitative coding similar to the one used for analyzing the interview data. First, we tabulated all responses to each relevant question using a spreadsheet and f4analyse tool. We then used open coding [26] to assign codes to text fragments. For example,

Table 2. List of interviewees and their contexts

ID[a]	Location	Role	Experience	Team size[b]	Market	Domain	Context overview
R11	Indonesia	Project Manager	6 years	100	Market driven, internal use	Insurance	IT Department of a multinational Fortune 500 company. Adopted 13 practices except pair-programming, TDD, and metaphors & user stories.
R14	Brazil	Developer, Trainer, System architect	3 years	20	Internal use	Government (Court)	IT Department from the Brazilian court of accounts. Adopted 15 practices except pair programming and retrospective. Abandoned on-site customer and tracking progress.
R32	Canada	Developer, Quality Assurance, System Analyst	6 years	13	Market driven, bespoke	Independent Software Vendor (ISV)	Start-up company initiated in 2012. Adopted 14 practices except for on-site customer, simple design, and TDD. Abandoned pair programming and tracking progress.
R33	Sweden	Scrum Master	6 years	6	Internal use	Telecoms	A small project team within a large multinational company. Adopted 15 practices except on-site customer and TDD. Abandoned 13 practices except face-to-face meeting and stand up meeting.
R34	Indonesia	CEO	3 years	33	Bespoke	ISV	Start-up company initiated in 2014. Adopted 14 practices except TDD, collective ownership, and metaphors & user stories.
R35	Ireland	Scrum Master, Developer	3 years	6	Bespoke, market driven, maintenance	ISV	Start-up company initiated in 2012. Adopted 14 practices, except TDD, coding standard, and simple design.
R36	Sweden	Program Manager	23 years	1000+	Market driven	Telecoms	A solution development program in a large multinational company. Adopted 16 practices except TDD.
R37	Sweden	Scrum Master	20 years	1000+	Market driven	Telecoms	A solution development program in a large multinational company. Adopted 16 practices except TDD.
R38	Sweden	Scrum Master, QA	7 years	70	Market driven	ISV	A project in a large multinational company. Adopted 15 practices except on-site customer and simple design.
R39	USA	Researcher, Developer	3 years	6	Bespoke, market driven	Research & development, biomedical	A project in a university to develop biomedical research support tool. Adopted 12 practices except pair programming, tracking progress, stand up meeting, metaphors & user stories, and TDD.
R40	Finland	CTO, Developer, Scrum Master	6 years	11	Market driven	ISV	A start-up company initiated in 2012. Adopted 10 practices, except on-site customer, planning game, refactoring, retrospective, metaphors, TDD, and collective ownership.

[a] Respondent ID according to the order they are received in the survey tool.
[b] Reflects the size of software development team affected by the Agile implementation. Not overall company size.

for the following response regarding used success measures: "Success can be measured by completion of tasks on time with high quality and without any blockers", we assigned two codes: *time to deliver* and *product quality* as measures of success. The measures of success were then classified into product, process and resource measures according to Fenton and Bieman [27].

3.4. Validity threats

In reporting the validity threats, we follow the classifications suggested by Petersen and Gencel [28].

Theoretical validity. It refers to pertains to the issue of capturing the construct intended to be collected. Both the survey and the interviews, are retrospective. The respondents may not remember precisely when an Agile practice was introduced. To minimize the issue, we did not inquire exact months or dates for the start or the end of a practice. We only refer to the year, half a year, or quarters. The slider design does not support exact dates and only one start and one end time. To minimize the issue we added comment text boxes next to the sliders to supply details. Maturation could pose as a threat if the survey takes too much time to complete. To minimize maturation, we minimized the number of included Agile practice, i.e. 17 practices. It is possible that we missed one or more Agile practices. To reduce maturation, we merged practices that are similar in their definitions, as described in [23]. It is also possible that merging some of the practices caused confusions to the respondents. In this survey, we also provided definitions of the Agile practices primarily from the literature, e.g., [2, 23]. It is known that how Agile practices are implemented in the industry may differ from their definitions in the literature [2]. This may lead to respondents answering "don't know" or "never used", when the practices are actually in use. These issued are partially mitigated by piloting the survey and performing follow up interviews with 11 of the survey respondents. Another concern pertaining theoretical validity is with the sampling. In this survey, we used convenience sampling by recruiting participants from professional groups and personal contacts. The former may lead to reliability issue, while the latter may lead to bias. To minimize reliability issue, we checked each response to ensure coherence (see *Data Screening* Section 3.3). For example, if a respondent indicated to have 1–3 years of experience, but used the sliders indicating a period longer than that, we deemed the answer to be invalid. To minimize bias from the personal contacts, as well as the other respondents, we did not specify that we aim to collect Agile practices that are being abandoned. It is also important to clarify that these personal contacts were not individuals whom the authors had prior close collaborations. Thus, they were never given information about the plan of the study.

Descriptive validity. It concerns with the accuracy of capturing the reality. In this study data collection was done through a survey and interviews. As researchers, we cannot observe the reality, and the responses we obtained are based on the respondents' perception. For example, a respondent's experience can influence his/her answers; a new hire may not be aware that a practice was used before but has been abandoned. It is also possible, that a respondent perceives a practice was used because he/she used it, but it was not institutionalized in the team or project. The follow-up interviews helped to capture better information that was otherwise missing from the survey. However, in survey and interviews studies, such a threat cannot be fully eliminated, since no actual observation was done. Although we were not able to eliminate the issue, it is important that we acknowledge it. In this survey we provided instructions for the respondents to reflect on an experience that they were most familiar with, it could be an experience from a specific team or a specific project. The experience could also be from present or past experience. It is possible that a respondent reflected on past experience, and indicated all practices had been abandoned. For such a case, unless the respondent wrote a note that it was past experience, we deemed the answer to be invalid.

To improve thoroughness and trustworthiness of the survey, we reported as many details as possible regarding the design and execution of

the survey, following the criteria described by Stavru [29]. A self-assessment on the thoroughness of our survey using Stavru's criteria and calculation procedure resulted in a score of 0.8 on a scale 0–1 (see Table C.1 in Appendix C for details). Stavru does not provide interpretation of the scale. However, our score is higher than other Agile surveys examined by Stavru in [29], where the highest score was 0.64. This indicates that we have provided sufficient information to demonstrate the thoroughness of our survey [29].

Interpretative validity. It concerns with researchers' bias in drawing a conclusion. This study primarily relies on qualitative data collected from a survey and from interviews. Researchers bias can affect the conclusions that are drawn. In analyzing the data, the first author was responsible for the qualitative coding. To reduce bias, another co-author validated the coding post-hoc after the first five interviews, to see if there could be disagreements in the codes.

Generalizability. It refers to the extent that the results of the study are generalizable to a larger population. In this study, both for the survey and the interviews, we used convenience sampling. The selection of the respondents was non-purposeful and based on willingness. Respondents have various roles and tasks in different organizational contexts. However, some roles such as consultant and C-level managers are under-represented. Furthermore, most of the respondents work in small organizations. Although we did not collect company name and geographical location of the respondents, we could ascertain that our sample represents 20 unique organization from 11 different countries. Although some countries like the Canada, Italy, and New Zealand are under-represented, our sample represents different geographical locations. In this survey, we also small sample size. We cannot claim that our results are generalizable to a large population or in anyway represents the current state of Agile practice. However, the demographics of our respondents include a large variety of contexts that adds to the richness of the data and minimizes the risk of confounding factors that could be present due to a homogeneous context.

4. Results and analysis

In total, 200 people started the survey, 70 completed the survey but only 43 answers were valid, i.e. consistently answered part 1 – 4 of the survey. Out of 43 respondents, 32 of them completed part 1A and used the sliders from part 1B of the survey. The remaining 11 respondents did not use the sliders (part 1B). Including the new interviewee recruits, in total, we have 51 respondents and 40 of them used the sliders. From 40 respondents who used the sliders, 22 retained all practices that were used. Meanwhile 18 abandoned one or more practices.

Out of 51 respondents, 10 participated through direct invitations, and 3 participated through referrals. In the survey, we did not inquire company name and location where the respondents were or had been employed. Based on direct invitations, referrals, and a number of respondents who provided their work emails, we could ascertain 20 unique companies from 22 respondents. We could also ascertain the geographic location of 19 respondents; they were from Sweden (5), Ireland (3), US (2), Indonesia (2), Canada, New Zealand, Finland, Portugal, Brazil, Germany, and Italy (1 of each).

The 51 respondents were primarily developers (20; 39.2%) followed by Scrum Masters (15; 29.4%) and quality assurance specialists (13; 25.5%). Please note that multiple roles could be selected. Further roles are system architect and department head (8; 15.7% for each), project manager and department head (each 7; 13.7%), business analyst, system analyst trainer, product owner, C-level managers (e.g., Chief Executive Officer, Chief Technical Officer, etc.), and other roles (<6; <10%).

Regarding their level of experience in software development, 14 (27.4%) respondents had more than 6 years of experience, 15 (29.4%) had 3–6 years of experience, 15 (29.4%) had 1–3 years of experience, and 7 (13.7%) had less than one year of experience. Most of the respondents (21; 52.5%) were part of a small organization with less than 50 people involved with software development. Eleven (27.5%) were part of organizations

with 50–249 people, and 7 (17.5%) were part of organizations with 250–4999 people.

In terms of distribution, 28 (54.9%) of the respondents mentioned that their Agile software development teams were collocated; 19 (45%) of them worked in a single team and nine (17.6%) in multiple teams. The remaining 23 (45.1%) stated that their Agile software development teams were distributed; 10 (19.6%) of them worked in a single distributed team and 13 (25.5%) in multiple distributed teams.

Regarding application domains (multiple selections possible), most of the respondents were from independent software vendors (17; 33.3%), followed by financial services (15; 20%). Respondents also came from the following domains: research and development (11; 21.6%), telecoms (12; 23.5%), medical (8; 15.7%), media and entertainment (4; 7.8%), government (3; 5.9%), and manufacturing (1; 2%). For the types of software systems that respondents develop and type of market, please refer to Figure 3a and 3b.

To complement the survey, we also conducted 11 interviews with industry practitioners. The list of interviewees and their respective contexts are presented in Table 2.

4.1. Usage of Agile practices (RQ1)

Figure 4 shows the rate of Agile practice usage. From Figure 4, we can see that out of 51 respondents, *face-to-face meeting* was the most commonly used Agile practice among our respondents (48 respondents), followed by *tracking progress* (47 respondents). Other commonly used Agile practices by our respondents were: *self-organizing team*, *planning game*, and *retrospective*. Practices like *TDD* (27 respondents) and *pair-programming* (28 respondents) were less commonly used by our respondents.

The follow up interviews identified Agile practices that were not included in the survey, they are: (1) Behavior-driven development/BDD (R35), (2) Scrum of scrums (R38). R32 mentioned that in addition to *retrospective* at the end of a sprint, they also do a project level retrospective which was done every two months.

The follow up interviews also revealed that some respondents interpreted the definition of Agile practices slightly different to our definitions. R35 and R36 indicated in the survey that *on-site customer* was used. However, in the follow up interviews, they clarified that they did not actually have customers present on their premises. Rather they have a dedicated team member who acted as a proxy to the customers, i.e. *product owner*.

The rationales for never using certain Agile practices are summarized in Table 3. From the interviews, we identified that respondents R11, R14, R32, R39, and R40, marked some of the practices as "never used" or marked as "don't know" because they were not adopted according to our provided definitions or were not adopted consistently. For example, when inquired why *stand up meeting* was never used respondent R39 mentioned that " because of the word daily in the definition, we do not do daily meeting". Meanwhile respondent R11 mentioned the reason for marking "do not know' for *collective ownership* is because the project involved outsourced developers and the level of collective ownership varies from the internal team to the outsource team: "internal [team] is not a problem, but the outsource team has no collective ownership". This indicates that the usage of Agile practices is not binary (used or not). Often Agile practices are modified from how they are defined or implemented inconsistently.

From Table 3, we can see that some practices are not suitable in certain contexts. Some practices may not be applicable given certain contextual factors like regulation, team/organization's culture, and organization set-up. The characteristics of the software system, e.g., legacy code and product complexity, could also make some Agile practices unsuitable.

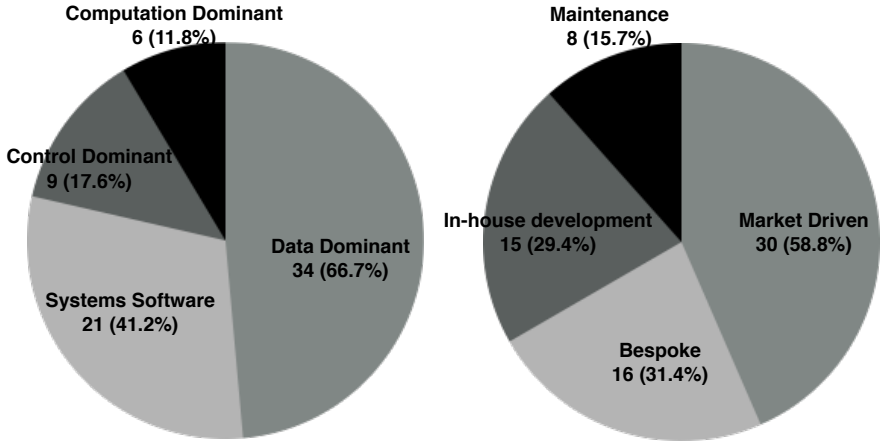
4.2. Abandonment of Agile practices (RQ2)

As mentioned earlier, 18 of the respondents abandoned one or more Agile practices. Each respondent abandoned at least one Agile practice. One of the 18 respondents abandoned up to 13 Agile

Table 3. Rationales for never using certain Agile practices and the supporting quotes from the respondents

Rationale	Practice	Quotes (with Respondent's ID)
Incompatibility with the domain or market of development	Short iteration	Release of each sprint to end customer is not possible in case of regulatory development (R20).[a]
	On-site customer	Our customers are 100M people (R13).[a]
		We are product company, it is a [software as a service] product over the Internet (R40).
		Some of our customers are not even in the province (R32).
Challenges in implementing a practice	User stories	The biggest challenge was conforming to the structure of developing user stories (R19).[a]
	TDD	We do not have the patience to follow through with it. It is quite challenging with a big ecosystem [of 26 products] like this (R36).
	Metaphors	We use user stories but not metaphors, metaphors are too obscure for most people to grasp (R35).
Product complexity	Simple design	The product we were working on was extremely complex, we had a lot moving pieces and that was an unavoidable complexity the domain was complex [...] the hardware aspect definitely have to do with it, hardware and firmware development (R32).
	TDD	Our product is very explorative, [we are] creating new software, we rather implement TDD next time (R40).
Legacy code	Simple design	We are left with a mess from the previous development team. We are adding and maintaining the legacy we are left with to get the product to the market (R35).
	TDD	We have a lot of [legacy] in our code, [it was not easy] for us just to jump into [TDD] [because] the old code was not done in that way (R37).
Organization set up	On-site customer	we never interact with customers because were in the R&D department, the department that interacts with the customers is called customer unit (R33).
Lack of resources	Collective ownership	We have a massive product and too few people, collective ownership is not possible, we need specialists (R40).
Lack of management involvement or enforcement	Retrospective	Most of time management would trust the team to work, they [would not] be picky and asking people to do retrospective and that kind of thing (R14).
	TDD	I [do not] know why we [do not] use TDD, We at [the company] just never use TDD (R33).
	PP	Management [did not] talk about it at all. I [do not] think we ever discussed whether to use pair programming or not (R14).
Lack of perceived value	Planning game	There is no need for a planning game because each developer is responsible for a component of a feature. I [do not] think planning game helps in this case. Just keep releases small and often (R40).
	Refactoring	It [does not] make sense to refactor because the components that you refactor would be obsolete anyway in a very short time (R40).
Conflict with team's culture	Retrospective	We want to foster the kind of culture where you are not keeping something for a [sprint]. You just bring it up immediately (R40).

[a] Respondent provided answer through the survey.



(a) Type of systems (b) Type of markets

Figure 3. Respondents' type of system and type of markets (multiple selection possible)

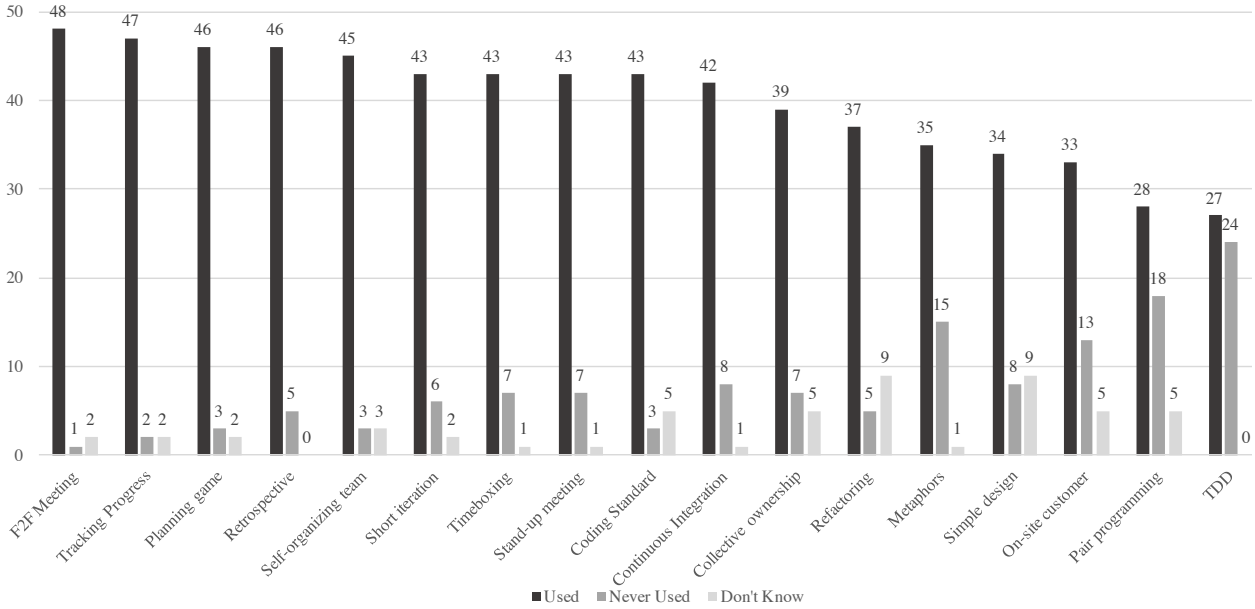


Figure 4. Adoption of Agile practices

practices. All 17 Agile practices included in the survey were abandoned at some point. From Table 4, we can see that *face-to-face meeting* has the lowest abandonment ratio (0.05). Meanwhile, *Tracking progress* has the highest abandonment ration (0.29) followed by *planning game* (0.2). This finding may indicate that certain practices, like *face-to-face meeting*, are more easily retained than others. Meanwhile, a practice like *TDD* may not be as popular, but once it was adopted, it is more likely to be retained, as we can see the abandonment ratio is quite low (0.11).

From Table 4, the number of respondents who abandoned individual Agile practices is relatively low when compared to the number of respondents who retained the practices. This shows that most of the time each Agile practice is still in use.

4.2.1. Usage until abandonment (RQ2.1)

Table 4 summarizes the periods of time that an Agile practice was in use. Practices are most often abandoned within the first half year after their introduction. After 3 years of use, the rate

Table 4. Agile practices that have been abandoned and how long they had been in use before abandonment

Practices	≤6 mon	≤12 mon	≤ 24 mon	≤36 mon	≤48 mon	60+ mon	Abandon	Still in use	Total usage[c]	Abandon ratio[d]
Tracking progress	3	2	3	1	1[a]	1	11	26	37	**0.29**
Planning game	3	1	1	1		1[a,b]	7	28	35	0.2
Retrospective	2	1	2	1			6	30	36	0.17
Time-boxing	2	1	1	1			5	30	35	0.14
Collective ownership	2		1	1		1[a,b]	5	25	30	0.17
Self organizing team	1	1	1	1			4	34	38	0.11
Pair programming	2		1	1			4	17	21	0.19
Simple design		2		2			4	22	26	0.15
Stand up meeting		2	1				3	32	35	0.08
Refactoring		2		1			3	26	29	0.1
Short iteration	2			1			3	29	32	0.09
Metaphors and stories	1	1		1			3	24	27	0.11
Continuous Integration			1	2			3	30	33	0.09
TDD	1	1					2	15	17	0.11
F2F Meeting	1				1		2	37	39	**0.05**
On-site customer	2						2	22	24	0.08
Coding standard	1			1			2	32	33	0.06
	23	13	12	15	2	2				

[a] Respondents in financial domains. [b] Response from the same respondent.
[c] Total usage based on 40 respondents who used the sliders. [d] Ratio = abandon/total usage.

of abandonment drops significantly. Only *tracking progress, planning game, collective ownership* and *face-to-face meeting* were abandoned after having been in use for more than 3 years.

This finding may indicate that in some contexts, certain practices are not suitable to be introduced in the first place, or introduced in the wrong order due to dependencies on other Agile practices. Also, as the findings from subsection 4.1 shows that Agile practices may be modified or implemented inconsistently, it is possible that the modifications, or the lack thereof, has undesired side effects that may present themselves at various time periods. The rationales for abandoning Agile practices are presented in the following subsection.

4.2.2. Rationales for abandonment (RQ2.2)

Eight respondents provided rationales for abandoning the following practices: *pair programming, tracking progress*, and *on-site customer*. Meanwhile, two respondents, R28 and R33, abandoned 5 and 13 practices respectively. They did not provide a rationale for each practice. Instead, they provide a common rationale for abandoning a group/set of Agile practices (indicated as Not specific in Table 5). Most rationales were obtained for *tracking progress*. Table 5 summarizes the rationales for abandoning Agile practices.

The statements from R14 and R38 in the discontinuation of *tracking progress* indicate that Agile practices dependencies are not well understood. In the case of R14, *tracking progress* was introduced before sprint planning was established. Because sprint planning was not done, new tasks could be added throughout the week, and tracking progress became ineffective, as respondents R14 explained: "It seems like we were walking backwards. We were working towards the end of the week, and things just got worse. Because somebody would suddenly add a workload to the sprint." Meanwhile, in the case of R38, tracking progress was introduced before the team members develop better product knowledge. This shows that there could be prerequisites before introducing certain Agile practice. The prerequisites could be other Agile practices or acquiring product or project-related knowledge.

From Table 5, one of the more interesting rationale for abandoning one or more Agile practices is the influence of a person, as reported by

Table 5. Rationales to abandon Agile practices and the supporting quotes from the respondents

Rationale	Practice	Quote (with Respondent's ID)
Poor estimation and team dependency	Tracking progress	Due to bad estimation and dependency on other teams we are unable to track progress by burn-down chart (R32).[a]
Lack of product knowledge	Tracking progress	[The team members] complain that we [do not] have the product knowledge, how do we estimate it if we [do not] know the complete technicalities (R38).
Team member discomfort	Pair programming	People were uncomfortable and people did not really want to engage in that (R32).
Lack of engagement	Tracking progress	Half of the were tracking progress and they other half [were not], management did not really care (R14).
Conflict with other Agile values	Pair programming	The idea of sustainable pace, [. . .] we are only expected to be at the office at certain core hours [. . .]. I would be one of the people showing up around 9.30-16.30 [. . .]. so if I want to pair program with one of the latecomers, it would only really work from 13–15 (R32).
Influence of a person	Not specific	It was because one person was quite very opinionated, the person thought why do all these things, it's a waste of time (R33).[b]
	On-site customer	The guy [who initiated on-site customers] went on vacation and he did not come back (R14).
	Tracking progress	The new product owner did not want/care for [statistics], and the team did not demand them (R32).
Lack of perceived values	Tracking progress	As we do product development of a rather mature product, the tracking of progress was not all that valuable. Stuff at the top of the backlog has most value. Stuff lower has a lower value, and will be released later. No real forecast of this was needed (R21).[a]
	Tracking progress	We just try to push things to production all the time (R40).
	Tracking progress	The team did not feel the need for it (R30).[a]
	Not specific	The part that can be handled by Agile is finished. Other part cannot use Agile (R28).[a,c]
Dependency on other practice	Tracking progress	We tried to do tracking progress but sprint planning was not done [yet] (R14).

[a] Respondent provided answer through the survey. [b] 13 out of 15 Agile practices were abandoned.
[c] 5 out of 12 Agile practices were abandoned.

R14, R32, and R33. Respondent R14 mentioned that *on-site customer* was adopted for only two months because the person in charge had to leave the company. This individual was crucial to make *on-site customer* worked smoothly because the person can bridge between the technical team and the end users (court officers): "He was both an engineer and a lawyer. So he could very easily talk to the business people and to us". Meanwhile, R33 indicated that the practices were adopted for up to three years until they are abandoned: "They've been practicing Scrum since 2012. Suddenly in 2015, they stopped completely [. . .]. They just dropped everything, and they only do stand up meeting [. . .]". This indicates the influence of an individual can affect the abandonment of Agile practices, but also how long they were adopted until abandonment.

From Table 5, we can see that there could be more than one cause to abandon an Agile practice. For example, we identified multiple reasons for abandoning *tracking progress*. One of the more common reasons is lack of perceived values. To abandon *tracking progress* due to the decrease of perceived value seems counter intuitive because the need for tracking progress would increase as

the product grows and more tasks are associated with delivering the product.

In the case of tracking progress, it is possible that the practitioners did not completely abandon tracking progress altogether, but abandoned tracking progress according to the definition in the survey. Respondent R14, R23, R33, and R38 indicated in the survey and interviews that they use *Kanban board* to replace burn up or burn down charts as a means of tracking progress.

The results from the survey and follow up interviews indicate that there could be multiple factors that can contribute to abandoning an Agile practice. Engagement, knowledge, and dependencies between development teams can contribute to the abandonment of one or more Agile practices.

4.3. Perceived success of Agile practice adoption (RQ3)

In Figure 5, we looked at the perceptions of success of Agile practice adoption by industry domain to see whether our sample shows differences between domains.

From Figure 5, we can see that the adoption of Agile practices was generally perceived as being successful. Most of the respondents (30; 60%) perceived the adoption of Agile practices as successful and 11 (22%) as very successful. Only one respondent (2.8%) perceived the adoption of Agile practices in his/her organization to be unsuccessful. No respondent answered "Very unsuccessful". There were only minor differences between domains.

In the follow up interviews, we identified a number of factors that contribute to the perceived success:

- Management: Trust and commitment from managers on Agile adoption (R32), a clear vision of Agile transformation from the upper management (R37).
- Leadership: Ability of the leader to provide guidance (R38).
- Team members: Engagement (R36), experience and technical skills (R40).

During the follow up interview, R11 who indicated unsuccessful adoption of Agile practices mentioned that the issue was with the company policy, which is also related to management, of providing documentation at the end of every sprint: "if you want to be effective, with the small chunks of deliverables, there are more effort because the amount of procedure is the still the same as the big one. Agile implementation somehow is "heavier" on the procedure side. For every deliverable we need to provide documents like technical documentation, deployment guide, training material, [user acceptance test] sign off". Respondent R11 also felt that the kind of product they were developing did not fit Agile: "You need 6 months to develop the core engine. I cannot split a function into two releases, because it will be useless for the user. We have heavy rule engine and workflow. For this type of project, Agile does not work".

The respondents who perceived Agile practice adoption as very successful or successful (43 respondents) were primarily from small and medium sized organizations (25 and 15 respondents respectively out of the 43 respondents). This, however, does not indicate that Agile practice adoption is more successful in small organizations. We simply cannot make such assertions, since we have a small sample size and more than 50% of the respondents were from small organizations. Performing inferential statistics to examine the correlation between success rate and organization size would not be meaningful.

Overall, our survey respondents perceived their Agile practice adoption to be successful. We did not find significant variations of perceived success across the different domains. We identified factors that may influence the perception of success from the respondents, such as management, leadership, and team members.

4.3.1. Success rates: retained vs. abandoned practices (RQ3.1)

We also compared the success rates of 40 respondents who retained all adopted Agile practices and respondents who abandoned one or more Agile practices. From Figure 6, we can see that the perceived success of Agile practices was similar in both groups. This result indicates that

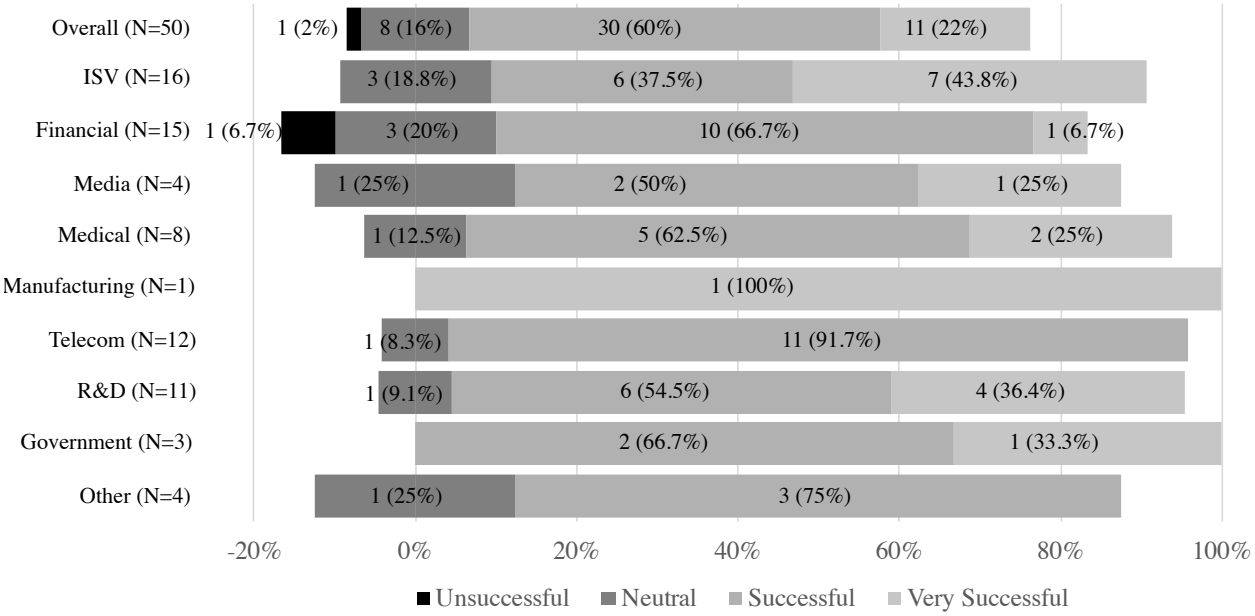


Figure 5. Perception of success of Agile practice adoption for all participants ("Overal", top row) and by industry domain (row 2–9). $N = 50$; one of the 51 respondents did not answer the question about perceived success

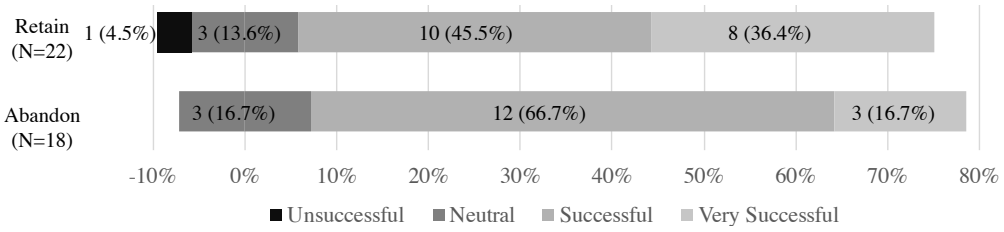


Figure 6. Success rates: retained versus abandoned practices

an abandoning of one or more Agile practices might be required to achieve or sustain an overall successful Agile adoption.

However, it is also important to remember that not all respondents adopted the same set of Agile practices. Those who achieved successful or very successful Agile adoption by retaining all Agile practices may have found the more suitable set of Agile practices or have successfully found an optimal way to tailor the Agile practices. We, however, do not claim that those who abandoned practices were less successful in selecting the suitable of Agile practices.

4.3.2. Measures of success (RQ3.2)

As we can see from subsection 4.3 and 4.3.1, our survey respondents generally perceived their Agile practice adoption to be successful. It is important to understand how success is measured since there could be different ways to perceive success. We collected measures of success from 35 respondents and classified them into product, process, and resource measures [27, Chapter 3, pp. 87–98]. Table 6 summarizes the measures that were reported the respondents and the number of respondents that reported them.

Among the product measures, "product quality" and "customer satisfaction" were named most frequently (12 and 9 times, respectively). Among the process measures, "time to deliver" was named most frequently (16 times). "Team spirit (happiness)" was the most frequently named resource measure.

Respondents considered a large diversity of indicators as being success-relevant, including

measures from all three categories. Table 6 lists 16 unique "process" measures, 11 unique "resources" measures, and 8 unique "product" measures. This result shows that success of Agile practice adoption can be perceived in many different ways.

Looking at the number of different measures and the number of respondents who contributed them, we can see that our respondents put much focus on how well a "process" is executed and on the quality of the "product". On a more detailed level, the respondents focused on product quality, customer satisfaction, and time to deliver, and good team spirit. This result is in line with the overall goals of the Agile manifesto [30] and the principles behind it.

We can see that the respondents reported measures at different levels of granularity. For instance, most respondents referred to "product quality" or "customer satisfaction" as measures for quality without going into detail about how those were measured. Few respondents named actual specific measures, like "number of defects/bugs" or "number of met sprint goals".

When looking at the respondents' experience and roles, we could not identify any specific patterns regarding the measures they provided. Respondents with more technical roles, e.g., developers or testers as well as those with managerial roles provided both specific and generic measures.

5. Discussion

In this study, we conducted a survey and 11 interviews on Agile practices adoption and abandonment. To guide the discussion, we reflect our findings and compare them against known recommendations from Agile maturity models (AMMs).

The respondents of our survey indicate that face-to-face meeting and tracking progress are frequently used. Meanwhile, TDD and pair programming are less commonly used by our survey respondents. From the follow up interviews, we identified different rationales from our respondents why some Agile practices were never used. The rationales for never using certain Agile practices indicate that all Agile practices are not always applicable in different contexts. Agile practices are not used due to incompatibility with the development context, challenges, or lack of management enforcement. AMMs typically recommend to gradually add more and more Agile practices (see Table 1) without considerations on whether the practices are suitable within a context. For example, 24 of our respondents never used TDD, but two out three AMMs that we exemplified in this paper recommend that TDD is to be introduced. Our study also indicates that Agile practices could be modified from its definition. However, the AMMs that we exemplified in this paper do not provide their definitions of the Agile practices. This raises the question regarding the suitability of AMM in industry.

The result of our survey indicates that not all Agile practices are sustainable. Eighteen of the respondents have abandoned one or more Agile practices. Our survey also shows that Agile practices were more frequently abandoned within the first six months after their adoption. Meanwhile, some Agile practices, like *continuous integration, planning game*, and *collective ownership* were adopted for extended period of time. This finding complements the findings of a previous study by Solinski and Petersen [3]. The AMMs indicate that Agile practices are to be gradually added. However, in certain contexts, it is not always possible to sustain a practice, as indicated by a number of our respondents. The question that needs to be raised when adopting an AMM is, if a practice is abandoned, how would this affect the practices that are to be adopted next? And how would this affect the overall maturity? The findings from our study add more questions to the suitability of the suggestions in the AMMs.

One of the rationales for abandoning Agile practices was the influence of a person. For respondent R14, *on-site customer* was introduced by the IT manager, the person's skills and abilities were so crucial that upon his departure from the organization, the practice had to cease. Meanwhile, respondent R33 the influence of one very opinionated individual convinced the rest

Table 6. Measures of success

Category	Measure	No. of respondents
Product	Product quality	12
	Customer satisfaction	9
	Number of defects/bugs	2
	Number of relevant working products/deliverables	2
	Other: Number of newly acquired users, code quality, code change quality, business value	1 each
Process	Time to deliver	16
	Cost	3
	Delivery frequency/cadence	3
	Lead time	2
	Ease to track progress (transparency)	2
	Other: Time to resolve defects, time to implement change, correct use of development process, effective use of Agile practices, number of development issues, amount of maintenance work, number of story points, number of released new features, number of met sprint goals, non ad-hoc development process, velocity	1 each
Resource	Team spirit (happiness)	6
	Budget conformance	2
	Productivity	2
	team autonomy	3
	Other: Collaboration, stress level, team engagement, ownership, mutual understanding, continuous learning, collective ownership	1 each

of the team to stop using 13 Agile practice. The case reported by R14 and R33 shows the presence of a "maverick" [31], a highly competent and influential individual that can influence the introduction and abandonment of Agile practices. The AMMs generally suggested that Agile practices are to be introduced in certain orders, and do not provide details on how these practices are to be introduced or sustained. This indicates that the AMMs have not considered the social aspects and uniqueness of different software development teams.

The results of our survey and interviews also indicate that an Agile practice could be abandoned because it needed another practice to be established beforehand or concurrently. For example, *tracking progress* was abandoned because *sprint planning* was not yet used (as reported by R14). This suggests that there might be dependencies between Agile practices, which the practitioners may yet to be aware of. In such cases, it would be preferred if practitioners can turn to the AMM. However, when we look at the examples of the AMMs in Table 1, we can see that each AMM has different suggestions as to which practices are introduced at which maturity level. For example, Patel and Ramachandran [5] suggested that *tracking progress* need to be introduced at the same time as *planning game*; such suggestion may not work in favor of R14. However, Sidky et al. [6] suggest that (collaborative) *planning game need* to be introduced before *tracking progress*, which could have provided a better guideline for R14. This indicates there could be a need for guidelines. However, instead of suggesting to gradually introduce Agile practices in fixed orders, like the AMMs, more research can be directed to evaluate which Agile practices need to be introduced first, or later, given the contexts of the software teams or organizations.

The result of our survey indicates that practitioners, both who retained and abandoned one or more Agile practice perceive their Agile practice implementation to be successful. AMMs typically suggest that Agile practices should be continuously added in a certain order to achieve successful Agile adoption [4–6]. This indicates that successful Agile adoption could still be achieved

without following the suggestions from AMMs. Our follow up interviews also revealed that an Agile practice could be replaced by another practice, such as a Lean practice. This shows that introducing Agile practices may not be as straightforward as what AMMs suggest. The follow up interviews also revealed a number of factors that could contribute to success, such as, management, leadership, and team members. This indicates the AMMs lack of consideration of the different situations and contexts in different software development team. This, again, raises the question on the merits of gradually introducing Agile practices in a certain order as suggested by AMMs.

Most of our survey respondents (82%) perceived that their Agile practice adoption to be successful and very successful. However, our respondents do not measure success the same way, for example, 12 respondents use product quality as a measure of success, and six respondents measure success given the team happiness. It indicates that success is perceived differently in different contexts by different respondents. A similar result is reported by Solinski and Petersen that indicate practitioners have different priorities on the perceived benefits and limitations of Agile practices [3]. The AMMs do not consider such prioritization of benefits and limitations that practitioners may have. This further highlights the limitation of a hierarchical approach to Agile adoption like the AMMs, as previously suggested by Gren et al. [32]. More research is needed to support practitioners in deciding which Agile practices are suitable for adoption given the benefits that they prioritized.

The results of our survey suggest that retaining or abandoning Agile practices can lead to a successful Agile adoption. This shows that Agile adoption is not as straightforward and gradual as suggested by the AMMs [4–6]. Practitioners may need to abandon, or very rarely pause, the implementation of one or more Agile practices. This indicates that practitioners are constantly assessing whether Agile practices are delivering the values they expected. Sidky et al. [6] included a step to assess whether to continue or discontinue the whole Agile transformation process, but not at the practice level. Practitioners might need support to systematically evaluate their state of Agile adoption so that decisions to add, modify, discontinue, or replace a practice is based on a rigorous and traceable process.

Implications towards Agile adoption guidelines. We noticed differences between the recommendations in AMMs and the results of our survey. At the same time, our survey also indicates the need for Agile adoption guidelines. Such guidelines need to take into account that Agile practices might not be sustainable and that there might be dependencies between Agile practices, as indicated by one our respondents, that suggests certain orders or combinations of adoptions. Furthermore, the situations and operating environment of software organizations may change [33]. The guidelines need to provide an appraisal means for practitioners on the benefits and limitations of adopting Agile practices, given the changing situations.

Implication towards Agile research. The results of our survey shares similarity to those of Kurapati et al. [16]. However, we also observed some differences, particularly pertaining to the adoption rate of *planning game*. The respondents in our survey indicate that *planning game* is a commonly used practice (47 out of 51 respondents), but Kurapati et al. reported the opposite. We observed that Kurapati et al. defined the practices slightly different. Their definition of *planning game* includes the presence an *on-site customer*. In our survey, we separated *planning game* from *on-site customer*. To be able to synthesize existing evidence regarding Agile practice adoption, there is a need for commonly agreed and consistent definitions of Agile practices.

The respondents in our survey indicate that *TDD* and *pair programming* are less commonly used practices. This result corroborates with past surveys such as [16] and [17]. *TDD* and *pair programming* are also less frequently abandoned. This observation is rather interesting because a tertiary literature study in Agile shows that TDD and pair programming is highly studied [34]. There are also many reports on their benefits and limitations to name a few: [35, 36]. This raised the question of whether knowing better the benefits and limitations of different Agile practices

can help practitioners to make better decisions on whether to introduce a practice. Therefore once the decision is made to adopt such practices it is based on an informed decision. Thus the practices are less likely to be abandoned.

6. Conclusion

We conducted a survey on Agile practices with a particular focus on when adopted practices were abandoned. We received 51 valid answers, 40 provided detailed start and end period for the practices. We also conducted 11 follow up interviews with the survey respondents. In the following, we revisit our research questions by summarizing answers:

RQ1. What is the rate of adoption of Agile practices? The rate of adoption of each practice can be seen in Figure 4. Commonly adopted practices by our respondents were *face-to-face meeting*, *tracking progress*, and *planning game*. Comparably less commonly adopted practices by our respondents were *TDD* and *pair programming*.

RQ2. Which Agile practices have been abandoned? All 17 Agile practices included in this survey have been abandoned at some point (see Table 4). Consistent with the answer to the previous research question, the more commonly used practices, particularly *tracking progress* and *planning game*, also had high abandonment ratio. The rationales for abandoning Agile practices include lack of perceived values, the influence of a person, and team member discomfort. Agile practices were used between 6–60 months until they were abandoned. Most of our respondents abandoned practices within the first half year of the introduction. Agile practices are less likely to be abandoned by our survey respondents after three years (36 months) of use.

RQ3. What is the perceived success rate of Agile practices implementation? The adoption of Agile practices was perceived as being successful or very successful. Only one respondent perceived the Agile adoption as unsuccessful and none as very unsuccessful. The respondents used a large variety of measures of success. The following measures were used by the majority of respondents: product quality, customer satisfaction, and time to deliver. Furthermore, our survey indicates no differences in the perceptions of success between respondents who abandoned practices and those who retained them. This result indicates that some teams or organization needed to abandon some practices to achieve or maintain an overall successful adoption of Agile methodologies.

Future work: For future work, we suggest the following avenues of research: (1) examine how different Agile practices contribute to maturity (2) better understand the impact of gradually adding, or abandoning Agile practices, and (3) developing a common definition of Agile practices to ease aggregation of evidence.

Acknowledgement

This work had been supported by ELLIIT, a Strategic Research Area within IT and Mobile Communications, funded by the Swedish Government.

References

[1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods – review and analysis," VTT Publications, Tech. Rep. 478, 2002.

[2] L. Williams, "Agile software development methodologies and practices," in *Advances in Computers*, ser. Advances in Computers, M.V. Zelkowitz, Ed. Elsevier, 2010, Vol. 80, pp. 1–44.

[3] A. Solinski and K. Petersen, "Prioritizing Agile benefits and limitations in relation to practice usage," *Software Quality Journal*, Vol. 24, No. 2, 2016, pp. 447–482.

[4] J. Nawrocki, B. Walter, and A. Wojciechowski, "Toward maturity model for extreme programming," in *Euromicro Conference, 2001. Proceedings. 27th*, 2001, pp. 233–239.

[5] C. Patel and M. Ramachandran, "Agile maturity model (AMM): A software process improvement framework for Agile software development practices," *International Journal of Software Engineering, IJSE*, Vol. 2, No. 1, 2009, pp. 3–28.

[6] A. Sidky, J. Arthur, and S. Bohner, "A disciplined approach to adopting Agile practices: The

Agile adoption framework," *Innovations in Systems and Software Engineering*, Vol. 3, No. 3, 2007, pp. 203–216.

[7] T. Schweigert, D. Vohwinkel, M. Korsaa, R. Nevalainen, and M. Biro, "Agile maturity model: A synopsis as a first step to synthesis," in *Systems, Software and Services Process Improvement*, ser. Communications in Computer and Information Science, F. McCaffery, R.V. O'Connor, and R. Messnarz, Eds., 2013, Vol. 364, pp. 214–227.

[8] M. Leppänen, "A comparative analysis of Agile maturity models," in *Information Systems Development*, R. Pooley, J. Coady, C. Schneider, H. Linger, C. Barry, and M. Lang, Eds., 2013, pp. 329–343.

[9] T. Schweigert, D. Vohwinkel, M. Korsaa, R. Nevalainen, and M. Biro, "Agile maturity model: Analysing Agile maturity characteristics from the spice perspective," *Journal of Software: Evolution and Process*, Vol. 26, No. 5, 2014, pp. 513–520.

[10] O. Ozcan-Top and O. Demirörs, "Assessment of Agile maturity models: A multiple case study," in *Software Process Improvement and Capability Determination*, ser. Communications in Computer and Information Science, T. Woronowicz, T. Rout, R. O'Connor, and A. Dorling, Eds., 2013, Vol. 349, pp. 130–141.

[11] Version One, *8th Annual State of Agile^{TM}Report*, 2013. [Online]. https://www.versionone.com/pdf/2013-state-of-agile-survey.pdf (Accessed May 2018).

[12] Version One, *9th Annual State of Agile^{TM}Report*, 2014. [Online]. https://explore.versionone.com/state-of-agile/9th-annual-state-of-agile-report-2 (Accessed May 2018).

[13] Version One, *10th Annual State of Agile^{TM}Report*, 2015. [Online]. https://explore.versionone.com/state-of-agile/versionone-10th-annual-state-of-agile-report-2 (Accessed May 2018).

[14] Version One, *11th Annual State of Agile^{TM}Report*, 2016. [Online]. https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2 (Accessed June 2017).

[15] Version One, *12th Annual State of Agile^{TM}Report*, 2017. [Online]. https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report (Accessed May 2018).

[16] N. Kurapati, V.S.C. Manyam, and K. Petersen, *Agile Software Development Practice Adoption Survey*. Berlin, Heidelberg: Springer, 2012, pp. 16–30.

[17] B. Murphy, C. Bird, T. Zimmermann, L. Williams, N. Nagappan, and A. Begel, "Have Agile techniques been the silver bullet for software development at Microsoft?" in *Proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement (ESEM 2013)*, Oct. 2013, pp. 75–84.

[18] M. Kropp, A. Meier, and R. Biddle, "Agile practices, collaboration and experience," in *Product-Focused Software Process Improvement. PROFES*, ser. Lecture Notes in Computer Science, P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, and T. Mikkonen, Eds. Springer, 2016, pp. 416–431.

[19] P. Ralph and P. Shportun, "Scrum abandonment in distributed teams: A revelatory case," in *The Pacific Asia Conference on Information Systems (PACIS)*, 2013, p. 42.

[20] J. Linåker, S.M. Sulaman, R. Maiani de Mello, and M. Höst, "Guidelines for conducting surveys in software engineering," Lund University, Tech. Rep., 2015.

[21] C. Robson, *Real world research*, 2nd ed. West Sussex, UK: John Wiley & Sons, 2011.

[22] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula, "Survey on Agile and lean usage in Finnish software industry," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Sept 2012, pp. 139–148.

[23] K. Petersen, "Is lean Agile and Agile lean?" *Modern Software Engineering Concepts and Practices: Advanced Approaches, IGI Global*, 2011, pp. 19–46.

[24] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E. Hanser, and C.R. Prause, "Hybrid software and system development in practice: Waterfall, scrum, and beyond," in *Proceedings of the 2017 International Conference on Software and System Process*, ser. ICSSP 2017, 2017, pp. 30–39.

[25] A. Forward and T.C. Lethbridge, "A taxonomy of software types to facilitate search and evidence-based software engineering," in *Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds*, ser. CASCON '08, 2008, pp. 14:179–14:191.

[26] J. Saldaña, *The Coding Manual for Qualitative Researchers*. SAGE Publications Limited, 2012.

[27] N. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*, 3rd ed. Boca Raton, FL, USA: CRC Press, Inc., 2014.

[28] K. Petersen and C. Gencel, "Worldviews, research methods, and their relationship to validity in empirical software engineering research," in *Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, 2013, pp. 81–89.

[29] S. Stavru, "A critical examination of recent industrial surveys on Agile method usage," *Journal of Systems and Software*, Vol. 94, 2014, pp. 87–97.

[30] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mallor, K. Shwaber, and J. Sutherland, "The Agile Manifesto," The Agile Alliance, Tech. Rep., 2001. [Online]. http://agilemanifesto.org/

[31] H. Sharp and H. Robinson, "Some social factors of software engineering: The maverick, community and technical practices," in *Proceedings of the Workshop on Human and Social Factors of Software Engineering*, ser. HSSE '05, New York, NY, USA, 2005, pp. 1–6.

[32] L. Gren, R. Torkar, and R. Feldt, "The prospects of a quantitative measurement of Agility: A validation study on an Agile maturity model," *Journal of Systems and Software*, Vol. 107, 2015, pp. 38–49.

[33] I. Nurdiani, S.A. Fricker, and J. Börstler, "An analysis of change scenarios of an IT organization for flexibility building," in *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*, 2015.

[34] I. Nurdiani, J. Börstler, and S. Fricker, "The impacts of Agile and lean practices on project constraints: A tertiary study," *Journal of Systems and Software*, 2016.

[35] A. Causevic, D. Sundmark, and S. Punnekkat, "Factors limiting industrial adoption of test driven development: A systematic review," in *Proceedings of the 4th IEEE International Conference on Software Testing, Verification and Validation (ICST 2011)*, March 2011, pp. 337–346.

[36] T. Dyba, E. Arisholm, D. Sjoberg, J. Hannay, and F. Shull, "Are two heads better than one? On the effectiveness of pair programming," *IEEE Software*, Vol. 24, No. 6, Nov 2007, pp. 12–15.

Appendix A. Definition of Agile practices in the survey

Agile practices definitions adapted from Solinski and Petersen [3]:

1. *Face-to-face meeting*: Team sits together, open space office facilitating interaction, video conference if the team is distributed.
2. *Self-organizing cross functional team*: Small team with no more than 10 members that consists of people with different competences (developer, tester, etc.). Team is independent, takes full responsibility of the task.
3. *On-site customer*: Continuous user involvement in the development process, customer can be consulted anytime if it is needed.
4. *Pair programming*: Two developers work together at one workstation.
5. *Planning game/sprint planning meeting*: The entire team participates in selecting the feature to be implemented in the following iteration.
6. *Tracking progress*: Tracking of the project progress using burn down chart, burn up chart.
7. *Refactoring*: Restructuring code for better understandability and reduced complexity.
8. *Iteration reviews/retrospective*: Meeting after each iteration to review the project, discuss threats to process efficiency, modify and improve.
9. *Short iterations & frequent releases*: Frequent releases of the software, early and continuous delivery of partial but fully functional software.
10. *Simple design*: Goal to design simplest solution.
11. *Time-boxing/sprint/iteration*: Fixed start and end dates are set for iterations and projects, e.g. 30 days sprint.
12. *Stand up meeting*: Short daily meeting where the whole team communicate and reflect on the completed and ongoing work.
13. *Metaphors & stories*: A metaphor is a very high level requirement outlining the purpose of the system and characterizes what the system should be like. The metaphor is broken down into short statement of the detailed functionalities called stories. The stories are kept in a backlog.
14. *Test-driven/test-first development*: Writing automated test cases for functionalities and then implementing (coding) the tested functionalities until the tests are passed successfully.
15. *Continuous integration*: Software is built frequently, even a few times a day, accompanied with testing (unit test, regression test, etc.).
16. *Coding standards*: Coding rules that are followed by the developers to make sure that developers write code in the same way.
17. *Collective ownership*: Everybody in the team can change the code of other developers in case of maintenance, bug-fixing or other development activities.

Appendix B. Survey design

Experience in Agile Practice Adoption (Page 1 of 2)

Throughout your professional career, you may have closely observed or personally experienced instances of Agile practices being adopted in different organizations. In this section of the survey, we would like you to reflect on a particular experience from your current or past employment regarding the adoption of Agile practices. An experience is your personal observation regarding the adoption or termination of Agile practices in the organization that you are currently employed or were employed in the past. Please reflect on an experience that you are most familiar with and answer the following questions with respect to this particular experience. If you want to share several cases or experiences, you can add further experiences later.

1. Which practices have been adopted/used in this case?

Please mark all practices that are/were used in this experience. For all practices that are/were used, please use the **blue** pointer to mark the **start** of a practice and **grey** pointer to mark the **end** of a practice. If a practice is still **in use** please drag the grey pointer to the end. If a practice has never been used please mark it with "Never used". If you are not sure about some practices, please mark it as "Don't know".

If you have difficulty to drag the slider pointer, please do the following:

1. Click on the start (blue) pointer.
2. Click the scale (the black line) on the respective year e.g., 2007, 2012 H1, etc.
3. Repeat step 1 for the end (grey) pointer.

In addition to the slider, please use the comment section if:

- You know precisely when one practice starts and ends, e.g., a specific date and/or month of the year that you marked on the slider.
- The slider does not accommodate the start and end of a practice, e.g., a practice started and ended before 2006.
- You have other remarks that you wish to add.

	Used	Never used	Don't know	Before 2006	2007	2008	2009	2010	2011	2012 H1	2012 H2	2013 H1	2013 H2	2014 Q1	2014 Q2	2014 Q3	2014 Q4	2015 Q1	2015 Q2	2015 Q3	2015 Q4	2016 Q1	In Use	Comment
Face-to-face meeting																								
Self-organizing cross functional team																								
On-site customer																								
Pair programming																								
Planning game/ sprint planning meeting																								
Tracking progress																								
Refactoring																								
Iteration reviews/ Retrospective																								
Short iterations & frequent releases																								
Simple Design																								
Time-boxing/ Sprint/ Iteration																								
Stand up meeting																								
Metaphors & stories																								
Test-driven/ test-first development																								
Continuous integration																								
Coding standards																								
Collective ownership																								

From this point onwards, we will refer to Agile practices adoption that you described in question 1, as **this experience** .

2. In this experience, did/do you find the adoption of Agile practices to be successful?

Very Unsuccessful | Unsuccessful | Neutral | Successful | Very Successful | Don't Know

3. In this experience, what is/was the measure of success?

4. In this experience, what are/were the limitations/challenges of adopting Agile practices in this particular way or order? (If any. Optional)

5. In this experience, why are/were the Agile practices adopted in this particular way or order?

6. In this experience, why were some practices terminated? (If any. Optional)

7. If you have other remarks or comments that you would like to add to your answer, please kindly put them here (Optional).

Back Next

Context (Page 2 of 2)

In this section we would like to gather some contextual information pertaining to this experience that you described in Page 1.

8. This experience is/was

- ○ Current experience from my current employment
- ○ Past experience from my past employment

9. Please select the role(s) that you assume/assumed in this experience.

- ☐ Programmer
- ☐ Project/Program Manager
- ☐ Business Analyst
- ☐ Systems Analyst
- ☐ Systems Architect
- ☐ Quality Assurance
- ☐ Trainer
- ☐ Scrum Master
- ☐ Product Owner
- ☐ Consultant
- ☐ Department/unit head
- ☐ C – Level Manager (CEO, CIO, etc.,)
- ☐ Other

10. How many years have/had you been employed in the organization where the experience takes/took place?

- ○ < 1 year
- ○ 1 – 3 years
- ○ 3 – 6 years
- ○ 6 – 10 years
- ○ > 10 years

11. In this experience, approximately, how many team members are/were affected by the adoption of Agile practices?

- ○ Less than 50 people
- ○ 50 – 249 people
- ○ 250 – 4999 people
- ○ More than 5000 people

12. In this experience, which of the following statements best describe the distribution of team members affected by the adoption Agile practices?

Distributed means that development teams are distributed across multiple locations

- ○ This experience involves/involved one development team that is collocated at a single location
- ○ This experience involves/involved multiple development teams that are collocated at a single location
- ○ This experience involves/involvedone development team that is located at multiple locations
- ○ This experience involves/involved multiple development teams that are located at multiple locations
- ○ Other

13. In this experience, how was/is the the adoption of Agile practices determined?

- ○ Each project determines its own development process
- ○ Each department/unit determines the development
- ○ The company defined a common development process
- ○ I don't know
- ○ Other

14. What is/was the main development type(s) in this experience?

- ☐ Bespoke development for a specific external customer
- ☐ Market driven for large open market of potential customers
- ☐ In-house development (developing software for internal organization use)
- ☐ Maintenance
- ☐ Other

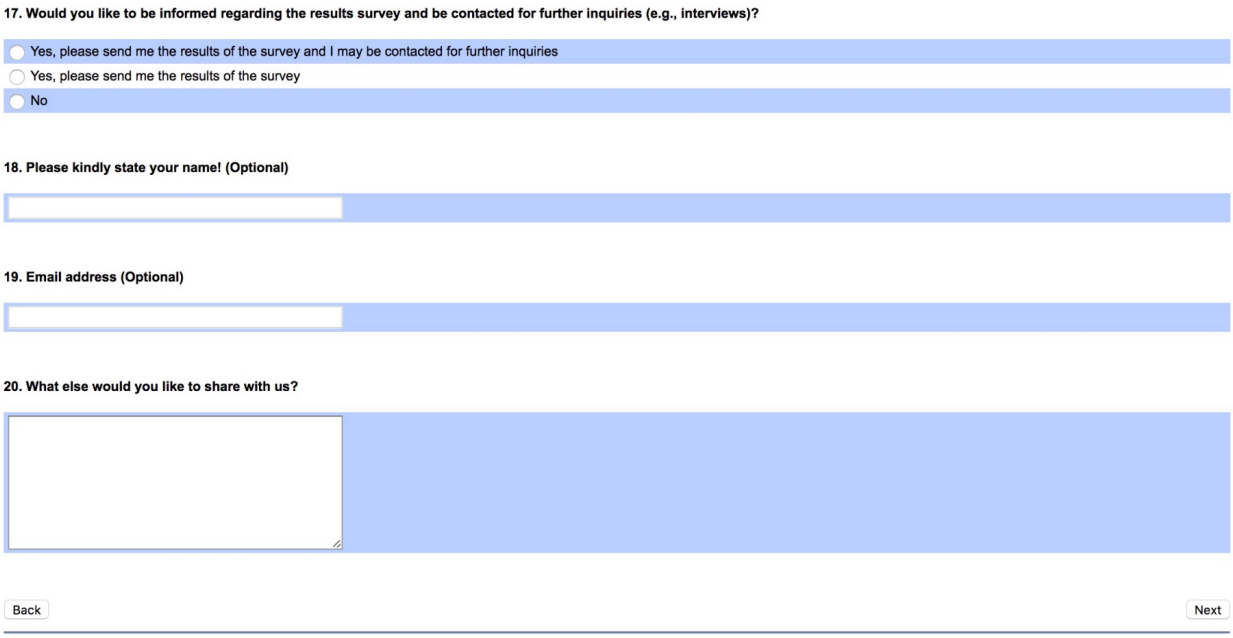
15. Which industry domain(s) do best describe the context of this experience?

- ☐ Independent Software Vendor
- ☐ Financial Services
- ☐ Media & Entertainment
- ☐ Healthcare/Medical
- ☐ Military
- ☐ Manufacturing
- ☐ Telecom
- ☐ Research & Development
- ☐ Government
- ☐ Other

16. Which type(s) of system(s) do best describe the context of this experience?

- ☐ Data-dominant system (e.g., web browsers, human resource management, e-commerce, data mining, etc.)
- ☐ Systems software (e.g., operating systems, networking/communication, anti-virus, database/email/ftp server, integrated development environment, etc.)
- ☐ Control dominant software (e.g., firmware, embedded systems, real-time control software, process control software)
- ☐ Computation dominant software (e.g., operation research, signal processing, image/video editor, robotics/cybernetics, etc.)
- ☐ Other

Back Next



Appendix C. Survey thoroughness assessment

Calculating thoroughness score. We summed up the weights for every criterion that was fulfilled by this survey (total score). Then, we divided the obtained total score by the total weight of all criteria. For more details on survey thorough assessment, see [29].

Table C.1. Survey thoroughness assessment based on [29]

Criteria	Weight	Score	Criteria	Weight	Score
Objectives	1	1	Questionnaire evaluation	3	3
Sponsorship	1	0	Questionnaire	3	3
Survey method	4	4	Media	1	1
Conceptual model	4	4	Execution time	1	1
Target population	4	4	Response burden	1	0
Sampling frame	5	5	Follow-up procedures	2	0
Sampling method	5	5	Responses	3	3
Sample size	5	5	Response rate	5	5
Data collection method	3	3	Assessment of trustworthiness	5	0
Questionnaire design	4	4	Discussions of validity threats	3	3
Provisions for securing trustworthiness	3	3			
Total weight: 66			Total score: 57		