

# **Agile Practices in Practice**

## **- A Mapping Study**

**Authors:**  
Philipp Diebold  
Marc Dahlem

IESE-Report No. 022.14/E  
Version 1.0  
June, 2014

---

A publication by Fraunhofer IESE



Fraunhofer IESE is an institute of the  
Fraunhofer Gesellschaft.

The institute transfers innovative software  
development techniques, methods and  
tools into industrial practice, assists com-  
panies in building software competencies  
customized to their needs, and helps  
them to establish a competitive market  
position.

Fraunhofer IESE is directed by  
Prof. Dr. Dieter Rombach  
(Executive Director)  
Prof. Dr.-Ing. Peter Liggesmeyer  
(Scientific Director)  
Fraunhofer-Platz 1  
67663 Kaiserslautern  
Germany



# Agile Practices in Practice - A Mapping Study -

Philipp Diebold  
Fraunhofer IESE  
Fraunhofer-Platz 1  
67663 Kaiserslautern, Germany  
philipp.diebold@iese.fraunhofer.de

Marc Dahlem  
Software Engineering Research Group  
University of Kaiserslautern  
67663 Kaiserslautern, Germany  
m\_dahlem@cs.uni-kl.de

## ABSTRACT

*Background:* Agile software development has been increasingly adopted during the last two decades. Nonetheless, many studies show that using agile methods as defined in the literature does not work very well. Thus, companies adapt these methods by just using parts of them (called agile practices). *Objective:* The goal of the literature study was to understand which agile practices are used in industry under different circumstances, such as different project types, domains, or processes. *Method:* We conducted a mapping study of empirical studies using agile practices in industry. The search strategy identified 1110 studies, of which 24 studies including 68 projects were analyzed. *Results:* The results of this study show that there are practices that are used more often and that the domain and the process also influence the application of different practices. Additionally, the findings confirm the assumption of Ken Schwaber that in most cases, agile methods are not used “completely” but that rather certain practices are adopted. *Conclusions:* Our results can be used by researchers to get a better idea of where and how to follow up research as well as by practitioners to get a better idea of which practices fit their needs and which are used by others. Therefore, our contribution increases the body of knowledge in agile practices usage.

## Categories and Subject Descriptors

[General and reference]: cross-cutting tools and techniques – *empirical studies, evaluation*. [Software and its engineering]: software creation and management – *software development process management, software development methods, agile software development*.

## General Terms

Measurement, documentation, experimentation.

## Keywords

Empirical SE, systematic review, mapping study, agile software development, agile methods, agile practices, industrial usage.

## 1. INTRODUCTION

Software processes are an important part of software engineering that influence product outcome [13]. Therefore, the software engineering world has come up with a huge number of software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EASE'14, May 13–14, 2014, London, UK.

Copyright 2014 ACM 1-58113-000-0/00/0010 ...\$15.00.

processes, which have evolved in the past decades. The biggest change in software processes was the introduction of agile processes as a contrast to plan-based processes.

Although agile software development started some decades ago, publications on applying agile methods in industry mainly started to appear around 2005. However, Ken Schwaber, one of the inventors of Scrum, states that 75% of the companies that claim Scrum usage do not really use Scrum [22]. This also seems to be the case for other agile methods, such as XP. For this reason, our research and the literature study we conducted focus on individual agile practices instead of “complete” agile methods.

This mapping study aims at evaluating and presenting empirical findings on the current usage of agile practices in industry. Additionally, we also provide various possible reasons, strengths, and weaknesses encountered when using different agile practices in different domains and processes. This overview will be important for practitioners as an introduction to the state of the art in this area and as a source of ideas for their companies. It is also interesting for researchers, showing them where to focus their research in order to deal with topics that are hot in industry.

This paper is organized as follows: In Section 2, we give an overview of agile software development, especially agile practices, identify existing reviews, and state our research objectives. Section 3 in detail describes the method that was used in this research. Section 4 reports the findings by providing information about the studies and projects. These results, their benefits, and limitations are further discussed in Section 6 following the threats to validity in Section 5. In Section 7, we summarize the paper, draw conclusions, and provide recommendations for future work in the area of agile practices.

## 2. BACKGROUND – AGILE PRACTICES

This section starts with a description of the field of agile software development, its history, and the current state of the practice, as well as a description of agile practices. This is followed by a summary of previous reviews regarding agile development and their relationships to our research. Finally, the need for our study is specified in the research questions.

### 2.1 Agile Software Development

Agile Software Development is defined in the Agile Manifesto [6] on the level of core values and principles. The implementation of the principles are both the agile practices and the combination of agile practices called agile methods.

The number of methods grew over the years so that there are now around 20 different agile or lean methods. How these methods evolved over the years is described by Abrahamsson et al. [2]. Nonetheless, only a small number is used in industry. The most commonly used methods for agile development are Scrum and Extreme Programming (XP).

Even though they are the most common, there are only a few companies that really apply these methods as described in literature. Most companies adapt or tailor these methods due to various problems such as complexity encountered during the introduction or change to agile development [9] [22]. This leads to many agile method adaptations that appear in literature, e.g. [10]. Most often they either omit specific parts of the original agile method, change them, or replace them with traditional aspects. The most prominent adaptation is the so-called “ScrumBut” method, which uses Scrum to some extent. Ken Schwaber, one of the Scrum guide [24] authors, estimates that “75% of all companies claiming to do Scrum, don’t do Scrum” [22], because they are adapting or omitting some practices.

## 2.2 Agile Practices

In contrast to the agile methods explained above, agile practices are one level below, because these are a small and very specific part of a method that addresses different aspects. Some known examples are pair programming or daily meetings. There is no common literature definition of agile practices, but an easy explanation can be given by considering XP [5], which is a collection of primary, business, and corollary practices.

Before conducting our review, we defined a set of agile practices that we could use to answer the research questions at the end. This was necessary because agile methods name their practices differently. For this reason we merged the practices of different agile methods and call them ‘universal’. However, they can still be mapped to the common agile methods such as Scrum.

To get a set of universal agile practices, we analyzed the most common agile methods, extracted the practices, and grouped them under a common name. A good starting point for this was the report of Abrahamsson et al. [1], which contains seven of the most common agile methods and provides the practices they use. These were then consolidated with the literature about the different methods, such as [24] for Scrum, [5] for XP, [14] for Adaptive Software Development, and [9] for DSDM.

This resulted in the following list of universal agile practices, which was used in our study, especially in Sections 4 and 5 of this paper:

1. Quality check
2. Refactoring
3. Customer involvement
4. Unattached communicative teams
5. Validation practice
6. Learning loop
7. Outcome review
8. Planning meeting
9. Time boxing
10. Common knowledge
11. Progress monitoring
12. Product vision
13. Evolving and hierarchical specification
14. Continuous integration/deployment
15. Delivering frequent releases
16. Small cross-functional teams
17. Daily discussion
18. Continuous specification analysis

This list summarizes the practices of the most common agile methods and was used in our study to collect evidence regarding our research questions and study goals.

## 2.3 Previous Reviews

Although agile software development is a relatively new paradigm, several literature reviews have been published already.

Most of these published reviews focus on agile methods such as [3], [9], [12], [21], and [26]. In addition to these, there are also some that focus on specific agile methods, such as on the most common one, Scrum [7]. Some of these reviews also try to focus on a specific domain, such as embedded systems ([3] and [26]).

As mentioned above, in most cases no agile methods are used purely as stated in the literature, for example in the Scrum Guide. However, nothing can be found in the literature that deals with the usage of agile practices. Besides this, no mapping study has been conducted yet in this field.

## 2.4 Objectives of this Review

Agile software development is an area that receives high interest in all domains. The 7<sup>th</sup> annual State of Agile Development survey shows that 84% of the companies are using agile development and most of them for at least about two years [25].

As stated above, we are not aware of any literature investigation of the usage of agile practices in industry and the specific domains, which is the main reason why we conducted this study.

The results could help practitioners and researchers to improve development processes and adapt them to their specific needs. Additionally, our study can indicate which agile practices are supported by scientific studies. Formally, the main study goals (SG) of this literature review were:

*Analyze **agile practices** in order to explore their **industrial usage** (SG1) with respect to their distribution over different **domains** (SG2) and **processes** (SG3) from the perspective of software engineers.*

The objective of our review was to answer the following research questions:

1. Which agile practices are used in industry? (RQ1)
2. Which agile practices are used in which domain?  
How are these practices distributed? (RQ2)
3. Which agile practices are used in which processes?  
How are these practices distributed? (RQ3)

Our idea was also to identify relationships between these various aspects. In addition, we also wanted to gather some background information about reasons for a specific usage or relationships with domain-specific aspects.

## 3. REVIEW METHOD

Influenced by the established method of systematic literature reviews (SLR) [17] we performed the mapping study [3] [20]. Since we first started with the idea of a SLR we performed the following stages: (1) development of the review protocol, (2) identification of inclusion and exclusion criteria, (3) search for relevant studies, (3) critical appraisal, (4) data extraction, and (5) analysis. The remainder of this section describes these stages and the methods used in detail.

### 3.1 Protocol Development

We developed a review protocol for our study based on [16], which was enriched with guidelines and procedures from [13] and [15]. Our protocol was reviewed by a systematic literature review expert. This protocol includes all important elements of the study, such as background, research questions, search strategy, inclusion,

exclusion, quality criteria, and data extraction strategy. The most important elements are described in the following subsections.

### 3.2 Inclusion and Exclusion Criteria

Studies were included in this review if they presented empirical data on applying agile practices in a company and passed the minimum quality threshold (see Section 3.5). We only included studies by professional software developers from one or more industrial environments. Because this review was not restricted to any other specific type, it includes qualitative as well as quantitative studies. We only considered studies written in English. At this stage, there were no limitations regarding the publication date.

On the other hand, we excluded studies if their main focus was not on agile software development or if they did not present some empirical data. As a more specific refinement, we also excluded (controlled) experiments because they do not represent real industrial environments. In addition to this, we also excluded secondary studies such as systematic literature reviews as well as huge surveys because it is impossible to find data about the practices in a specific project context.

### 3.3 Data Sources and Search Strategy

Our search strategy encompassed the use of electronic databases that include the most important specific journals and conference proceedings for our topic:

- IEEEXplore
- ACM digital library
- Google scholar
- Springerlink

Because a piloting showed that the meta-search engine Scopus<sup>1</sup> covers all relevant sources of these databases for our topic, it was used as the sole search engine.

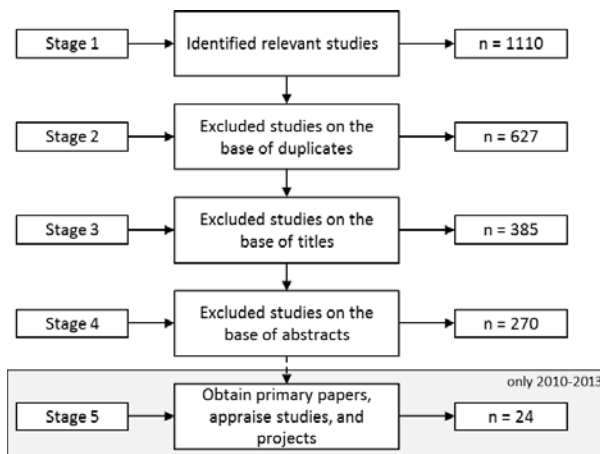


Figure 1. Stages of the study selection process

Figure 1 shows the systematic review process and the number of papers identified at each stage. In stage 1, the papers' titles, keywords, abstracts, and contents were searched, using the following search terms:

- (1) industry OR industrial OR practical environment OR industrial environment OR real environment OR real context OR case study OR field study OR company OR experience report OR exploration

- (2) agile practice OR iterative practice OR agile method OR agile artifact OR agile artefact OR agile process

While piloting some of the search terms, we found several synonyms, which were then included. All these search terms were combined by using the Boolean "AND" operator. This means that each term of (1) was combined with all the terms of (2), which resulted in 63 different searches. The 1110 results of all these searches were documented in one Excel file based on the Scopus export functionality.

In stage 2, we removed all duplicates as well as editorials, prefaces, discussions, workshops, panels, and article summaries. After this stage, we had 627 results.

### 3.4 Citation Management, Retrieval, and Inclusion Decisions

All the different steps were documented, starting with the export of the different searches to Excel. Here, we recorded all the data obtained from the Scopus export, such as author(s), title, abstract, year, source (including conference, journal, and page numbers), keywords, and many more. In addition to this given data, the Excel sheet also contained information about the study selection process such as codes for duplicates, title exclusion, abstract exclusion, review, and further discussion.

In stage 3, the authors went through the titles of all the papers remaining from stage 2 to determine their relevance for our review. At this stage, we first excluded all papers that are not in a software engineering environment at all (6 exclusions). After this step, the second iteration excluded all papers whose titles were clearly not about agile software development or which were from a non-industrial environment (191 exclusions). The last step in this stage was a joint review with the other author to check the reliability of the results. For the 627 papers, the number of agreements between both researchers was 582 (93%). We also computed Cohen's Kappa [7] to check the agreement. The Kappa coefficient of this stage was 0.54, which is "moderate agreement" [13]. All disagreements were discussed and resolved among the two researchers, finally resulting in an agreed result set for the following stages of 385 (61.4%) studies.

In stage 4, we went through the remaining 385 abstracts of the papers and checked their relevance for our study according to the inclusion and exclusion criteria. Here obscurities were directly discussed between the two researchers to reduce the amount of time needed. Again, all disagreements were resolved. At the end of this stage, we had 270 (70%) remaining results.

Because it was impossible to go through this huge number of results (within a short period of time) after stage 4, we decided to check the different publication dates and then focused on the most recent papers to obtain the primary studies.

We decided that all papers from the beginning of 2010 would be included in this study to get the most up-to-date results. The low number of papers from the year 2013 is due to the point in time when we conducted the study search, which was on 20<sup>th</sup> August 2013. After the exclusion based on abstracts in combination with the publication date threshold, we had 121 results spread over three and a half years (2013: 9; 2012: 32; 2011: 38; 2010: 42).

<sup>1</sup><http://www.scopus.com> (see Scopus title list)

### 3.5 Quality Assessment

In the last stage, each of the remaining 121 papers was reviewed to see if their main focus fits our study goals; this was followed by an assessment of their quality. Because we performed a mapping study and not a systematic literature review, the criteria for checking the quality were on a higher level and partly overlapped or refined with the inclusion criteria. We checked all papers according to the following five quality criteria:

1. The study reports *empirical research* independent of the type of empirical evidence, from lessons learned to industrial case studies.
2. The study reports a set of *agile practices* used.
3. The study was conducted in one or more *industrial environments* and this industrial context is described adequately.
  - a. The description of the *project domain* is important for our research according to SG2.
  - b. The description of used *process* or *lifecycle* is important for our research according to SG3.
4. The study describes the *motivation* for using agile software development and specific agile practices.
5. The study describes the *success* of using agile software development and specific agile practices.

The first three criteria constituted minimum thresholds for the inclusion of a study. The last two quality criteria were used to gain more insight into the reasons for the use of agile practices.

All these five quality criteria gave us confidence that all the findings of the remaining studies would be valuable contributions to our study.

In stage 5, based on the presented quality criteria, the 121 results from stage 4 with the time threshold led to the final 24 results (cf. APPENDIX). These study results include 68 projects that were used for data extraction, analysis, and discussion. A summary of the quality assessment criteria is presented in Table 1.

**Table 1. Quality Criteria**

1.	Is the paper based on any kind of empirical research?
2.	Are the agile practices used mentioned in the paper?
3.	Is there a clear and adequate description of the context in which the project was performed?
3.a / 3.b	Is there a clear and adequate description of the project domain / process in which the research was performed?
4.	Is the motivation for using agile software development and agile practices described?
5.	Is the success of using agile software development and agile practices described?

### 3.6 Data Extraction

During this stage, all necessary data for our mapping study from the 24 studies that remained after stage 5 was extracted based on a predefined extraction form (see Table 2). This form allowed extracting all data with all details needed for the analysis of the research questions. Because our focus was on the projects and not on the studies, the data extraction was performed individually for each project.

**Table 2. Data Extraction Form**

Attribute	Description
study ID	Unique identifier for the study <sup>2</sup>
project ID	Unique identifier for the project
study type	Specification of the study type, e.g., experience report, case study, etc.
company context	Specification of the company's context, e.g., country, name, size, experience, etc.
project context	Specification of the project's context, e.g., team, location, process, etc.
domain	Specification of the domain in which the study took place
agile practices	List of agile practices that were used in the respective project. Explicitly removing some practices from an agile method could also be possible.
notes	Any additional notes

To simplify and speed up the extraction process, we used the tool MaxQDA<sup>3</sup> during the qualitative assessment (Section 3.5). With this tool for qualitative data analysis of textual data the data extraction attributes (cf. Table 2) were marked with different codes. These were later inserted into another Excel file for the data extraction, which was used for the data and result analysis.

Our overall code system at the end included 904 text areas marked with codes in all papers, with most of them being on motivation (n = 117), paper context (n = 246), context (n = 203), and practices (n = 220). The specific context attributes process (n = 38) and domain (n = 74) for our research questions were used less frequently.

During this extraction and the encoding in the Excel file, we distinguished between fully using an agile practice, using it partially, and not using it at all. *Partially used* was inserted in our case to get more specific data and also to cover the aspect that some companies modify and adapt agile practices or only some of the team members use them. Most common example of a partial usage is an exceeded time-range of the practice time boxing.

## 4. RESULTS

At the end of the five stages of the study selection process, we identified 24 studies (cf. APPENDIX) on the usage of agile practices for software engineering in industry. We extracted the necessary data from the 68 projects included in these studies to answer our research questions and study goals. The data about which study or project uses which agile practices cannot be included here due to size restrictions. However, these data can be provided upon request.

In this section, we will first present an overview of the different studies and their projects by providing the general data that resulted from this study. Then the results needed to answer the research questions and discuss the three study goals regarding agile practices usage in general, in different domains, and in different processes.

<sup>2</sup> Using this ID, it is possible to find more study data such as bibliographic data, type of article, etc.

<sup>3</sup> <http://www.maxqda.com/>

## 4.1 Overview of Studies and Projects

The overall results of this study are presented in Figure 2, which shows the usage (fully and partially) and non-usage of agile practices based on all 68 projects in the 24 studies. The y-axis presents the different universal agile practices (see Section 2.2), with each practice being encoded with three bars for full usage (black), partial usage (gray), and non-usage (white). The x-axis of this figure shows how often the different agile practices were found in the papers.

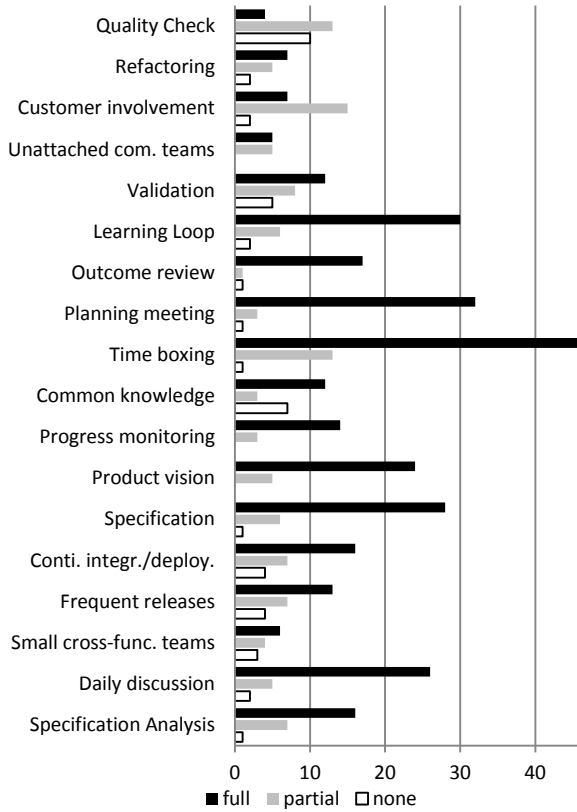


Figure 2. Overall Usage of Agile Practices

The figure clearly shows that the non-usage of agile practices is a publication problem because such results are very often missing in the literature. In general, the full usage of agile practices was found most often, even if there are some practices that are used more often partially than fully.

Across all 68 projects, all 18 agile practices were used. The number of usages ranges from at least 8 (11.8%) usages (full and partial usage combined) to 54 (79.4%). The most frequently full used agile practice is *time boxing* (n = 46), which only once was not used. In contrast, the least frequently used practice is *small cross-functional teams*. The most frequently adapted or partially used practice is *customer involvement* (n = 15), whereas other practices such as *outcome reviews* were only used partially once. Of the non-used practices, there is one that surpasses all others, the *quality check*. On the list of non-used practices, three are not mentioned and nine are mentioned equally or less than twice.

The black bars in Figure 2 for full usage of the agile practices show that there are six most commonly used practices: *time boxing* (n = 46), *planning meeting* (n = 30), *learning loop* (n = 28), *evolving and hierarchical specification* (n = 26), *daily discussion(s)* (n = 24), and *product vision* (n = 22). For the

adapted or partially used practices, there are only three that exceed 10 usages: *customer involvement* (n = 15), *time boxing* (n = 13), and *quality check* (n = 12).

Further details about the number of agile practices usage will be covered in the next subsections, with the focus on the study goals domains and processes.

## 4.2 Agile Practices Used in Different Domains

Before presenting the different practices used with respect to different domains, we show the overall distribution of the projects according to domains, independent of the practices. Since some papers did not provide the domain of the company or the project directly, we defined a list of domains during the extraction that included the domains depicted in Figure 3.

In discussion between both researchers, a mapping from the described domains to our predefined domains was performed. For this purpose, the extracted context, the project's description, or the company's name was used. However, since the domain was rarely mentioned in the papers, this may have introduced incorrectness and is a threat to validity of the results. Also, some domains could not be mapped effectively and were therefore grouped under 'Unspecific'.

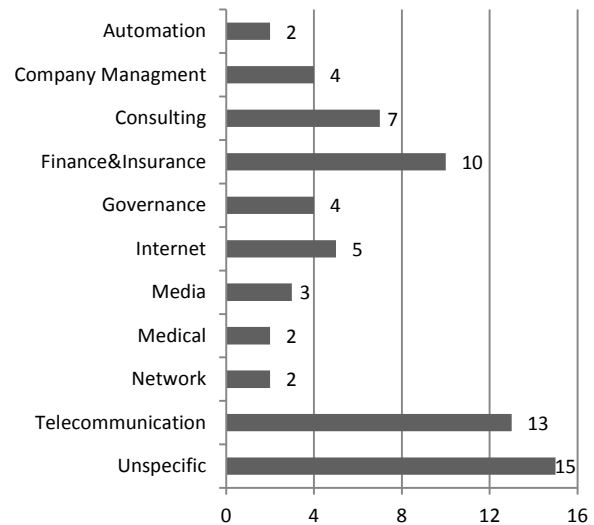
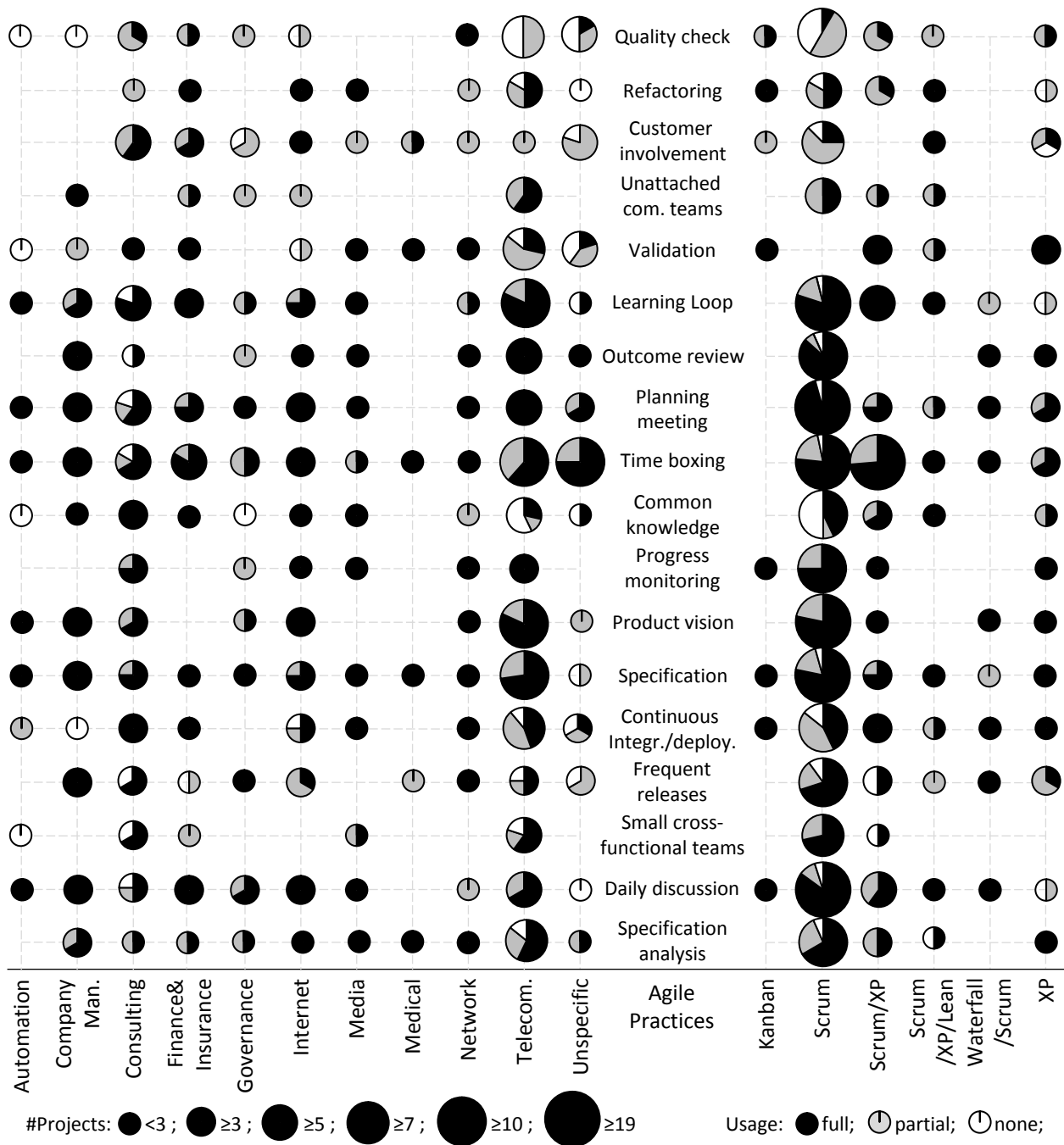


Figure 3. Distribution across domains

Figure 3 shows the distribution of these projects across the domains. Most projects were located in the Telecommunications domain (n = 13). The domains Finance and Insurance (n = 10) and Consulting (n = 7) had the second and third-highest number of projects. The remaining seven domains appeared between two and five times. Although the projects were mapped to different domains, 15 projects could not be assigned to any domain due to missing descriptions.

Based on these domain results, we mapped the occurrence of the different agile practices to the domains in Figure 4 (left part). The x-axis shows the 11 domains, while the 18 agile practices are on the y-axis. The six different sizes of the bubbles indicate the number of projects (<3; ≥3; ≥5; ≥7; ≥10; ≥19) that mention the practice in the specific domain. In addition, the three colors represent the usage of the practices similar to Figure 2 (full: black; partially: gray; none: white). E.g. *refactoring* in the telecommunications domain is used six times, in three projects full, in two partial, and in one not.



**Figure 4. Systematic Map of Agile Practices related to Domains (left) and Processes (right)**

As the size of the bubble shows, the most valuable results can be found in the Telecommunications domain (only domain that covers all practices), followed by Consulting and Finance & Insurance. 7 out of the 11 domains use almost all of the practices: Telecommunications, Consulting, Internet, Finance & Insurance, Networks, Governance, and Company Management. In contrast, there are other domains that use very few practices, such as Medical or Automation.

Among the different practices there is none that contains more valuable results than the others because almost each practice has a bigger bubble in at least one domain (often Telecommunications).

Only the practices *time boxing* and *evolving and hierarchical specification* are used in all eleven domains. Most of the other practices are used in many domains but not in all. Only *small, cross-functional teams*, *unattached communicative teams* (both in five domains), and *progress monitoring* (in six domains) are used rarely.

Figure 4 (left part) also shows that some specific practices are used similarly to the overall results (cf. Figure 2), but differently in the specific domains. For example, in the agile practices *quality check*, *refactoring*, *customer involvement*, and *unattached communicative teams* partial usage predominates. Non-usage is

mainly found for the practices *quality check* and *common knowledge*. None of the practices is used only fully.

### 4.3 Agile Practices Used in Different Processes

Before presenting the different practices used with respect to specific processes, we present the overall distribution of the projects according to processes, independent of the practices. Figure 5 shows that the process used most often is Scrum ( $n = 35$ ) and its combination with XP ( $n = 19$ ). It is also remarkable that 61 out of 68 projects mentioned the process used (89.71%).

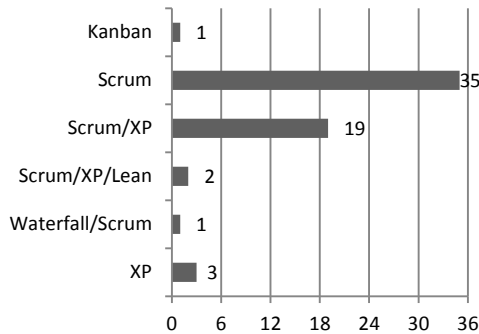


Figure 5. Distribution across processes

Based on these overall data, which show the processes found in all the different projects, the next analysis step was the mapping of agile practices to these processes. For this purpose, we created the map in Figure 4 (right part) with the agile practices mapped to the six different processes presented above. The x-axis shows the six processes, and the 18 agile practices are on the y-axis. The meaning of the map is equivalent to the mapping of the practices to the different domains explained above.

Similar to the domains, the most reliable results can also be found in the column that represents Scrum. This is followed by the combination of Scrum and XP. In addition to these two processes, the combination of Scrum, XP, and Lean as well as XP itself also used almost all agile practices. The only plan-based process mentioned, Waterfall enriched with Scrum, used nine of our agile practices. The process with the lowest number of mentioned practices is Kanban, which could be the case because we did not search for specific agile methods and Kanban seems less often published in combination with our search terms.

In contrast to the domains, among the processes there are more practices that are used in all processes: *evolving and hierarchical specification*, *continuous integration/deployment*, and *daily discussion*. Most of the other practices are used in five out of the six processes; only in Kanban or Waterfall they are not used. Only *unattached communicative teams* and *outcome reviews* are used in three processes. The least-used practice is *small cross-functional teams*, which is only used in Scrum and its combination with XP.

Similarly to the domain mapping, the processes also show practices that are more often used partially. These are the same four as for the domains, plus *validation practice* and *delivering frequent releases*. In contrast, there are no practices that show a significant amount of non-usage across several domains. Among the other practices, full usage dominates, whereas there is no practice that is only used fully.

## 5. DISCUSSION

Following the presentation of the results of our study and the threats to their validity in the previous sections, we will now discuss these results by considering the data from the results in terms of the study goals and research questions stated at the

beginning of our investigation. Additionally, we present a brief comparison between our findings and the agile survey in [25]. All this will result in several hypotheses being derived from the results and issues that will require further research in the future.

### 5.1 Research Questions / Study Goals

The first research question (RQ1) about the general usage of agile practices is discussed on a high level because all of them are used in industry, although some are used more often than others. It did not come as a great surprise that the *time boxing* practice appeared most often, as currently most companies try to increase their development time by using smaller increments. In addition, planning aspects seem to be a major issue in the development, which can be seen in the practices *planning meeting* and *daily discussions*. This might be the case because these are some of the few parts from the planning phase of traditional plan-based development that are also part of agile development. From our point of view, the high usage of the *learning loop* is one of the major ingredients of agile development. The high partial usage of some practices, which is mainly an adaptation or a usage in a specific area, has different reasons. For example in *time boxing*, the time intervals in the partial cases are higher than recommended in agile methods such as XP or Scrum, e.g., in [S3], [S14], and [S16]. In addition, *customer involvement* which is important for agile development, is often not performed as recommended, e.g., by XP with an on-site customer. The *quality check* has one of the highest proportions of partial usage and non-usage due to one specific practice in particular: *pair programming*. It is partially used because some projects only check the quality of highly secure parts of the system. This is the practice for which most of the publications also indicate non-usage, e.g., [S8] and [S14]. An interesting finding is the non-usage of *common knowledge*. The only reason we could imagine was the use of documentation instead of common knowledge.

With regard to the usage of agile practices in different domains (RQ2), the data shows that domains that are highly (safety-) critical such as Medical, Governance, and Automation only use a small set of agile practices and also report non-usage. In contrast, others try to use many of them, with some interesting findings. For example, *quality check* and *validation practices* are used in the Finance and Insurance domain. In contrast, in the Telecommunications domain, these techniques have lower usage. The reason is a contradiction with *time boxing* because “the length of the sprint created bigger pressure to get testing done in shorter time than before” [S20] and “the lack of time in the sprint to conclude the automated tests resulted in back to manual execution of regression tests” [S8]. These problems with *time boxing* were also found in other domains such as Governance, Finance, and Media because they adapted it to a longer time period. One company working in Consulting mentioned that “there were just a small number of new features and plenty of bugs or improvements to make. Some of the later could be postponed, but there were urgent bugs that could not wait for another Sprint to begin, despite the fact that the team had already reduced the Sprint length to one week” [S11]. A similar reason was given in a telecommunication company: “The sprint length made the sprint planning an impossible task as the support requests were pouring in daily with critical status” [S14]. Additionally, the Telecommunications domain had high non-usage of *common knowledge*, whereas other domains had knowledge sharing. E.g., [S14] reports the issue that “with collective code ownership, the code development would be on the whole team’s responsibility, and [...] nobody would care about the good quality of the code”. Another company mentioned

that there was no communication from the testers to the developers. [S8]

The results for the third research question (RQ3) support Kent Schwaber's statement that Scrum is often not used as described because there are many process combinations (with XP and/or Lean), adaptations, and non-usages. Nonetheless, the enrichment of Scrum by other methods or practices improves development. *Common knowledge* is used less often in Scrum, but if Scrum was combined with Lean or XP, the number of positive answers increased. In contrast, the practices *customer involvement*, *planning meetings*, *evolving and hierarchical specification* are nearly stable across all processes. For the usage of XP, some practices, e.g., *daily discussions*, *learning loop*, or *refactoring*, are disregarded compared to the other processes. Very interesting for us was the project with a Waterfall-Scrum combination described as follows: "During this time, Scrum is mentioned several times, and requirements are from that time on written in the form of user stories. However, new functionality was still developed in the form of waterfall projects, where all functionality was designed upfront. Requirements prioritization and selection was still done based on 'fingerspitzengefühl'" [S7].

## 5.2 Comparison to Agile Development Survey

The annual survey about Agile Software Development [25] and the most frequently used practices (in this survey called techniques) are compared to our results in the following.

Four of the top six practices of this survey (*daily standup*, *iteration planning*, *release planning*, and *retrospectives*) also appear as our most frequently used agile practices, even if they are called differently or are subsumed by our practices.

Only *unit testing* and *burn-down / team-based estimation* do not directly appear in our most frequently used practices. In our case, we assigned the estimation part to planning, so we also cover this aspect, but in our classification *progress monitoring*, *burn-down* ranks only tenth among the most frequently used practices. The usage of the last top practice of the survey, *unit testing*, differs widely from our *quality check*, which is the practice used least often. This is probably the case because most often, experiences and non-usage of pair programming are reported in publications.

## 5.3 Hypotheses

Based on these results and their subsequent discussion, we identified several hypotheses. They are rather specific, but we tried to generalize them for future work.

The fact that domains with fewer regulations such as Telecommunications seem to use fewer *validation practices* leads to the hypothesis that the "degree of usage of *validation practices* increases with the degree of regulation in the domain" (H1).

Because domains like Finance & Insurance are more restricted than most of the other domains, "stricter usage of *time boxing* should improve development" (H2).

In addition to these two hypotheses, we also assume that "there are common combinations of agile practices (H3), such as *evolving and hierarchical specification*, *learning loop*, and *product vision*".

## 6. THREATS TO VALIDITY

There are a number of threats to validity that we are aware of and attempted to minimize by means of various mitigation actions.

**Researcher's bias:** There is a researcher's bias because only one researcher was performing the different stages. We minimized this

bias by having another expert researcher perform some stages in parallel and comparing the results. For the remaining stages, both researchers discussed the results of each stage before starting the next one with an agreed set. The analysis after the extraction was also reviewed and discussed.

**Publication bias:** We acknowledge that it is nearly impossible to achieve complete coverage of all publications written in the specific topic. We tried to find as many of the relevant publications as possible by using a meta-search engine covering the most important search engines, conferences, and journals for this topic. In addition, we defined our research question in advance, performed a multi-stage process, and documented all inclusions and exclusions in order to avoid selection bias.

Additionally, our search especially the data sources did not cover industrial presentations on non-scientific conferences. This would increase the validity of the findings.

Another problem with publications is the absence of negative publications reporting on practices that did not work as expected. This is also the reason why the number of non-usages of agile practices is low. In general, this is a problem encountered in all literature reviews.

In addition, we mentioned in the review method that we had a look on the latest publications from 2010 up to mid of 2013. This was the case because of time restrictions.

## 7. CONCLUSION

This paper presented a mapping study regarding the usage of agile practices in industry. This study was described in detail, from the planning with the detailed research questions via the execution to the presentation and analysis of the result of the publications up to 2010. Among the results we presented a map showing the combination of agile practices with the domains and practices as stated in the study goals. The results of this study provide evidence that the usage of the appropriate agile practices is very important for industry.

By performing this study and analyzing its results, we have gained a better understanding of the usage of agile practices in industry in general and for different domains and processes in particular. The study also enriches existing literature studies in the area of agile software development, which did not cover the aspect of agile practices.

An important follow-up work to this study would be to analyze the published studies that are excluded after stage 4 in order to get more datasets. This is not that much time-consuming because we already performed stages 1 to 4 with all studies, downloaded them, and documented them in our Excel sheet. Additionally, an idea was a follow-up search specifically for the single agile practices. Another addition to increase the scope of the results would be to combine the findings of other literature reviews or to perform a detailed comparison with the Agile Development Survey [25].

Furthermore, we are planning to use the results of this study to prepare for interviews using interview guidelines to collect direct evidence from industrial companies. This would also reduce the bias that negative results were not published because we can discuss and collect these data face-to-face with practitioners from industry.

## ACKNOWLEDGMENTS

Parts of this research were conducted as a student project at the Software Engineering: Processes and Measurement Research Group of the Technical University of Kaiserslautern supervised by Prof. Rombach.

We would like to thank Liliana Guzmán for her help while performing this study. We also thank Constanza Lampasona and Andreas Jedlitschka for their comments, feedback, and review as well as Sonnhild Namingha for revising the paper.

## APPENDIX. Studies included in the Review

- [S1] Wang, X., Conboy, K., and Pikkarainen, M. 2012. Assimilation of agile practices in use. *Info. Sys. Jour.* 22, 6 (Nov. 2012), 435-455.
- [S2] Collins, E. and deLucena, V. 2012. Software test automation practices in agile development environment: An industry experience report. In *Proceedings of 2012 7<sup>th</sup> International Workshop on Automation of Software Test*. AST'12. IEEE Computer Society, Washington, DC, 57-63.
- [S3] Bass, J. 2012. Influences on agile practice tailoring in enterprise software development. In *Proceedings of AGILE India*. IEEE Computer Society, Washington, DC, 1-9.
- [S4] Korhonen, K. 2012. Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. *Software Quality Jour.* 21, 4 (Dez. 2013), 1-26.
- [S5] McHugh, O., Conboy, K., and Lang, M. 2012. Agile practices: The impact on trust in software project teams. *IEEE Software* 29, 3 (Jun. 2012), 71-76.
- [S6] Melo, C., Cruzes, D., Kon, F., and Conradi, R. 2013. Interpretative case studies on agile team productivity and management. *Information and Software Technology* 55, 2 (Feb. 2013), 412-427.
- [S7] Vlaanderen, K., Stijn, P., Brinkkemper, S., and Weerd, I. 2012. Growing into agility: Process implementation paths for scrum. In *Proceeding of 13<sup>th</sup> International Conference on Product-Focused Software Process Improvement*. PROFES'12. Springer-Verlag Berlin, Heidelberg, 116-130.
- [S8] Collins, E., Dias-Neto, A., and de Lucena, A. 2012. Strategies for agile software testing automation: An industrial experience. In *Proceedings of the 2012 IEEE 36<sup>th</sup> Computer Software and Applications Conference Workshops*. COMPSACW'12. IEEE Computer Society, Washington, DC, 440-445.
- [S9] Collins, E., Macedo, G., Maia, N., and Dias-Neto, A. 2012. An industrial experience on the application of distributed testing in an agile software development environment. In *7<sup>th</sup> International Conference on Global Software Engineering*. ICGSE'12. IEEE Computer Society, Washington, DC, 190-194.
- [S10] Tufail, R. and Malik, A. 2012. A case study analyzing the impact of software process adoption on software quality. In *2012 10th International Conference on Frontiers of Information Technology*. FIT'12. IEEE Computer Society, Washington, DC, 254-256.
- [S11] Fernandes, C. 2012. There and back again: From iterative to flow ... and back to iterative! In *Proceedings of the 2012 Agile Conference*. AGILE'12. IEEE Computer Society, Washington, DC, 103-110.
- [S12] Monteiro, C., da Silva, F., dos Santos, I., Farias, F., Cardozo, E., Leitao, A., Neto, D., and Filho, M. 2011. A qualitative study of the determinants of self-managing team effectiveness in a scrum team. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*. CHASE'11. ACM, New York, NY, 16-23.
- [S13] Senapathi, M. and Srinivasan, A. 2011. Understanding post-adoptive agile usage - an exploratory cross-case analysis. In *Proceedings of the 2011 Agile Conference*. AGILE'11. IEEE Computer Society, Washington, DC, 117-126.
- [S14] Korhonen, K.. Adopting agile practices in teams with no direct programming responsibility - a case study. In *Proceedings of the 12<sup>th</sup> International Conference on Product-Focused Software Process Improvement*. PROFES'11. Springer-Verlag Berlin, Heidelberg, 30-43.
- [S15] Hajjdiab, H. and Taleb, A. 2011. Agile adoption experience: A case study in the U.A.E. In *2011 IEEE 2nd International Conference on Software Engineering and Service Science*. ICSESS'11. IEEE Computer Society, Washington, DC, 31-34.
- [S16] Block, M.. Evolving to agile: A story of agile adoption at a small saas company. In *Proceedings of the 2011 Agile Conference*. AGILE'11. IEEE Computer Society, Washington, DC, 234-239.
- [S17] Smits, H. and Rilliet, K.. Agile experience report: Transition and complexity at cisco voice technology group. In *Proceedings of the 2011 Agile Conference*. AGILE'11. IEEE Computer Society Washington, DC, 274-278.
- [S18] Bahli, B., Benslimanne, Y., and Yang, Z. 2011. The impact of absorptive capacity on the ex-post adoption of agile methods: The case of extreme programming model. In *2011 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEM'11. IEEE Computer Society, Washington, DC, 1660-1664.
- [S19] Sindhgatta, R., Narendra, N., and Sengupta, B. 2010. Software evolution in agile development: A case study. In *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*. SPLASH '10. ACM, New York, NY, 105-114.
- [S20] Petersen, K. and Wohlin, C. 2010. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering* 15, 6 (Jul. 2010), 654-693.
- [S21] Hannay, J. and Benestad, H. 2010. Perceived productivity threats in large agile development projects. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM'10. ACM, New York, NY, 15:1-15:10.
- [S22] Hoda, R., Kruchten, P., Noble, J., and Marshall, S. 2010. Agility in context. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications*. OOPSLA'10. ACM, New York, NY, 74-88.
- [S23] Korhonen, K.. Evaluating the effect of agile methods on software defect data and defect reporting practices - a case study. In *Proceedings of 2010 7<sup>th</sup> International Conference on the Quality of Information and Communications*

Technology. QUATIC'10. IEEE Computer Society, Washington, DC, 35-43.

- [S24] Ktata, O. and Levesque, G. 2010. Designing and implementing a measurement program for scrum teams: What do agile developers really need and want? In *Proceedings of the Third C\* Conference on Computer Science and Software Engineering*. C3S2E'10. ACM, New York, NY, 101-107.

## REFERENCES

- [1] Abrahamsson, P., Salo, O., and Ronkainen, J. 2002. *Agile Software Development Methods: review and analysis*. Technical Report. VTT Electronics and University of Oulu.
- [2] Abrahamsson, P., Warsta, J., Siponen, M., and Ronkainen, J. 2003. New directions on agile methods: a comparative analysis. In *Proceedings of the 25th International Conference on Software Engineering*. ICSE'03. IEEE Computer Society, Washington, DC, 244-254.
- [3] Albuquerque C., Antonino P., and Nakagawa E. 2012. An investigation into agile methods in embedded systems development. In *Proceedings of the 12th International Conference on Computational Science and Its Applications*. ICCSA'12. Springer-Verlag Berlin, Heidelberg, 576-591.
- [4] Anderson S., Allen P, Peckham S., and Goodwin N. 2012. Asking the right questions: scoping studies in the commissioning of research on the organisation and delivery of health services. *Health Res. Policy and Syst.* 6, 7 (Jul. 2008).
- [5] Beck, K. 2004. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman, Amsterdam.
- [6] Beck, K., Beelde, M., van Bennekum, A., Cockburn, A., et al. 2001. *Manifesto for Agile Software Development*. Agile Alliance.
- [7] Cardozo, E., Araujo Neto, J., Barza, A., Franca, A., and da Silva, F. 2012. SCRUM and Productivity in Software Projects: A Systematic Literature Review. In *Proceedings of 14th International Conference on Evaluation and Assessment in Software Engineering*. EASE'10. British Computer Society, Swinton, UK, 131-134.
- [8] Cohen, J. 1960. A coefficient agreement for nominal scales. *Educational and Psychological measurement* 20, 1 (Apr. 1960), 37-46.
- [9] Conboy, K. and Fitzgerald, B. 2010. Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *Transactions on Software Engineering and Methodology* 20, 1 (Jun. 2010), 2:1-2:30.
- [10] Diebold, P, Lampasona, C., and Taibi, D. 2013. Moonlight Scrum: An Agile Method for Distributed Teams with Part-Time Developers Working during Non-overlapping Hours. In *Proceedings of 8th International Conference on Software Engineering Advances*. ICSEA'13. IARIA XPS Press, Wilmington, DE, 318-323.
- [11] DSDM Consortium. 2008. *DSDM Atern - The Handbook*. Whitehorse Press Limited.
- [12] Dyba, T. and Dingsoyr, T. 2008. Empirical studies of agile software development: A systematic review. *Information Software Technology* 50, 9-10 (Aug. 2008), 833-859.
- [13] Higgins, J. and Green, S. 2006. *Cochrane Handbook for Systematic Reviews of Interventions 4.2.6*. The Cochrane Collaboration.
- [14] Highsmith, J.A. 2000. *Adaptive Software Development: A collaborative approach to managing complex systems*. Dorset House Publishing Co. Inc., New York, NY.
- [15] Khan, K., Ter Riet, G., Glanville, J., Sowden, A., and Kleijnen, J. 2001. *Undertaking systematic review of research on Effectiveness: CRD's guidance for carrying out or commissioning reviews*. CRD Report. NHS Centre for Reviews and Dissemination, York, UK.
- [16] Kitchenham, B. 2004. *Procedures for Performing Systematic Reviews*. Joint Technical Report. Keele University and National ICT Australia Ltd.
- [17] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (Jan. 2008), 7-15.
- [18] Landis, J. and Koch, G.: The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (Mar. 1977), 159-177.
- [19] Osterweil, L. 1987. Software processes are software too. In *Proceedings of the 9th International Conference on Software Engineering*. ICSE'87. IEEE Computer Society Press, Los Alamitos, CA, 2-13.
- [20] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. 2008. Systematic Mapping Studies in Software Engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. EASE'08. British Computer Society Swinton, UK, 68-77.
- [21] Rao K.N., Naidu G.K., and Chakka P. 2011. A study of the Agile software development methods, applicability and implications in industry. *Journal of Software Engineering and its Applications* 5, 2 (Apr. 2011), 35-46.
- [22] Salo, O. and Abrahamsson, P. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *IET Software* 2, 1 (Feb. 2008), 58-64.
- [23] Schwaber, K. 2010. *On Shoulders Of Giants*. Interview. Agile Collab. (<http://mattcallanan.blogspot.de/2010/02/ken-schwaber-on-scrum.html>)
- [24] Sutherland, J., and Schaber, K. 2013. *The Scrum Guide™ – The Definitive Guide to Scrum: The Rules of the Game*.
- [25] VersionOne. 2013. 7th Annual State of Agile Development Survey.
- [26] Xie M., Shen M., Rong G., and Shao D. 2012. Empirical studies of embedded software development using agile methods: A systematic review. In *Proceedings of the 2nd international workshop on Evidential assessment of software technologies*, EAST'12. ACM, New York, NY, 21-26.

# Document Information

Title: Agile Practices in Practice -  
A Mapping Study

Date: June, 2014  
Report: IESE-022.14/E  
Status: Final  
Distribution: Public Unlimited

Copyright 2014 Fraunhofer IESE.  
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.