International Journal of Software Engineering and Knowledge Engineering
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

A SYSTEMATIC MAPPING STUDY OF EMPIRICAL STUDIES ON THE USE OF PAIR PROGRAMMING IN THE INDUSTRY

JARI VANHANEN

Department of Computer Science and Engineering, Aalto University
PO BOX 15400, FI-00076, AALTO, Finland
jari.vanhanen@aalto.fi

MIKA V. MÄNTYLÄ

Department of Computer Science and Engineering, Aalto University
PO BOX 15400, FI-00076, AALTO, Finland
mika.mantyla@aalto.fi

Received
Revised
Accepted

Previous systematic literature reviews on pair programming (PP) lack in their coverage of industrial PP data as well as certain factors of PP such as infrastructure. Therefore, we conducted a systematic mapping study on empirical, industrial PP research. Based on 154 research papers, we built a new PP framework containing 18 factors. We analyzed the previous research on each factor through several research properties. The most thoroughly studied factors in industry are communication, knowledge of work, productivity and quality. Many other factors largely lack comparative data, let alone data from reliable data collection methods such as measurement. Based on these gaps in research further studies would be most valuable for development process, targets of PP, developers' characteristics, and feelings of work. We propose how they could be studied better. If the gaps had been commonly known, they could have been covered rather easily in the previous empirical studies. Our results help focus further studies on the most relevant gaps in research and design them based on the previous studies. The results help also identify the factors for which systematic reviews that synthesize the findings of the primary studies would already be feasible.

Keywords: Pair programming; systematic mapping study; industrial studies; empirical studies; gaps in research.

1. Introduction

Pair programming (PP) is a decades-old practice [22], but it got its explicit name and became better known when it was described as part of extreme programming (XP) [3]. A book focusing on PP [22] contained more guidelines for practicing PP and defined it as a practice where two programmers design, code and test software together at one computer actively communicating with each other. Based on many surveys (see sec. 4.4.5), PP is a practice that is currently used in the industry.

More than three hundred papers about PP can be found in scientific databases. Therefore, structuring and synthesizing the existing research has become important for focusing new primary studies adequately to increase the understanding of PP as efficiently as possible.

A few literature reviews on PP have already been published. Gallis et al. [9] proposed a framework for PP studies based on an unsystematic literature review of previous research on PP and related topics. Ally et al. [1] proposed small extensions and modifications to that framework based on another unsystematic literature review and their own studies. These frameworks provide a good checklist of relevant dependent and context variables related to studying PP, even though the majority of the PP studies have been published after these frameworks were developed.

Two systematic reviews on PP have been conducted. Salleh et al. [20] conducted a systematic literature review (SLR) of empirical studies on using PP as a pedagogical tool in higher CS/SE education. They analyzed 73 papers and focused on analyzing the effectiveness of PP, and how pair compatibility affects it. Hannay et al. [10] conducted a meta-analysis of 18 PP experiments of which 5 used professionals as subjects. They analyzed the differences between PP and solo programming (SP) regarding effort, duration and quality of code.

Due to their focus on CS/SE education [20] or on experiments [10] both of the previous systematic reviews have excluded almost completely those over 150 papers reporting data on PP from the industry. Due to their narrower focus on only a few factors of PP, which is sensible for making a good synthesis of the included papers, they have not covered at all data related to many other factors such as adopting PP, partners' communication, infrastructure, and effects to developers' knowledge. However, Hannay et al. [10] conclude that the effects of PP depend on the context, and emphasize the importance of increasing understanding of the moderating factors. They explicitly mention task complexity and developers' expertise as factors that have already been found to affect the effects of PP.

Thus, the previous reviews have not attempted to cover all research on PP. Considering the studies on PP in the industry, only five experiments and a few other papers of more than 150 papers were reviewed. In addition, the information on many potentially relevant factors of PP have not yet been covered at all in either industry or education context.

Therefore we conducted a systematic mapping study [16, 17] that considerably decreases the lack of coverage. We aimed to: 1) identify and describe all factors of PP that have been empirically studied in the industry and 2) to characterize the existing research separately for each factor of PP, and 3) to identify potential gaps in PP research.

We extract data from each included paper for each factor separately. The extracted data includes numerous objective research properties such as research approach and data type. Based on that data, we evaluate also the overall relevance of research for each factor in each paper.

We analyze quantitatively the amount of data in the various categories of each research property. Our qualitative analysis identifies the studied factors of PP, and organizes them into a new PP framework containing many detailed examples of the studied aspects of each factor. We also describe for each factor the most relevant studies and gaps in their research. The identification of these gaps in research allows targeting new primary studies appropriately. Based on the gaps in research and the relative importance of the factors, we choose four factors for which further studies would be most valuable. We propose how they could be studied better in the future.

As we have identified papers having the best available knowledge of each factor, it is easier to build the new studies on the existing knowledge and try to avoid the limitations of the previous studies. In addition, our results serve as a comprehensive and classified reference list of PP studies in the industry for anyone who is interested in some particular factor of PP. Synthesizing the results of the identified primary studies for each factor is out of scope for a mapping study, but our results serve as a very good basis for making such SLRs on any specific factor of PP studied in the industry.

2. Theoretical Background

Here we present the necessary background information to understand our study. It includes the systematic mapping study method, our organization of the various aspects of PP into a PP framework, and the viewpoints used for characterizing the PP research.

2.1. Systematic mapping studies

Systematic mapping studies [17], a.k.a. scoping studies [18], give an overview of a research area. According to Petersen et al. [17], they identify the quantity and type of research, and results available within the research area. Often they also show yearly publication trends and identify used publication forum [17]. Petticrew and Roberts [18] give a rather similar goal of determining the sort of studies, assessed outcomes and populations, as well as publication forums and databases indexing the studies. The results of a mapping study can be used; e.g., to identify suitable research areas for conducting SLRs or areas where further primary studies are more appropriate [16], and to identify relevant related research when conducting further primary studies [13].

Systematic mapping studies are similar to SLRs in the sense that both aim at providing a trustworthy, rigorous and auditable methodology to identify and analyze all available research relevant to a particular research topic [16]. However, there are some major differences in the scope and goals. Systematic mapping studies generally have a larger number of research questions, which are also broader [16]. They also cover more studies and present results as summaries of classifications of the included studies instead of synthesizing their results [16].

Systematic mappings studies and SLRs can also be thought of as two points on a continuum [17]. A systematic mapping study may also go deeper into the papers, e.g., due to poor abstracts, and become more like a SLR [17]. Our systematic mapping study is rather deep, but it is not a SLR, because we do not interpret or synthesize the results presented in the included papers. A tertiary study of systematic reviews in software engineering (SE) [7] actually found that half of the studies that referred to themselves as systematic reviews did not include synthesis and were, rather, mapping studies.

The guidelines for performing SLRs in SE by Kitchenham and Charters [16] discuss also mapping studies, but for mapping studies, their guidelines are applicable for the steps of searching and selecting studies. They give very little advice on how to undertake data extraction and analysis for a mapping study. However, deciding how to classify and categorize the included studies may be one of the major problems in a mapping study [5].

Only a high-quality systematic mapping study can support follow-on research [13]. It means that all the references must be cited, classification information for each study must be reported, and the study must be based on a stringent search process and a well-defined and reliable classification system [13].

2.2. PP framework

When preparing the review protocol we created a tentative PP framework based on 1) the existing PP frameworks [1, 9], 2) our long history in studying PP, and 3) reading more than a hundred papers on PP. We took the existing frameworks as the basis and modified them based on the papers we read and on our own ideas of a logical structure.

The purpose of our PP framework is to structure all the aspects of PP identified in this study in a way that facilitates understanding and communicating the relevant concepts of PP. Similar aspects are grouped under factors of PP. The factors include: 1) aspects that may be effects of PP, 2) aspects that may affect the realization of the effects of PP, and 3) any other relevant aspects of PP. Factors having a common theme are further grouped.

The factors in the framework have an additional purpose as they are used as target categories when all PP-related information in the primary studies is extracted. Therefore, we aimed at defining non-overlapping factors so that each piece of PP-related information from a paper could usually be associated to one factor, or two factors, if the information discusses their relationship. We aimed at defining detailed enough factors, because the results of the study are reported per factor. However, too-detailed factors were avoided in order to keep the factors non-overlapping and the data extraction granularity and effort reasonable.

During the systematic data extraction no totally new factors were added, but a few factors were merged or split in order to have a more logical structure for the framework. The final PP framework is presented in sec. 4.1, where concrete examples from the reviewed papers describe the aspects of PP belonging under each factor. An overview of the framework is given in Table 1.

Table 1. Overview of the final PP framework (see sec. 4.1. for details).

Theme	Factors
Preparations for PP	Adoption, Managing PP, Pair formation, Targets
Environment	Infrastructure, Development process
PP session	Partner combinations, Partners' roles, Communication, Breaks
Developer	Feelings of PP, Feelings of work, Knowledge of work, Characteristics
Utilization rate	Local amount, Prevalence
Main effects	Productivity, Quality

2.3. Characterizing the PP research

2.3.1. Research properties

Dozens of systematic mapping studies and SLRs in the SE domain have been conducted [7, 15]. They have classified primary studies according to numerous research properties, many of which can be used to classify studies also on any other SE topic such as PP. These generic research properties have included, e.g., research approach [e.g. 6, 12, 20], data collection method [e.g. 5], context such as academic vs. industry [e.g. 6, 12] or software application type [e.g. 6], publication forum [e.g. 5, 12, 14, 20], yearly distribution (e.g. 4, 12), authors [e.g., 15], and country [e.g. 4].

Table 2. Analyzed research properties[a].

Property	Level	Categories
Forum	Paper	1. Journal 2. Conference/workshop
Paper focus	Paper	1. PP (is one of the main focuses) 2. Other
Authors' role	Paper	1. Internal; i.e., at least one author worked in the studied organization 2. External; includes also visitors who worked at most a month in the studied organization
Research approach	Paper	1. Experiment 2. Survey 3. Case study; i.e., an in-depth, possibly multi method study of one or a few cases 4. Experience report; i.e., personal experiences from some case(s) without reporting the use of any scientific data collection method
Data collection method	Factor	1. Measurement; i.e., data collection where the error caused by subjectivity is small 2. Rigorous observation; e.g., audio/video tapes, or someone making rigorous notes on site 3. Interview 4. Questionnaire 5. Informal observation; e.g., an author was present, but the use of any data collection method is not reported 6. Defined; value(s) fixed by the authors; e.g., controlled variables in experiments
Data type	Factor	1. Quantitative 2. Qualitative
Discussion type	Factor	1. Comparative; i.e., evaluates how this factor was affected by some variation, or how variation of this factor affected some other factor. Comparative claims based on informal observation only are classified as descriptive. 2. Descriptive

[a] Table 2 includes only properties analyzed in this paper. Other extracted properties are: detailed forum, citation information, context (software was to be released vs. exercises, project vs. isolated tasks, team vs. isolated pair(s), subjects' PP experience), number of subjects, duration of the study, and extent of text of a factor.

The paper level research properties, such as research approach, are always identical for all factors discussed in a certain paper. Our study goes deeper than the level of a paper by classifying separately data on each factor (Table 1) studied in each paper. Thereby it also makes sense to analyze factor level research properties such as data collection method, which may vary among the factors discussed in a certain paper.

Term factor instance denotes the data extracted about a single factor from a certain paper. For each factor discussed in a paper, a separate factor instance is created. A factor instance contains a value for each research property. For example, a paper may provide two factor instances, one about productivity and another about quality. The content of the factor instances regarding the factor level properties may differ as there may be, e.g., measured, quantitative and comparative data about productivity factor, but only interview based qualitative and descriptive data about quality factor.

We characterize the research using the research properties listed in Table 2. We included most of the research properties used in the previous mapping studies and SLRs in the SE domain listed above, but also some additional ones that we considered important such as discussion type on the factor level. Most of the research properties have predefined categories (see Table 2) for the data extraction.

2.3.2. Relevance of a factor instance

In addition to characterizing the factor instances through the several objective research properties, we combine the research properties into a single overall relevance value for each factor instance. The relevance is considered especially from the viewpoint of how relevant the paper is to read by the PP researchers and practitioners interested in the scientific papers of a particular factor. Despite of potential limitations due to compressing many dimensions into one, the combined value gives at least some clear way to identify the most relevant papers about a specific factor and to compare the relevance of the research among the different factors.

Defining a fixed formula for calculating the relevance value based on the research properties would give perfect transparency and repeatability for evaluating relevance. However, it is very difficult to develop a formula that would contain justified weights for all the research properties and their categories, and that would also consider the possible mutual relationships between the categories. By considering the mutual relationships we mean, e.g., the danger of misjudgment if a rule states, e.g. that quantitative data is more valuable than qualitative data in all studies.

No generally accepted definition for such a multidimensional concept as overall relevance of research in the context of SE research exists. Neither an objective formula nor the use of subjective heuristics is a perfect way for the evaluation of the overall relevance of research. For example, experiments generally provide data with higher relevance than case studies. However, an experiment with many poorly controlled factors focused on some small detail does not necessarily produce clearly more relevant data on a factor than a more diverse case study.

We chose to use subjective evaluation by an experienced PP researcher (the primary author of this paper) because we believed it would lead to a smaller number of misjudgments than a formula defined by ourselves. Even if defining a formula, there would still be subjectivity in the form of giving the weights for the research properties and their categories.

The relevance of research is evaluated on a 5-point scale (Table 3). The categories of each research property in Table 2 are listed in the order of decreasing effect for relevance, even though the order should not be taken too literally. The following heuristics indicate higher relevance: 1) rigorous data collection method, 2) comparative data, 3) larger number of subjects 4) higher industrial realism of the PP usage context, 5) longer duration of the study, 6) larger amount of text about the factor, and 7) publishing in a recognized forum. Additionally, the number of aspects covered about a factor, and the general impression of the paper are also slightly considered.

Table 3. Scale for the relevance of a factor instance.

Category	Value of reading the paper	Example of studies
4 – Excellent	Must read	Measured, comparative data from a large experiment
3 – Good	Worth reading	Comparative data from a good case study
2 – Moderate	Likely to be worth reading	Lots of descriptive data from a case study
1 – Fair	May be worth reading	Descriptive data from an experience report
0 – Poor	Not worth reading	Uninteresting, general remarks of PP

2.3.3. State of research index

In order to compare the relative state of research among the factors some quantitative metric is needed. Therefore, we defined a *state of research index* for a factor. It sums up the numbers of factor instances of a factor giving exponentially more weight to higher relevance instances. Exponentially increasing weights emphasize that more advanced studies are considerably more important for advancing the state of research than less advanced studies. The index is calculated using Eq. (1). $R_{[i]}$ denotes the number of instances having relevance i.

$$\text{state of research index} = R_1 + 2R_2 + 4R_3 + 8R_{4.} \qquad (1)$$

2.3.4. Identifying the most relevant gaps in research of factors

The benefits of further studies vary between factors, because the gaps in their research and the importance of the factors vary. We use these two attributes to identify factors for which further studies would be most relevant.

We use three viewpoints for analyzing the degree of gaps in the research of a factor:

1. The state of research index.
2. The gaps related to the main research properties. For example, lack of measured, comparative, or quantitative data often indicate gaps in research.
3. The gaps in research coverage on some aspects of a factor.

We use three viewpoints for analyzing the importance of a factor:

1. The extent of the topic that the factor covers. The extent can be characterized, e.g., through the number and extent of the different aspects of PP belonging under a factor. For example, *development process* is a broader factor than *breaks* during a PP session.
2. The context factors of PP have a varying impact on other factors of PP. Unfortunately, if there are still large gaps in the research, the impact can only be speculated based on tentative studies or theoretical reasoning.
3. The different outcomes of PP vary in their importance for software development in general.

3. Research Method

We used the systematic mapping study method. Our protocol follows the guidelines for performing SLRs by Kitchenham and Charters [16], but for the data extraction and analysis we needed to adapt the guidelines. The protocol was developed through many piloting rounds. Below each section describes one of the six steps of the final protocol.

3.1. Research goals and questions

The research goal was to characterize the empirical PP research done in the industry. The research questions (RQ) and their rationales are listed in Table 4. The terminology used in the research questions was presented in section 2.

Table 4. Research questions.

Research question
RQ 1 What factors of PP and their detailed aspects have been studied?
RQ 2 What are the characteristics of PP research in general regarding the amount and types of research? **RQ 2.1** How many papers are there in each research property category[a]? **RQ 2.2** How many factor instances are there in each research property category?
RQ 3 What is the relative state of research among the factors of PP?
RQ 4 What are the characteristics of the most relevant studies for each factor of PP? **RQ 4.1** How many high relevance studies are there for each factor? **RQ 4.2** What are the study settings behind the most relevant studies?
RQ 5 What kind of gaps in research are there for each factor?
RQ 6 How could the most relevant gaps in research be filled?

[a] Only the paper level research property categories are considered in RQ 2.1.

3.2. Paper sources and search string

We searched papers from five publisher-specific and two generic databases listed in Table 5. The chosen publisher-specific databases cover most of the relevant forums we are aware of, but the two general databases contain a few additional, relevant journals and proceedings. In addition, we checked the reference lists of the included papers. We did not search for books, PhD dissertations or other theses. We searched only papers written in English. Publication years were not limited in the searches.

Generating the search string was easy, because pair programming became an established term before interest in PP research started to grow. This term became popular after its appearance in the XP literature in 1999 [3]. Because we wanted to have as complete coverage of scientific PP papers as possible, we did not add more specific limiting keywords such as trying to avoid PP papers from student context. Checking each paper manually against the paper selection criteria is more reliable for removing papers that discussed PP but did not belong within the defined scope of this study.

The search string was validated by checking that 150 PP papers collected by the first author over the years were found with the search string, if the paper existed in the used databases. These 150 PP papers used for validation were not previously classified in more detail, and thus validated the search string from the viewpoint of finding all papers discussing PP, not just those from the industry context. Based on the validation we added three synonymic keywords: "paired programming," "pair-programming" and "pairprogramming." Our final search string was the union of all the synonyms modified as required for each search engine (Table 5).

Table 5. Databases, searches and search results.

Database	Type	Search string	Fields	Hits[b]	Included papers[b]
ACM Digital Library	Publisher	"pair programming" "paired programming" "pairprogramming"	full text	532	21
IEEE Xplore	Publisher	"pair-programming" <in> pdfdata <or> "paired programming" <in> pdfdata <or> "pairprogramming" <in> pdfdata	full text	680	66
ScienceDirect	Publisher	"pair* programming"	full text	67	4
SpringerLink	Publisher	"pair* programming"	full text	402	30
Wiley Interscience	Publisher	"pair* programming"	full text[a]	9	0
SCOPUS	Generic	pair* pre/0 programming	all fields	559	51
Web of Science	Generic	"pair* programming"	topic & title	58	12

[a] The full-text search failed to process the full texts of many journals.
[b] These numbers contain the duplicates among the databases.

Because lots of information on PP is included as a side topic in agile software development papers, many relevant papers do not include our search string in their metadata. Fortunately, the publisher-specific databases allow one to apply the search string on the full text of the papers. For the generic databases, the search string was applied to relevant metadata fields (Table 5).

When comparing the results of the database searches to our previously collected set of PP papers, we found that certain relevant conferences were not indexed in the databases (PPIG 1999–2009, XP 2000–2002, XP Universe 2001) or the search engine failed to access their full texts (XP 2003 and XP/Agile Universe 2002–2003 in SpringerLink). For them we searched the search string automatically from the full texts if we had electronic proceedings available, and otherwise manually browsing the printed proceedings.

Together, the database searches found 123 of the 154 included papers, and the remaining 31 papers resulted from the manual searches. The number of hits and included papers for each database are listed in Table 5. The searches to ACM Digital Library, IEEE Xplore, and SpringerLink found together 115 of the 123 papers. Each of these databases contained a very distinct set of papers, because only two of the 115 papers were found in more than one of them. Thus, the exclusion of any of these three databases would have left many papers out of the review. The search to SCOPUS found 51 of the 123 papers, including all the remaining eight papers that were not found in the searches to the databases of ACM, IEEE and Springer. Therefore, searching only these four databases of the seven chosen ones would have been enough.

Even though SCOPUS indexes a large proportion of the papers existing in the databases of ACM, IEEE and Springer, only 41% of the papers included from these three databases were found in our search to SCOPUS because SCOPUS contains only paper metadata. Thereby, we estimate that the database searches would have missed at least half of the now included papers, if we had not applied the searches to the full texts of the papers when possible.

3.3. Selection of papers

The criteria used when selecting the papers are described as exclusion criteria in Table 6 in the order they were applied to each hit. A paper found in the searches was included if none of the exclusion criteria applied to it.

Table 6. Paper exclusion criteria.

ID	Exclusion criteria
L	The paper is not written in English.
F	The paper is not published in a scientific journal, conference or workshop: • all journals and proceedings indexed in the selected databases were considered scientific • PhD dissertations and other theses were excluded • books other than proceedings were excluded
T	The paper is not a research article: • e.g., editorial, letter to the editor, book review
M	The paper could not be acquired in its entirety.
C	The PP content is non-existent or very poor: • does not include authors' own data or ideas on PP, e.g., only a summary from literature, or • is very general and worth the relevance of 0 (table 2), e.g., "PP was used in the project"
D	The PP content is relevant only for distributed PP.
P	The PP content is not about the use of PP by professional developers.
E	The PP content is not empirical.

Firstly, we had three criteria related to the language, forum and type of a paper. Secondly, the paper had to contain *empirical data* on some factor(s) in the context of *professional* developers working as *co-located* pair(s) together with the same task consisting of some of the following *activities*: software analysis, design, programming, or programming based testing. These four content-related criteria can be explicated in more detail as follows:

1. *Empirical* data refers to data collected or observed by the authors including also expert opinion.
2. *Professionals* refer to developers who have work experience and participate in the study in the role of a professional either in a real or an artificial software development context. Studies where professionals act in a student role on a university course are excluded.

3. *Co-location* excludes papers that discuss PP only from the viewpoint of partner distribution such as IT tool support for distributed PP.
4. *Activities* scoped the study so that:
    - testing activities that involve programming were included
    - test execution or static methods, such as formal code reviews, were excluded
    - end-user programming; e.g., spreadsheet development was excluded

The paper selection process steps are listed in Table 7. The first author performed the database searches on January 7, 2010, resulting in 2307 hits, of which 1749 were unique. Exclusion criteria L, F, and T were applied on the paper metadata resulting in 1478 papers. The remaining exclusion criteria were applied after browsing the full papers, and left us with 130 papers. The manual searches to the relevant proceedings missing from the databases added 33 papers and checking the reference lists of all the included papers added four papers. Later, we removed 13 papers due to being duplicates, i.e. some better paper of the same study was included. The final number of papers was 154.

The first author conducted the paper selection as a whole. In addition, the second author applied the exclusion criteria for a random subset of 88 hits; i.e., 5% of the unique hits. For only 3.4% (3/88) of the hits, the exclusion decision differed between the authors. The 95% confidence interval for a different decision is from 0 to 7.1%. However, a discussion between the authors resulted in classifying the three papers as so poor that none of them were worth including, meaning that based on the validation sample we can estimate that practically no papers were erroneously excluded during the paper selection.

Table 7. Paper selection process steps.

Step	Papers included or excluded	Papers remaining
1. Database searches	+2307	2307
2. Exclusion of duplicate hits	-558	1749
3. Exclusion due to language (L), forum (F) or type (T)	-271	1478
4. Exclusion due to missing papers (M)	-25	1453
5. Exclusion due to poor PP content (C)	-1084	369
6. Exclusion due to distributed PP content only (D)	-39	330
7. Exclusion due to non-professionals using PP (P)	-167	163
8. Exclusion due to non-empirical nature (E)	-33	130
9. Manual searches to certain proceedings	+33	163
10. Checking the reference lists of the included papers	+4	167
11. Exclusion due to duplicated PP content	-13	154

3.4. Study quality assessment

In SLRs it is critical to assess the quality of the primary studies. The assessment results are used to exclude poor studies or to refine the analysis of the primary studies [16]. In our study, the characterization of research can also be seen as study quality assessment. It was done separately for each factor in each paper. However, we did not use the results of the characterization of research for excluding studies with lower quality, because the purpose was to achieve as broad coverage of papers as possible regardless of their quality.

We piloted a detailed, paper-level, overall study quality evaluation using nine criteria based on those used in [8]. The criteria seemed to be very relevant, but similarly to [8], we found their quantitative evaluation difficult, which led to low inter-rater agreement. The difficulty, the required effort, and our focus on characterizing the research on the factor level led us to exclude this kind of study quality evaluation.

3.5. Data extraction

3.5.1. Data extraction process

All PP related data in each paper was classified according to the factors (Table 1) and the categories of the research properties (Table 2). Two additional categories were available to all properties. The *unknown* category was used if some data could not be extracted. The *mixed* category was used, if multiple categories applied equally.

Each paper was browsed either line-by-line or by searching the occurrences of word "pair" until some PP-related text was found. The latter way was used only for some of the papers that did not focus on PP and thereby discussed PP only in a small part of the paper. When relevant text was found, it was classified by creating a new factor instance in an Excel sheet. If more text on the same factor followed in the same paper, the same factor instance was refined.

We ignored general remarks such as "PP was good" or that PP was used for the "default case"; i.e., for programming activity and working whole tasks together. We also ignored text referring to other papers.

3.5.2. Special issues in extracting factor instances

If some text discussed explicitly the relationship between two factors, a separate factor instance was created for both factors. For example, text on the effect of task complexity on the productivity of PP is related to both *productivity* and *targets* of PP.

A factor that may often overlap with other factors is *adoption* of PP, because difficulties in, and aids and reasons for, it are typically closely related to other factors. If some text on some other factor than *adoption* also explicitly discussed the adoption point of view, it was extracted under *adoption* in addition to the particular factor. For example, text on a limited number of workstations as an aid for adopting PP created factor instances for both *adoption* and *infrastructure*.

3.5.3. Ensuring the consistency of the data

The development of the data extraction process involved lots of preparation and piloting. The first author read more than a hundred previously found PP papers to get an overview of their content. He also extracted data from a random sample of twenty papers. Piloting also involved another reviewer who extracted data from ten papers of the same sample. The main focus in piloting was ensuring that both reviewers identify the same factors from the papers and classify their properties equally.

During the review, the first author processed all papers. After extracting all data of a paper, he rechecked the data for correctness. After all the papers had been processed, he rechecked, one factor at a time, the relevance values of all instances of that factor to ensure the uniform usage of the relevance scale among the papers.

The second author processed a random sample of 17 papers; i.e., 11% of the 154 papers. The validation of the data extraction process was done by analyzing the differences between the authors in identifying and classifying the factor instances.

The first author identified 61 instances from the validation sample, and the second author 62 instances. Compared to the first author, the second author had identified nine new and 53 same instances, and missed eight instances. For four of the 53 same instances, the second author had classified the same piece of text under a different factor.

Table 8. Estimated error rates in the identification and factor classification of the factor instances.

Relevance	Missed instances	Instances with an uncertain factor classification
4 – excellent	0%	0%
3 – good	0%	0%
2 – moderate	0%	0%
1 – fair	14.8% +/- 8.8%[a]	7.6% +/- 6.8%[a]

[a] 95% confidence intervals based on the 11% validation sample of the 154 papers, which contained 61 of the 608 instances extracted from all the 154 papers.

The estimated errors in the identification and factor classification of the factor instances during the data extraction are summarized in Table 8. The estimates are based on the validation sample. Because all the differences between the authors were related to instances having only fair relevance, we estimate that there are no errors related to the instances having higher than fair relevance. For the instances having fair relevance, we estimate that 1) the use of two reviewers for all papers would have increased the total number of identified instances by 14.8% (=9/61), and 2) for 7.6% (=4/53) of the instances their factor classification may be unreliable.

Regarding the classification of the instances into the research property categories, the authors differed only for at most a few percentages of the instances for each research property. Regarding the relevance classification the authors gave a different value for 28% of the instances, but in all cases only by one level.

3.6. Data analysis

We conducted both quantitative and qualitative analysis. The quantitative analysis analyzed the distributions of the paper and factor instance counts among the research property categories; and, for the instance counts, also the distributions among the factors. The qualitative analysis identified the studied aspects of PP and organized them as factors and concrete examples into our PP framework. It also described the most relevant studies related to each factor, and gaps in the research of each factor.

4. Results and Discussion

Here we answer all the research questions and also discuss the implications and possible explanations of the results. When possible, we compare our results to prior similar review studies in the SE domain. In contrast to many other systematic reviews, our searches were applied also to the full texts of the papers, and we had very loose criteria for including a paper. Therefore, it is likely that we included a larger proportion of all papers discussing the specific research topic, but our set of papers contains a higher proportion of papers with low quality. This must be noted in any comparison made.

Section 4.1 presents the PP framework. Section 4.2 characterizes the PP research in general. Section 4.3 shows the state of research among the factors. Section 4.4 characterizes the research for each factor separately. In section 4.5 we propose further studies for filling the identified gaps in research.

4.1. PP framework

Here we answer RQ 1 by presenting the PP framework (Table 9). The identified factors are organized under 6 themes and described using concrete examples from the included papers. The framework does not include relationships among the factors, such as task complexity affecting productivity.

The *preparations for PP* factors are related to the initial adoption of PP and also to the recurrent preparations required when performing PP. The *environment* factors involve the software and hardware infrastructure, and the encircling software development process with all its practices. The *PP session* factors are directly related to working in PP sessions. The developer factors are related to the properties of a developer. *Feelings of PP* is a separate factor due to its significance for PP even though it can be seen as a part of *feelings of work*. The *utilization rate* factors are related to the amount of using PP. *Local amount* means the use of PP within a single case, whereas *prevalence* refers to the extent that PP is generally used in various organizations. The *main effects* factors contain two of the most typically affected project attributes: *productivity* and *quality*. Other affected factors are included under the other themes.

Our framework includes all the aspects of PP from the existing PP frameworks [1, 9] with some changes in their naming and grouping. For example, the role of PP in decreasing software development risks is not a factor in our framework, because almost any benefit of PP can also be seen as a way to avoid some risk. *Breaks* and *prevalence* are new factors compared to the previous frameworks. Also, the examples in our framework include many additional or more detailed aspects of PP. However, our framework may still lack some relevant aspects of PP, because we scoped out papers from student context as well as theoretical and unscientific papers.

Table 9. PP framework.

Theme	Factor	Examples
Preparations for PP	Adoption	**Difficulty level**: compared to other (XP) practices [P101], length of learning time [P143]. **Difficulties**: organizational culture [P122], management resistance [P12], evaluation of personal contribution [P141], lack of partners due to 1) different work times [P6], 2) small team [P57] or 3) distributed team [P126]. **Aids**: PP guidelines [P128], PP training [P57], PP champion [P145], alternative for reviews [P107], enforcement [P128], limited number of workstations [P143]. **Reasons**: many of the expected benefits listed under other factors.
	Managing PP	**Deciding on PP use**: mandatory to use [P87], who decides [P145], when decided [P145]. **Assigning PP tasks**: practices such as a pair chooses in daily meeting [P119]; task ownership options such as individual/pair [P14] or owned by workstation [P80]; task ownership problems such as no accountability [P49]. **Scheduling PP**: practices such as allocating time for PP [P145], problems such as experts work alone before DLs [P54], common time not found [P145] or working away from office [P141], accuracy of estimating PP tasks [P144]. **Degree of collaboration**: whole task together (default case), a task is split and both work alone for a while [P54], only one person works for a while [P94], synchronization after working alone [P107].
	Pair formation	**Initial pair formation**: organized by managers [P145], self-selected [P131], ad-hoc [P28], based on required skill set [P107]. **Partner rotation**: frequency [P14], who continues an unfinished task [P14].
	Targets	**Activities**: programming (default case), specification [P113], design [P94], refactoring [P9], TDD [P100], debugging [P9]. **Situations**: project initiation [P65], new developers join the team [P47], evaluating employee candidates [P53]. **Characteristics of targets**: task complexity [P7].
Environment	Infrastructure	**Hardware**: big screen [P145], dual keyboard [P28], two workstations [P145], whiteboard [P145], white noise generator [P151]. **Software**: large fonts [P43], standardized tools [P145]. **Furniture**: shape of desks [P131], whiteboard [P94]. **Office layout**: separate PP room [P145], open office [P145], cubicles [P84]. **Noise in workspace**: awareness [P30], disturbance [P145].
	Development process	**PP facilitates other practices**: TDD [P143], coding standard [P143], refactoring [P144], collective ownership [P43]. **PP replaces other practices**: code review [P57]. **PP disturbs other practices:** individual performance evaluation [P54] **Other practices facilitate PP**: e.g., test-driven approach [P38], collective ownership [P8], planning game [P8]. **Discipline with the process**: process conformance [P144], concentration on work [P43].
PP session	Partner combinations	**Combinations**: personality [P148], work expertise [P7], PP experience [P22], age [P87]. **Viewpoints**: frequency of combinations [P47].
	Partners' roles	**Characteristics of roles**: a leader [P94], keyboard possession [P28], level of thinking [P28]. **Switching the roles**: frequency [P121].
	Communication	**Content**: abstraction level [P20], representations used [P44], value, e.g. usefulness [P14]. **Amount**: amount of utterances [P20]. **Issues**: solving disagreements [P128], flow and mental blocks [P43], speed of work such as slow enough for the junior pair [P115] or typing speed [P91]. **Partners' relationship**: getting to know the partner [P81], courage to criticize the partner's work [P139].
	Breaks	**Types of breaks:** intrusions, distractions and breaks [P29] **Viewpoints**: amount [P29], reasons [P144].

Developer	Feelings of PP	**Feelings**: resistance [P128], satisfaction [P38], enjoyment [P114], "general" feelings [P145].
	Feelings of work	**PP affects feelings of work**: team spirit [P107], enjoyment [P104], enthusiasm [P43], exhaustiveness [P8], threatening [P3], pair pressure [P154]. **PP is affected by feelings of work** [P126].
	Knowledge of work	**PP affects knowledge of work**: developed software [P12], tools [P144], work practices [P144], or domain [P107], general knowledge of a new developer [P152]. **PP is affected by knowledge of work**: PP ability [P22], work experience [P33].
	Characteristics	**Demographics**: nationality [P109]. **Psychosocial factors**: personality [P37], self-esteem [P128], communication skills [P12], conflict handling style [P38].
Utilization rate	Local amount	**Dimensions**: realized proportion of development work [P33], realized amount [P145], proposed amount [P144], desired amount [P145].
	Prevalence	**Breadth of use**: world-wide [P35], nationally [P125], in embedded sw domain [P124], in a global company [P13]. **Depth of use**: use on an ordinal scale (systematically–never) [P124], used vs. not used [P35], using or planning to use [P13]
Main effects	Productivity	**Dimensions**: effort/duration [P7], scope [P107], lines of code [P65].
	Quality	**Code**: defects [P7], readability [P104], comment ratio [P65]. **Design**: understandability [P143], quality [P94]. **General**: confidence in results [P104].

4.2. Characteristics of PP research in general

Here we answer RQ 2 by analyzing how the papers (Table 10) and factor instances (Total columns in Table 11) distribute among the categories of each research property. The categories were described in Table 2.

For the sake of completeness, the numbers of factor instances are included also for the paper level research properties in Table 11. The distributions of the papers and factor instances among the categories differ slightly because the factor instances do not distribute evenly among the papers. Below, we refer to the paper distributions when discussing the paper level research properties, and to factor instance distributions when discussing the factor level research properties.

The papers are listed in Appendix A, and they are referenced by P1–P154. The classifications of the papers according to the paper level research properties are listed in Appendix B[a]. Appendix C[b] contains all the instances.

In total we found 154 papers containing some relevant information on one or more factors. The median of the factor instances per paper is four and the maximum 15. The total number of the instances is 608. It does not include the instances, which were evaluated as poor in the relevance evaluation.

The total number of papers and instances can be considered as a rather large body of evidence. However, the situation is worse when the distribution of this data among the research property categories is considered as will be discussed below.

[a] Appendix B is available at http://dx.doi.org/10.6084/m9.figshare.801065

[b] Appendix C is available at http://dx.doi.org/10.6084/m9.figshare.801065

Table 10. Number of papers per paper level research property.

Property		Total		Forum		Paper focus		Authors' role			Research approach			
	Category	All (%)	All (n)	Journal	Conference	PP	Other	Internal	External	Unknown	Experiment	Survey	Case study	Exp. report
Year	<2001	4.5	7	2	5	4	3	3	3	1	1	0	2	4
	2001	5.8	9	3	6	1	8	8	1	0	0	0	1	8
	2002	8.4	13	0	13	3	10	8	5	0	1	4	1	7
	2003	11.0	17	4	13	2	15	12	4	1	1	1	3	12
	2004	14.9	23	4	19	6	17	11	12	0	0	1	11	11
	2005	11.7	18	0	18	5	13	7	10	1	0	3	11	4
	2006	13.6	21	5	16	7	14	10	11	0	2	5	8	6
	2007	13.6	21	3	18	7	14	9	12	0	3	2	8	8
	2008	9.7	15	5	10	5	10	4	11	0	1	3	8	3
	2009	6.5	10	1	9	3	7	6	3	1	2	0	3	5
Forum	Journal	17.5	27											
	Conference	82.5	127											
Paper focus	PP	27.9	43	10	33									
	Other	72.1	111	17	94									
Authors' role	Internal	50.6	78	11	67	14	64							
	External	46.8	72	15	57	28	44							
	Unknown	2.6	4	1	3	1	3							
Research approach	Experiment	7.1	11	5	6	10	1	1	10	0				
	Survey	12.3	19	5	14	3	16	0	19	0				
	Case study	36.4	56	8	48	21	35	15	40	1				
	Exp. report	44.2	68	9	59	9	59	62	3	3				

4.2.1. Yearly distribution

The yearly number of papers increased steadily from 2000 to 2007, except for year 2004 being a peak with 23 papers. Starting from 2008, the yearly number of papers started to decrease. The decrease is most notable for *infrastructure*, *adoption*, *feelings of work* and *communication*.

The data did not reveal any particular reason for the decrease such as the ceasing of some previously active forum. Also, we ensured later that most of the publications from 2009 were already in the databases when we made the searches in January 2010. Thus, the decline is likely to indicate decreased research activity about PP.

4.2.2. Forum

The forum is journal for 27 papers (18%), and conference or workshop for 127 papers (82%). The most common forum is the XP conference (31 papers). Next comes the AGILE conference (19 papers), followed by its predecessors Agile Development Conference (10 papers) and XP Universe (10 papers). The fifth forum is IEEE Software (7 papers), and the remaining 51 forums each contain 1–4 papers.

Table 11. Number of factor instances per research property and per factor.

Property	Category	Total		Factor																	
				Preparations for PP				Environ-ment		PP session				Developer				Utilization rate		Main effects	
		All (%)	All (n)	Adoption	Managing PP	Pair formation	Targets	Infrastructure	Development process	Partner combinations	Partners' roles	Communication	Breaks	Feelings of PP	Feelings of work	Knowledge of work	Characteristics	Local amount	Prevalence	Productivity	Quality
Year	<2001	4.9	30	2	2	1	0	1	0	3	2	6	0	3	2	3	0	0	0	2	3
	2001–2003	25.7	156	15	9	10	10	9	5	7	4	7	2	10	10	19	3	9	3	12	12
	2004–2006	38.5	234	20	17	14	20	16	10	11	9	16	6	8	14	23	5	11	5	15	14
	2007–2009	30.9	188	12	11	15	14	7	7	14	9	9	3	10	8	19	2	14	5	12	17
Forum	Journal	14.8	90	6	6	2	5	5	4	3	1	5	1	3	8	9	1	6	4	8	13
	Conference	85.2	518	43	33	38	39	28	18	32	23	33	10	28	26	55	9	28	9	33	33
Paper focus	PP	39.0	237	13	20	13	15	12	8	15	14	16	6	9	13	25	8	10	1	21	18
	Other	61.0	371	36	19	27	29	21	14	20	10	22	5	22	21	39	2	24	12	20	28
Authors' role	Internal	53.3	324	27	24	24	22	17	13	17	10	15	5	17	18	43	5	17	0	21	29
	External	44.2	269	19	14	16	20	16	9	18	14	21	6	13	15	19	5	17	13	19	15
	Unknown	2.5	15	3	1	0	2	0	0	0	0	2	0	1	1	2	0	0	0	1	2
Research approach	Experiment	6.3	38	1	1	0	3	1	2	4	1	3	0	2	1	2	3	0	0	7	7
	Survey	10.5	64	8	2	2	2	1	2	3	1	1	1	4	4	6	2	1	13	7	4
	Case study	42.1	256	16	21	18	19	19	10	17	15	20	6	12	17	21	0	20	0	11	14
	Exp. report	41.1	250	24	15	20	20	12	8	11	7	14	4	13	12	35	5	13	0	16	21
Data collection method	Measurement	6.1	37	0	0	0	3	0	1	1	1	0	0	0	0	0	0	7	0	16	8
	Rigorous obs.	13.8	84	0	7	11	3	9	2	3	9	19	4	2	7	3	0	3	0	1	1
	Interview	10.2	62	8	3	1	4	5	4	4	1	2	0	5	6	8	2	1	0	4	4
	Questionnaire	14.6	89	8	3	2	3	2	2	6	2	2	2	9	5	8	4	7	13	7	4
	Informal obs.	48.5	295	32	22	25	27	15	12	16	10	14	4	14	13	38	3	13	0	12	25
	Defined	2.0	12	0	2	0	2	1	1	3	1	1	0	0	0	0	1	0	0	0	0
	Mixed	3.9	24	1	1	1	1	1	0	2	0	0	1	1	2	5	0	3	0	1	4
	Unknown	0.8	5	0	1	0	1	0	0	0	0	0	0	0	1	2	0	0	0	0	0
Disc. type	Comparative	13.3	81	3	2	2	4	2	3	7	5	5	1	3	2	7	0	3	1	15	16
	Descriptive	86.7	527	46	37	38	40	31	19	28	19	33	10	28	32	57	10	31	12	26	30
Data type	Quantitative	19.9	121	6	1	1	6	0	3	9	5	6	1	8	2	10	3	16	13	20	11
	Qualitative	76.3	464	43	34	36	37	32	18	26	18	32	9	23	30	52	7	18	0	18	31
	Mixed	3.8	23	0	4	3	1	1	1	0	1	0	1	0	2	2	0	0	0	3	4
Total	All (n)	-	608	49	39	40	44	33	22	35	24	38	11	31	34	64	10	34	13	41	46
	All (%)	100	-	8.1	6.4	6.6	7.2	5.4	3.6	5.8	3.9	6.3	1.8	5.1	5.6	10.5	1.6	5.6	2.1	6.7	7.6

The proportion of journal papers is low considering that two other reviews in the SE domain that did not use any quality criteria in the study selection [2, 19] both report about 50% proportion of journal papers. The low proportion may indicate lower maturity of research compared to other SE topics. Furthermore, there does not seem to be any increase in the proportion of journal articles over time.

Another explanation may be that agile/XP focused forums are a natural place for publishing PP studies. However, there are no scientific agile/XP journals, but well-known agile/XP conferences exist, and they were among the most common forums.

4.2.3. Paper focus

The number of papers focusing on PP is 43 (28%). A further analysis of the other papers (72%) shows that 96% of them discuss agile software development from some point of view. Typically they discuss PP among the other practices of XP. Until 2003 PP was discussed almost only in such papers.

Papers not focusing on PP contain a smaller proportion of higher relevance factor instances, but excluding them would have decreased the total amount of instances considerably, including 19% loss of excellent or good instances. Therefore, ensuring their inclusion by applying the database searches to full texts was worth the effort.

4.2.4. Authors' role

In 72 papers (47%) all authors were external; i.e. did not work in the studied organization, even though in some of these papers external author(s) were present; e.g., as observers. In the other half of the papers (51%) there was an internal author involved.

In papers focusing on PP, all authors were much more often external (65%, i.e. 28 of 43 papers) than in the other papers (40%, i.e. 44 of 111 papers). In 91% (62 of 68) of the experience reports there was an internal author involved. Together with 27% (15 of 56) of the case studies, these two research approaches contributed all but one paper where there was an internal author involved.

We are not aware of other SLRs or mapping studies in the area of software engineering that report data about authors' role. From the viewpoint of study quality, the authors' role is a two-fold topic. On the one hand, deep involvement increases the authors' knowledge on the topic, but on the other hand, it may threaten objectivity.

4.2.5. Research approach

Experience report is the most common (44%), but least rigorous research approach, and over 90% of their instances have at most moderate relevance. An explanation for the high proportion of the experience reports may be the simplicity of writing them about PP or XP than about more complicated SE topics. The recent decrease in the proportion of experience reports may indicate decreasing interest of the practitioners to report their experiences on XP or PP anymore as the topics age.

However, a mapping study [11] that focused also on real-life data on a popular topic in industry (SCRUM in global software development) reports as high as 80% proportion of experience reports. In addition, it used certain quality criteria for paper inclusion.

Experiments are the most rigorous research approach, but their proportion is only 7% (11 papers). The situation is much better for the papers focusing on PP, where the proportion is 23%. Experiments are most frequently related to *productivity* and *quality*.

As high a proportion of experiments as 59% is reported in a PP SLR that focused on papers in student context [20]. Even though that study may have had stricter inclusion criteria considering it lacked an experience report category, the difference to our results is huge. Conducting experiments with professionals is naturally much more expensive and difficult than with students. However, the small increase in the proportion and absolute amount of experiments in recent years may indicate some maturation of the PP research.

4.2.6. Data collection method

Informal observation is clearly the most common data collection method, contributing 49% of the instances. It is used mainly in the experience reports. Rigorous observations contribute 14% of the instances, and are almost solely used in the case studies. Rigorous observations are clearly most often related to *communication*, but often also to *pair formation*, *infrastructure* and *partners' roles*.

Questionnaires contribute 15% of the instances, and are naturally very often used in the surveys, but also in the case studies. Interviews contribute 10% of the instances and are used in all research approaches except experiments. Measurements contribute 6% of the instances, mostly from the experiments and case studies, both contributing about half of the instances. Measurements are mostly about *productivity*, *quality* or *local amount*.

4.2.7. Discussion type and data type

The discussion type is descriptive for 87% and comparative for 13% of the instances. The comparative instances are clearly most frequently related to *productivity* and *quality*.

The data type is qualitative for 76% and quantitative for 20% of the instances. The quantitative instances are usually about *productivity*, *local amount*, *prevalence* or *quality*.

4.2.8. Study context

We included only papers discussing PP used by professional developers. Therefore, at least 83% of the papers came from a typical, industrial software development context, where the pairs 1) worked within a team larger than two developers 2) performed tasks related to a large project, and 3) developed the software for real use instead of it being an exercise. In 6% of the included papers none of these contextual aspects was reported.

In at least 70% of the papers, the subjects had PP experience i.e., at least 2/3 of them had more than 40 hours of PP experience by the end of the study. In 25% of the papers PP experience was not reported or implicitly obvious. This deficiency calls for improvement, because PP experience can be an important context factor. We are not aware of other SLRs or mapping studies in software engineering domain that report data about the subjects' experience of the studied method or tool.

The experiments differed from the other studies. They were mostly conducted with novice pair programmers working as isolated pairs who performed isolated tasks not related to software to be delivered. This indicates the cost and difficulties in making experiments in realistic, industrial contexts.

4.2.9. Summary

For all research properties, the proportions of instances in categories indicating higher relevance research such as journal papers (15%), quantitative data (20%), comparative discussion (13%) and data collected using measurement (6%) or rigorous observation (14%) are low. The high proportion of experience reports produced lots of instances in the lower relevance categories such as qualitative data (87%), descriptive discussion (77%) and informal observation (49%). However, the absolute amount of higher relevance research is also rather low, and it focuses mainly to a few factors only.

4.3. State of research among the factors

Here we answer RQ 3 by analyzing the relative state of research among the factors based on the state of research index (see sec. 2.3.3). The index varies a lot among the factors (Table 12). The distribution of the instances is biased towards the lower relevance categories, and half of the factors have no excellent instances.

Table 12. Factors ranked based on the state of research index.

Factor	Rank	State of research index	Relevance of factor instances					Papers with excellent or good factor instances	
			1- Fair	2- Moderate	3- Good	4- Excellent	Total (n)	3- Good	4- Excellent
Communication	1	83	27	2	5	4	38	[P28,P39,P44,P45,P120]	[P20,P21,P24,P148]
Knowledge of work	2	80	50	13	1	0	64	[P144]	-
Productivity	3	73	23	13	4	1	41	[P65,P96,P104,P121]	[P7]
Quality	4	72	32	10	3	1	46	[P65,P96,P104]	[P7]
Local amount	4	72	20	8	3	3	34	[P65,P143,P151]	[P33,P47,P145]
Adoption	6	67	39	6	4	0	49	[P101,P125,P128,P145]	-
Targets	7	63	35	6	2	1	44	[P21,P121]	[P7]
Partner combinations	8	59	25	7	1	2	35	[P24]	[P7,P148]
Pair formation	9	58	24	15	1	0	40	[P14]	-
Managing PP	10	48	34	3	2	0	39	[P14,P145]	-
Prevalence	10	48	0	6	5	2	13	[P12,P13,P35,P63,P92]	[P124,P125]
Partner’s roles	10	48	18	1	3	2	24	[P21,P24,P121]	[P20,P28]
Infrastructure	13	46	22	10	1	0	33	[P145]	-
Feelings of work	14	44	28	4	2	0	34	[P8, P144]	-
Feelings of PP	15	42	24	5	2	0	31	[P12,P145]	-
Development process	16	28	18	3	1	0	22	[P144]	-
Breaks	17	21	7	3	0	1	11	-	[P29]
Developer’s character.	18	16	4	6	0	0	10	-	-
Total (n)			430	121	40	17	608		
Total (%)			70.7	19.9	6.6	2.8	100		

Communication ranks first, mostly due to many higher-relevance instances originating from the use of rigorous observation as the data collection method. However, five high-relevance instances are based on only two data sets that were analyzed from somewhat different viewpoints in five papers (see section 4.4.3).

The next factors in the ranking, k*nowledge of work*, *productivity* and *quality,* are related to the expected benefits of PP [22]. Therefore, their high ranking is not a surprise. *Knowledge of work* ranks high due to the high number of instances, even though they are generally of low relevance. *Productivity* and *quality* rank high mostly because they have been studied using experiments that produce high relevance instances. *Local amount* ranks next probably because it is rather easy to measure effort spent on PP quantitatively.

In the middle pack of the ranking there are many factors that are related to the context in which PP is used such as *targets of PP*, *partner combinations*, *pair formation*, and *infrastructure*. As stated in [10], context factors may have an important role in the realization of the effects of PP. Therefore, increasing research on them would be justified.

Developer's characteristics, *breaks*, and *development process* rank lowest. Not a single paper provides excellent or good instances for *developer's characteristics*. These factors are quite detailed aspects of PP, which may explain their low ranking.

4.4. Characteristics of the research per factor

Here we characterize the research separately for each factor using three viewpoints: 1) number of high relevance instances (RQ 4.1), 2) overview of the studies behind the most relevant instances (RQ 4.2), 3) main gaps in research (RQ 5).

There are 15 papers that contain excellent or good instances of at least two factors. These papers are summarized in Table 13 and only referred under all the related factors. The complete list of papers containing instances of each factor is in Appendix C[c].

4.4.1. Preparations for PP

Adoption ranks 6th according to the state-of-research index with no excellent and four good instances. Two good instances are surveys with almost a hundred respondents in each. The first [P101] reports data about the level of difficulty of adopting XP practices including PP. The second [P125] (Table 13) reports frequencies of reasons for not adopting PP. The third good instance is the case study [P145] (Table 13) about the adoption of PP. Fourth is an experience report that discusses obstacles and advice for adopting PP in an XP team that resisted the adoption of PP [P128].

Two moderate instances contain data comparing the difficulty of adopting PP to some other practices [P143,P72]. The other instances are mainly short, qualitative and descriptive remarks about certain problems faced and tactics used in individual cases.

The main gap in research is the lack of measured or rigorously observed instances about any aspect of adoption such as required learning time or potential difficulties. Also, there are only three comparative instances, of which two [P68,P85] compare mandatory and voluntary adoption, and third [P145] analyzes more variations in the adoption.

Managing PP ranks 10th with no excellent and two good instances. The first good instance [P14] (Table 13) analyzes variations in task ownership (individual vs. team) and task assignment (assigned vs. chosen, or assigned just-in-time vs. per iteration) regarding their effects to productivity. The second [P145] (Table 13) identified problems and solutions in organizing PP, and inquired developers' feelings of organizing PP.

Almost all the other instances are descriptive, qualitative and from case studies or experience reports. Typically they discuss briefly how some aspect of task management was done such as assigning tasks to pairs in a daily meeting, or what problems occurred, such as scheduling and resourcing problems.

The main gap in research is that there are no measured instances of any aspect of managing PP such as the effect of PP on task effort estimation accuracy. Also, there are only two comparative [P14,P38] and one quantitative instance [P38]. In [P38] the studied aspect was the degree of collaboration, which meant working alone vs. together after some preparation for coding work had been done together.

[c] Appendix C is available at http://dx.doi.org/10.6084/m9.figshare.801065

Table 13. Papers with several excellent or good instances.

Paper	Description	Factors
P7, P148	An experiment where 295 hired consultants performed artificial programming tasks for eight hours. In [P7] differences between PP and SP in effort and quality were studied using task complexity and partner combinations as moderating factors. In [P148] the effects of personality combinations on communication were analyzed for 44 of the pairs.	productivity, quality, targets [P7]; communication [P148]; partner combinations [P7,P148]
P12	A survey at Microsoft answered by 492 persons. The papers report the past, current or intended use of PP. The responses of 106 developers, who had used PP, were analyzed regarding the benefits and problems of PP, and attributes of good PP partners and teams.	feelings of PP, prevalence
P14	A case study of a 6–11 person team that lasted several one-week iterations. Variations in task ownership, task assignment and pair rotation frequency were studied regarding their effects to productivity ("velocity").	managing PP, pair formation
P20, P21, P24	All papers contain slightly improved analyses of partially same dataset of 24 one-hour transcribed recordings of verbal communication in PP sessions in four companies. They analyzed the effect of partner's role on the amount of communication on different abstraction levels during PP sessions. The viewpoint of the analysis differs in each paper.	partner combinations [P24]; communication, partners' roles [P20,P21,P24]
P28	A case study where 40 hours of PP session communication from two teams was recorded and transcribed. Based on qualitative analysis they discuss the lack of the driver and navigator roles.	communication, partner's roles
P65	A case study of four 8-week projects in close-to-industry setting with students and professionals.	local amount, productivity, quality
P96	Two experiments where 15 and 10 subjects solved deduction problems that simulated programming using PP or SP. Tasks lasted a couple of hours.	productivity, quality
P104	An experiment where 15 subjects performed a task related to their work lasting about half an hour and using either PP or SP.	productivity, quality
P121	An experiment where 16 developers developed a small system during a day using either PP or SP. The effect of partners' experience combination on role-switching frequency was studied as well as the differences between PP and SP regarding productivity and effort spent for various activities.	partners' roles, productivity
P125	A survey of 42 organizations on agile methods. Covers topics such as whether they knew or used PP, benefits of PP, and why PP was not adopted.	adoption, prevalence
P144, P145	A two-year-long case study in an organization. Consecutive surveys for dozens of developers were done. In [P145] trends in the amount of PP and in developers' feelings on PP, its organization and infrastructure were analyzed. Issues and tactics in adoption were discussed. In [P144], developers' perceptions on the effects of PP on numerous topics were surveyed.	feelings of work, knowl. of work, dev. process, targets [P144]; adoption, feelings of PP, infrastructure, local amount, managing PP [P145]

Pair formation ranks 9th with no excellent, one good and 15 moderate instances. The good instance [P14] (see Table 13) analyzes variations in both the partner rotation frequency (1 hour–3 days) and who continues with a task regarding their effects to productivity. The variations in who continues were: 1) a member of the initial pair until the task was ready; 2) the person having worked a shorter time with the task.

The moderate instances, typically from case studies or experience reports, discuss qualitatively, sometimes even quite broadly, how and when pairs were formed and how often pairs were rotated in individual cases.

The main gap in research is that there are no measured instances, and there are only two instances of comparative data [P14,P143] both of which study the partner rotation aspect. The aspects related to initial pair formation such as planned formation organized by managers vs. developers vs. ad-hoc formation, lack totally any good studies.

Targets ranks 7th with one excellent and two good instances. The excellent instance [P7] (Table 13) analyzed the effect of task complexity on the effort and quality differences between PP and SP. The first good instance [P21] (Table 13), compared the amount of communication during activities of different complexity such as writing new code, testing, and debugging. The second [P144] (Table 13) inquired developers' opinions on suitable amount of PP for various activities.

Most of the other instances are short, descriptive, qualitative remarks from individual cases discussing types of activities or tasks for which PP was used.

As listed above, there are already some good studies, and they have shown that this factor is a significant context factor of PP. However, there are still gaps in research, e.g. in studying also other activities than programming, and considering different task characteristics including even the constituent nuances of task complexity in more detail.

4.4.2. Environment

Infrastructure ranks 13th with no excellent, one good and 10 moderate instances. The good instance [P145] (Table 13) discusses infrastructure related challenges, some solutions to them, and changes in developers' feelings of the PP infrastructure before and after certain changes.

The moderate instances mostly describe physical infrastructure such as office layout, desks, and workstations used in an organization. Sometimes a short evaluation of their effect to performing PP is included. Many of the fair instances briefly mention noise from practicing PP in the open office as either disturbing noise or useful information.

The main gap in research is that there are no measured or quantitative instances of any aspects of infrastructure such as the amount of the utilization of two keyboards or two workstations, or amount of noise from PP. Also, there are only two comparative instances [P145,P91] of which [P145] analyzes variations in several aspects of infrastructure and [P91] variations in table types.

Development process ranks 16th with no excellent and one good instance. The good instance [P144] (Table 13) analyzed the effect of PP to the discipline with work practices and the amount of refactoring. Most of the other instances are qualitative and descriptive, and based on informal observations from experience reports or case studies. They are mostly related to PP's effect to increasing concentration to work, to increasing discipline on following other work practices, and to dependencies with PP and some other practice.

The main gap in research is that there is only one instance [P62] having measurement as the data collection method and only three quantitative instances [P144,P38,P62]. In these the studied aspects included discipline with work practices in general [P144], discipline with TDD [P62], and the creation of test cases vs. brainstorming as a preceding activity for PP [P38]. There is no good data on aspects such as the effect of PP on concentration on work, following coding standard or usefulness of separate code reviews.

4.4.3. PP session

Partner combinations ranks 8th with two excellent and one good instance. The first excellent instance [P7] (Table 13) analyzed the effects of different partner combinations on the effort and quality differences between PP and SP. The pairs were formed from two seniors, two intermediates or two juniors. The second [P148] (Table 13) analyzed for 44 pairs of the same experiment the effect of personality combinations on the content and amount of communication. The good instance [P24] (Table 13) analyzed the effect of partners' PP experience combinations to communication.

A few moderate instances contain quantitative data of, e.g., frequencies of using various experience combinations. The remaining instances are mostly short, qualitative and descriptive remarks from individual cases discussing what combinations were used, and positive and negative comments on the various combinations.

There is only one instance [P47] having measurement as the data collection method. It measured the frequency of different skill combinations in a project team. However, the main gap in research is related to getting comparative data of all potentially relevant aspects of developers' characteristics and knowledge combinations and their effect to the outcomes of PP. The available seven comparative instances cover only a small subset of potentially relevant combinations and outcomes.

Partners' roles ranks 10th with two excellent and three good instances. One excellent [P20] and two good [P21,P24] instances from the same study (Table 13) analyzed the effect of partner's role on the amount of communication on different abstraction levels. The second excellent instance [P28] (Table 13) analyzed the existence of roles. The third good instance [P121] (Table 13) analyzed the effect of partners' experience combinations to role switching frequency.

The only moderate instance [P144] (Table 13) reports the proportion of developers switching roles during PP sessions. The fair instances are mostly short, qualitative and descriptive remarks from individual cases discussing whether there were some roles between the partners, and if so, did the partners switch roles.

There is only one instance [P121] having measurement as the data collection method. It measured the effect of work experience on role switching frequency. There is no measured data on, e.g. the ratio of keyboard possession between the partners. The main gap in research is related to getting comparative data on, e.g., role switching frequency or keyboard possession proportions on productivity, quality, and developer's knowledge.

Communication ranks first with four excellent and five good instances. All excellent and good instances discuss the amount and content of communication [P20,P21,P24,P28,P39,P44,P45,P120,P148] including the effects of partner's personality [P148] or role [P20,P21,P24] to communication.

For six of these nine instances, dozens of hours of communication between partners was audio/video recorded and analyzed from several pairs. However, in [P120] and [P44,P45] only a few hours from one pair was recorded or analyzed. In all papers by Bryant et al. [P20,P21,P24] and in [P148] transcribed communication items were classified and analyzed quantitatively; e.g., as numbers of items per abstraction level. In

[P20,P21,P24,P28,P120] developers were observed in their daily work, whereas in [P39,P44,P45,P148] developers performed artificial tasks.

The other instances were mostly of fair relevance, descriptive, qualitative, short remarks related to observations of or guidelines for content of communication or relationship between the partners.

The main gap in research is the lack of comparative data on the effect of the amount and content of communication to productivity, quality and developer's knowledge.

Breaks ranks 17th with one excellent and no good instances. The excellent instance [P29] analyzed 40 hours of observation data from an organization considering three types of work interruptions: intrusions, distractions and breaks. The number and duration of interruptions were compared between two development teams of which only one used PP. All the other instances are descriptive, and nine of them qualitative.

The main gaps in research are the lack of more data similar to [P29], and in studying also other aspects of breaks such as the effect of refreshing breaks for productivity and quality considering the potential exhaustiveness of PP.

4.4.4. Developer

Feelings of PP ranks 15th with no excellent and two good instances. In the first good instance [P12] (Table 13), 106 respondents evaluated on a 5-point agreement scale whether PP "works well" for them, their partner, their team and their larger group. The second [P145] (Table 13) analyzed developers' preconceptions and changing feelings of PP compared to SP, the effect of work experience on feelings of PP, and the feelings of PP when working as the more or less skillful partner.

Two moderate instances [P52,P114] report percentages of developers who liked PP more than SP. All the other instances are short, descriptive and qualitative remarks from individual cases that often mention some developer's resistance to or enjoyment of PP.

The main gap in research is the scarcity of comparative data where developers who have plenty of PP and SP experience would evaluate their feelings of PP.

Feelings of work ranks 14th with no excellent and two good instances. In the first good instance [P145] (Table 13) 22 developers were queried about the effect of using PP vs. SP on team spirit and enjoyment of work. The second [P8] reports based on interviews in an organization many aspects of feelings of work that were affected by PP. The remaining instances are mostly short, qualitative and descriptive remarks from individual cases discussing, e.g., exhaustiveness of PP or effects of PP to team spirit.

Similar to feelings of PP, the main gap in research is the scarcity of comparative data where developers having plenty of experience of both solo and pair programming would evaluate the effect of PP on the numerous aspects of feelings of work.

Knowledge of work ranks 2nd with no excellent, one good, but as many as 64 instances in total. The good instance [P144] (Table 13) reports perceived effects of PP vs. SP regarding the changes in learning about developed software, development tools, work practices, refactoring old code, and new technologies. Most of the other instances discuss improvements in learning some aspects of work, but mention it only briefly and without basing the claims on any collected data.

The main gap in research is the lack of instances having measurement as the data collection method. Measuring changes in various aspects of knowledge of work should not even be too difficult, but plenty of studies are needed to fill all the gaps related to studying the potential effects of the numerous context factors of PP on them.

Developer's characteristics ranks 18th; i.e., last with no excellent, no good, and three moderate instances. A few moderate instances identified attributes of good partners based on surveys. Other instances only briefly mention some relevant attributes of developers.

There are many gaps in research. There are no instances having measurement or rigorous observation as the data collection method, and no comparative instances. Thus there is no good data about the suitability of PP for certain types of persons better than for some other persons. Of some potentially relevant aspects, such as the age or gender of a developer, there are not even instances of lesser quality.

4.4.5. Utilization rate

Local amount ranks 4th with three excellent and three good instances. Two of the excellent instances [P33,P47] analyze data from the same case study, where the amount of PP by 16 developers was measured using a nonintrusive software tool for several months. They also analyze the relationship between the work experience and the amount of PP. In third [P145] (see Table 13), the realized and desired amount of PP as hours per month was asked using four repeated surveys during a two year time period.

Three good instances are case studies that report the amount of PP as proportions of all work per iteration or release. In [P143] and [P65] (see Table 13) the developers reported effort data on task sheets, and in [P151] PP proportion was evaluated both objectively based on code headers and subjectively by developers in a survey.

All eight moderate instances provide quantitative data from individual cases typically regarding proportion of PP of all work. The fair instances are typically short, qualitative and descriptive remarks on using PP; e.g., for all production code or whenever possible.

The main gap in research is the lack of measured, comparative data on the effect of PP proportion of the task or project level effort on the productivity, quality and developer's knowledge.

Prevalence ranks 10th with two excellent and five good instances. The first excellent instance is a survey [P124] where the usage level of various SCRUM and XP practices including PP on a 5-point scale was enquired. The respondents were 35 projects from 13 different European embedded software development organizations. The second [P125] (Table 13) enquired where 42 Austrian organizations whether they knew and used PP.

Two good instances from the survey in [P12] and [P13] (Table 13) report: 1) the current or intended use for PP and other practices on a 5-point scale [P13] and 2) number of respondents having used PP in the past or using it in the current project [P12]. The third is a survey by Hofer [P63], who asked 70 small Austrian enterprises whether they were using PP now or planning to use it. The fourth is a survey [P92] reporting answers from 112 organizations (only 5.7% response rate) about the implementation of XP practices on a 4-point scale. Fifth is a survey [P35] about 104 projects world-wide that provides comparative data about the prevalence among four geographical areas. All the other instances inquired the use of PP on a binary yes/no scale.

There are already several surveys on the prevalence of PP. An improvement over the existing studies would be to think more carefully the scale used when surveying the extent of PP use in the companies. Based on the previous surveys it is very difficult to say how large a proportion of development work in each company is done as pairs.

4.4.6. Main effects

Productivity ranks third with one excellent and four good instances. The excellent instance [P7] (Table 13) compared the effort required to perform tasks correctly between PP and SP considering also the effects of task complexity and pair's experience level. Three good instances [P104,P121,P96], (Table 13) are from experiments with at most a few dozen subjects who performed tasks lasting at most a day using PP or SP. Fourth [P65] (Table 13) compared lines of code per hour between PP and SP.

There are many other instances with quantitative, comparative data from case studies, surveys or experiments that has been collected by measurement or questionnaires. However, their data originates from only one project or one pair, and in some cases data is only based on perceptions of subjects instead of direct measurement of productivity.

Despite of the rather large amount of good research data there are still gaps in research to be filled. There are numerous context variables, e.g., many aspects of targets of PP or developer's characteristics, whose effect on productivity has not been studied adequately or at all. Also, the experiments have involved tasks lasting only a few hours meaning that their results may not apply to long-term project level productivity. Sometimes the subjects have not had experience of PP, which combined with short tasks means comparing solo programmers with programmers who are only learning PP.

Quality ranks fourth with one excellent and three good instances. The excellent instance [P7] (Table 13) compared the proportion of correct solutions between PP and SP. The first good instance [P104] (Table 13) compared the readability and functionality of the resulting code between PP and SP. In [P96] (Table 13) the number of resubmissions until the correct answer was reached was compared between PP and SP. In [P65] (Table 13) coding standard deviations, comment ratio and defect density between PP vs. SP were evaluated.

There are a couple other experiments and surveys with moderate instances, and many with fair instances. The fair instances are typically short, qualitative remarks based on informal observation of improvements in some quality aspects.

The same gaps in research mentioned above under productivity apply also to quality. In addition, as quality is a more multifaceted factor than productivity, there are aspects of quality that have not been studied well such as quality of design.

4.5. Further studies for filling identified gaps in research

Below we propose how the most relevant gaps in research could be filled (RQ 6). Based on the criteria presented in section 2.3.4, we selected four factors for which we give recommendations for further studies. They are *development process*, *targets*, *developer's characteristics*, and *feelings of work*.

4.5.1. Development process

The *development process* factor is a moderately broad topic containing aspects related both to general discipline with following the agreed process and to various dependencies between PP and other work practices. It has practical relevance as PP is not going to be used in industry if it interferes with many of the current practices. Experience reports contain lots of tentative data on process conformance and various dependencies. However, advanced studies are missing, and therefore *development process* ranked very low in the state of research ranking.

There is only one PP study reporting measured data related to *development process*. In that study [P62] the TDD conformance was analyzed using a sophisticated TDD analysis framework. Differences in TDD conformance were compared only between different types of pairs regarding PP experience. The same experimental design could be used to study differences in TDD conformance between pairs and individuals.

The design of the largest and most rigorous PP experiment so far [P7] could also be used to study some aspects of process conformance by requiring the use of TDD or some other practice by the subjects, and then measuring the conformance. Actually, this aspect could have been studied with only minimal additional arrangement costs in the original experiment without affecting its other research questions. Also the measurement and analysis costs do not need to be high at least for practices such as TDD, unit test coverage or coding standard, whose conformance can be automatically measured from the source code. Thus, future studies of PP should consider such arrangement as it would greatly help our knowledge of the interplay of the development practices with minimal cost.

If the required work practices were varied between subjects, e.g. PP + TDD vs. PP + no TDD, the same design could be used to study the dependencies between PP and other practices. The number of possible dependencies is huge, but comments from experience reports could be used as a basis for choosing which dependencies are worth studying.

A case study [P144] reports comparative data on process conformance from a case study based on the perceived effects of the developers collected using surveys. Such data is easy to collect from any case study and would still be among the most reliable data available as long as no new experiments or case studies with measurement on process conformance are conducted.

4.5.2. Targets

The *targets* factor is a rather concise topic referring to the activities and situations for which PP is used, and to the characteristics of these targets. Most of the previous studies have focused on studying the programming activity. However, any other activity such as specification, design or testing could be studied in a similar way, if only activity specific quality metrics were used. One could argue that studying PP in the context of design would make more sense than for programming because the design quality has longer lasting effect to future development efforts than the source code, and because the design needs to be communicated to team members more often than details of the source code.

The effect of programming task complexity was considerable in the experiment [P7], which compared two tasks of different complexity. However, it is still impossible to say whether there is a linear relationship between task complexity and effects of PP or, e.g., some threshold after which PP becomes beneficial. Further studies could use a larger range of tasks, and also consider in more detail possible dimensions of task complexity. For example, in [P7], the difference in task complexity meant changing an application with a centralized control-style (easy) or one with a delegated control-style (difficult), which is only one very specific situation without a theoretical explanation of how and why the complexity of such tasks was different.

Studying the various situations, for which PP has been proposed to be beneficial, such as when a new developer joins a team, or when starting a new project, requires long studies. However, a study could, e.g. measure the changes in productivity, quality and knowledge of new and old developers over a longer period of time, after several new developers have joined an existing development team at the same time. Some of the new developers would pair with each other, some with old developers and some work alone.

4.5.3. Developer's characteristics

The *developer's characteristics* factor covers aspects such as personality, self-esteem, communications skills and nationality of an individual developer. It is an important factor as there may well be certain types of developers for whom PP is an especially good or bad practice. It is the least rigorously studied factor lacking any good data.

There is already a good study [P148] that analyzed the effect of partners' personality combinations on pair collaboration. However, the data from that study and all other PP studies should be analyzed considering the effects of the characteristics of the individual developers for the performance of a pair, in addition to the partner combination point of view. There are also characteristics with no studies at all such as age or gender.

4.5.4. Feelings of work

The *feelings of work* factor covers aspects such as team spirit, enjoyment of work, and exhaustiveness of work. These aspects are important as such but are also likely to affect the productivity of software development in the long run. Measuring such aspects typically needs to be based on the subjective opinions of the developers with a possible exception regarding physiological measurements of the exhaustiveness of work.

The subjective opinions have been rigorously collected only in one study [P144] even though such data could be easily collected in any case study. Acquiring most reliable subjective data would require that the developers have a good possibility to compare the solo and pair programming settings. For example, a setting where developers are inquired before and after there has been change in the use of PP, or one where the developers work both in PP and solo programming teams in the same organization.

5. Threats to Validity

The main threats to the validity of our results are related to: 1) the completeness of including all relevant papers and 2) the robustness of the classification system that forms the basis of the data analysis.

5.1. Completeness of the included papers

We claim that the completeness of including all relevant papers is very high within the defined scope, and thus our results based on the included papers can be generalized to represent the state of all the empirical, industrial PP research published in scientific journals and conferences. The completeness is based on the following aspects of our study. Firstly, we had a very careful search and paper selection process including 1) searching as many as seven databases, 2) manually looking through relevant proceedings missing from the databases, 3) checking the reference lists of the included papers, and 4) applying database searches on the full text of the papers when possible. Our analysis of the papers found from the different databases revealed that the three smallest databases no more provided any additional papers over the four largest databases. Secondly, the reference lists of the included papers revealed only four papers that were not already found by our other searches. Thirdly, we validated that our searches found all the papers in our existing, manually collected, large set of PP papers. Fourthly, the paper selection process was applied to 5% of hits by another reviewer. The inter-rater agreement between the reviewers was high indicating very low incorrect exclusion of relevant papers.

We excluded papers that were not published in scientific journals, conferences or workshops, and papers that were not written in English. It meant excluding some relevant material, but not much because we found only a few references in the included papers to this kind of material, such as theses or other books. We excluded studies conducted in student context or in fields other than software development, and therefore we do not attempt to say anything about them based on our study.

5.2. Robustness of the classification system

The robustness of the classification system includes: 1) the completeness of identifying all factors of PP, and 2) the reliability of the classification system (factors, research property categories and relevance categories). Deciding how to classify and categorize data may be one of the major problems in a mapping study [5]. We chose to tackle this problem by spending lots of effort for piloting various classification schemes for small subsets of papers until we were satisfied.

The completeness of identifying all factors has limitations, because all potentially relevant factors are not necessarily covered in the included papers. For example, studies in student context have mentioned additional, industrially relevant aspects of PP such as developer's gender [21]. Theoretical papers or unscientific material may also propose relevant aspects. However, being familiar also with such materials, we are not aware of any major aspects of PP that would not fit under the existing factors in our framework.

The reliability of the classification system was validated by having two authors process the same 11% sample of the papers. Based on the validation we estimate that 14.8% of the instances escaped our attention and for 7.6% of the instances the authors classified the same data under different factors. However, both types of errors were always related to instances having only fair relevance.

For all the research property categories, the inter-rater agreement was high as expected because their classification is a very objective and mechanical task. However, for relevance, the subjective classification varied between the reviewers for every third instance, but always only by one category level. Having a mathematical formula for calculating the relevance from the research properties would remove this error. However, subjective evaluation based on heuristics allows a sanity check and a possibility to consider viewpoints that are not easy to put into an explicit formula, which after all also has the subjective element in the form of some arbitrary weights given to the various research properties and their categories.

5.3. Other threats to validity

There can be many different opinions on how to characterize the research of a particular SE topic, and how to compare the state of research among factors. We chose to characterize each study using a certain set of commonly used research properties, and in addition gave an overall relevance value based on the research properties in order to facilitate quantitative comparison of the state of research among factors. In the state of research index we chose to give exponentially more value to data that has higher relevance. The index has weaknesses, when used for ranking the factors, e.g., regarding the varying "size" of the factors or regarding weighting the relevance of instances from different papers based on the same dataset.

The first author of this study had vested interests in the results as he has published several papers on PP. His familiarity with the PP research increased the probability of finding all relevant papers and decreased the review effort. However, it may have caused some bias, e.g., the PP framework may reflect the content of his studies on PP.

6. Conclusions and Future Work

6.1. Conclusions

This paper characterized the PP research in the industry based on a systematic mapping study of scientific, empirical papers about PP in the industry. The study was unusually broad considering the completeness of searching relevant, scientific papers; e.g., through applying the searches to full texts of papers instead of only metadata. It is also an unusually deep mapping study considering the thoroughness of analyzing the content of the included papers. The main contributions are: 1) the created PP framework (sec. 4.1), 2) the characterization of the previous industrial PP research in general and per factor, (sec. 0-4.4), 3) the identification of gaps in research (sec. 4.4) and concrete recommendations for filling the most relevant gaps (sec. 4.5).

We identified numerous aspects of PP that we grouped under 18 factors of PP. Compared to the previous PP frameworks, we present two new factors, *prevalence of PP* and *PP session breaks*, and many detailed examples of additional aspects of PP under each of the 18 factors.

Only 27 of the 154 papers (18%) were published in journals, and only 13 of the 154 papers (7%) were experiments. The proportion of the least reliable studies; i.e., experience reports was as high as 44% producing lots of data into the research property categories with lower relevance of research such as informal observation or descriptive data. The amounts of data in the high relevance categories such as quantitative data, comparative discussion, measurement and rigorous observation are from zero to a few instances for the majority of the factors.

Communication, knowledge of work, *productivity* and *quality*, are the best studied factors. *Developer's characteristics*, *breaks* and *development process* are the least studied ones. For half of the 18 factors there were no papers containing data that had excellent relevance. This was mainly because, for many factors, there are very few or no instances in the high relevance research property categories.

We identified many gaps in the PP research in industry. We gave recommendations for further primary studies on four factors for which further studies would be most valuable. These factors include *development process*, *targets*, *developer's characteristics*, and *feelings of work*. In many cases, if the researchers of the previous primary studies would have known the gaps in research, they could have extended their study design and data collection to cover them with only small additional costs. In the future, this systematic mapping study helps to avoid missing such opportunities.

We conclude that there is still plenty of research to be done on the use of PP in the industry. Our PP framework helps the PP researchers and practitioners consider broadly the relevant factors of PP. The identification of the most relevant papers of each factor allows the researchers quickly find the relevant previous research and build their further research upon them. Finally, the identification of the gaps in the PP research in the industry allows the researchers to focus further PP studies to fill these gaps.

6.2. Future work

The same review protocol can be applied to PP papers from the student context. With small modifications to the protocol it could be applied to other than empirical PP papers. These extensions to the scope would mean extracting data from about 250 further papers that were already identified. Including nonscientific material would allow finding more gaps in research by listing all claims made of PP and then comparing it to what has been studied. For the largest part the protocol could be used for making a mapping study on any SE topic as long as a topic specific framework is built.

A further study could compare methods for evaluating the relevance of research of the previous studies. We used heuristic evaluations applied to objective research property data. Another method would be to use some objective formula based on some subjective, fixed weighting of the research property data. Which method would provide the most similar ranking to one resulting from a group of PP researchers thoroughly reading a set of papers about a PP factor and ranking them without any guidelines?

Our data also allows further analysis of the state of the PP research in the industry. For example, the degree of basing further research on top of previous work could be evaluated by counting how large proportion of the most relevant papers discussing the studied factors is referenced in the later papers.

Appendix A. Included Papers

[P1] A. Ahmed, M.M. Fraz, and F.A. Zahid, "Some results of experimentation with extreme programming paradigm," Proc. 7th Int'l Multi Topic Conf. (INMIC), 2003, pp. 387-390.

[P2] J. Aiken, "Technical and human perspectives on pair programming," SIGSOFT Software Eng. Notes, vol. 29, no. 5, 2004, pp. 1-14.

[P3] M. Ally, F. Darroch, and M. Toleman, "A framework for understanding the factors influencing pair programming success," Proc. 6th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2005), 2005, pp. 82-91.

[P4] S.W. Ambler, "Survey says: Agile works in practice," Dr.Dobb's Journal, vol. 31, no. 9, 2006, pp. 62-64.

[P5] W. Ambu and F. Gianneschi, "Extreme programming at work," Proc. 4th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 347-350.

[P6] J. Andrea, G. Meszaros, and S. Smith, "Catalog of XP Project 'Smells'," Proc. 3rd Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2002), 2002.

[P7] E. Arisholm, H. Gallis, T. Dybå and D.I.K. Sjoberg, "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise," IEEE Trans. Software Eng., vol. 33, no. 2, 2007, pp. 65-86.

[P8] J. Auvinen, R. Back, J. Heidenberg, P. Hirkman, and L. Milovanov, "Software process improvement with agile practices in a large telecom company," Proc. 7th Int'l Conf. Product-Focused Software Process Improvement (PROFES), 2006, pp. 79-93.

[P9] R.-. Back, L. Milovanov, and I. Porres, "Software development and experimentation in an academic environment: The Gaudi experience," Proc. 6th Int'l Conf. Product Focused Software Process Improvement (PROFES), 2005, pp. 414-428.

[P10] R.-. Back, P. Hirkman, and L. Milovanov, "Evaluating the XP customer model and design by contract," Proc. 30th Euromicro Conf. 2004, pp. 318-325.

[P11] S. Beecham, H. Sharp, N. Baddoo, T. Hall, and H. Robinson, "Does the XP environment meet the motivational needs of the software developer? An empirical study," Proc. AGILE, 2007, pp. 37-48.

[P12] A. Begel and N. Nagappan, "Pair programming: What's in it for me?" Proc. 2nd Int'l Symposium on Empirical Software Eng. and Measurement (ESEM), 2008, pp. 120-128.

[P13] A. Begel and N. Nagappan, "Usage and perceptions of Agile software development in an industrial context: An exploratory study," Proc. 1st Int'l Symp. on Empirical Software Eng. (ESEM), 2007, pp. 255-264.

[P14] A. Belshee, "Promiscuous pairing and Beginner's mind: Embrace inexperience," Proc. AGILE, 2005, pp. 125-131.

[P15] J.A. Blotner, "Agile techniques to avoid firefighting at a start-up," Proc. OOPSLA 2002 Practitioners Reports, 2002, pp. 1-ff.

[P16] S. Borges, J. Gilmore, and S.E. Oliveira, "Agile: Adopting a new methodology at Harvard Business School," Proc. AGILE, 2007, pp. 249-254.

[P17] J. Bowers, J. May, E. Melander, M. Baarman, and A. Ayoob, "Tailoring XP for Large System Mission Critical Software Development," Proc. 2nd XP Universe and 1st Agile Universe Conf. Extreme Programming and Agile Methods, 2002, pp. 100-111.

[P18] T. Bozheva, "Practical aspects of XP practices," Proc. 4th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 360-362.

[P19] G. Broza, "Adapting Extreme Programming to research, development and production environments," Proc. 4th Conf. Extreme Programming and Agile Methods - XP/Agile Universe, 2004, pp. 139-146.

[P20] S. Bryant, P. Romero and B. du Boulay, "Pair programming and the mysterious role of the navigator," Int. J. Human Computer Studies, vol. 66, no. 7, 2008, pp. 519-529.

[P21] S. Bryant, P. Romero, and B. Du Boulay, "The collaborative nature of pair programming," Proc. 7th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2006), 2006, pp. 53-64.

[P22] S. Bryant, "Rating expertise in collaborative software development," Proc. 17th Ann. Workshop of the Psychology of Programming Interest Group, 2005, pp. 19-29.

[P23] S. Bryant, P. Romero, and B.d. Boulay, "Pair programming and the re-appropriation of individual tools for collaborative programming," Proc. Int'l ACM SIGGROUP Conf. Supporting Group Work (GROUP), 2005, pp. 332-333.

[P24] S. Bryant, "Double Trouble: Mixing Qualitative and Quantitative Methods in the Study of eXtreme Programmers," Proc. IEEE Symposium on Visual Languages - Human Centric Computing (VLHCC), 2004, pp. 55-61.

[P25] G. Canfora, A. Cimitile, F. Garcia, M. Piattini and C.A. Visaggio, "Evaluating performances of pair designing in industry," J. Systems and Software, vol. 80, no. 8, 2007, pp. 1317-1327.

[P26] L. Cao, K. Mohan, P. Xu, and B. Ramesh, "How extreme does extreme Programming have to be? Adapting XP practices to large-scale projects," Proc. Hawaii Int'l Conf. System Sciences, 2004, pp. 1335-1344.

[P27] J. Chao and G. Atli, "Critical Personality Traits in Successful Pair Programming," Proc. AGILE, 2006, pp. 89-93.

[P28] J. Chong and T. Hurlbutt, "The Social Dynamics of Pair Programming," Proc. 29th Int'l Conf. Software Eng. (ICSE), 2007, pp. 354-363.

[P29] J. Chong and R. Siino, "Interruptions on software teams: a comparison of paired and solo programmers," Proc. 20th Anniversary Conf. Computer Supported Cooperative Work (CSCW), 2006, pp. 29-38.

[P30] J. Chong, "Social behaviors on XP and non-XP teams: a comparative study," Proc. AGILE, 2005, pp. 39-48.

[P31] A. Cockburn and L. Williams, "The Costs and Benefits of Pair Programming," Proc. 1st Int'l Conf. Extreme Programming and Flexible Processes in Software Eng. (XP 2000), 2000.

[P32] S. Cohan, "Successful integration of agile development techniques within DISA," Proc. AGILE, 2007, pp. 255-260.

[P33] I.D. Coman, A. Sillitti, and G. Succi, "Investigating the Usefulness of Pair-Programming in a Mature Agile Team," Proc. 9th Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2008), 2008, pp. 127-136.

[P34] G. Concas, M. Di Francesco, M. Marchesi, R. Quaresima, and S. Pinna, "Study of the evolution of an agile project featuring a web application using software metrics," Proc. Product-Focused Software Process Improvement (PROFES), 2008, pp. 386-399.

[P35] M. Cusumano, A. MacCormack, C.F. Kemerer and B. Crandall, "Software Development Worldwide: The State of the Practice," IEEE Software, vol. 20, no. 6, 2003, pp. 28-34.

[P36] A.F. Da Silva, F. Kon, and C. Torteli, "XP south of the equator: An eXPerience implementing XP in Brazil," Proc. 6th Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2005), 2005, pp. 10-18.

[P37] A.J. Dick and B. Zarnett, "Paired Programming & Personality Traits," Proc. 3rd Int'l Conf. XP and Agile Processes in Software Eng. (XP 2002), 2002, pp. 82-85.

[P38] M.A. Domino, R.W. Collins and A.R. Hevner, "Controlled experimentation on adaptations of pair programming," Information Technology and Management, vol. 8, no. 4, 2007, pp. 297-312.

[P39] M.A. Domino, R.W. Collins, A.R. Hevner, and C.F. Cohen, "Conflict in collaborative software development," Proc. SIGMIS Conf. Computer personnel research (SIGMIS CPR), 2003, pp. 44-51.

[P40] J. Drobka, D. Noftz and R. Raghu, "Piloting XP on four mission-critical projects," IEEE Software, vol. 21, no. 6, 2004, pp. 70-75.

[P41] Y. Dubinsky, O. Hazzan, and A. Keren, "Introducing extreme programming into a software project at the Israeli Air Force," Proc. 6th Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2005), 2005, pp. 19-27.

[P42] A. Elssamadisy, "XP On A Large Project – A Developer's View," Proc. 1st XP Universe Conf. 2001.

[P43] B. Fitzgerald, G. Hartnett and K. Conboy, "Customising agile methods to software practices at Intel Shannon," European J. Information Systems, vol. 15, no. 2, 2006, pp. 200-213.

[P44] N.V. Flor, "Side-by-side collaboration: A case study," Int'l J. Human Computer Studies, vol. 49, no. 3, 1998, pp. 201-222.

[P45] N.V. Flor and E.L. Hutchins, "Analyzing distributed cognition in software teams: a case study of team programming during perfective software maintenance," Proc. 4th Ann. Workshop on Empirical Studies of Programmers, 1991, pp. 36-64.

[P46] T. Frever and P. Ingalls, "The pairing session as the atomic unit of work," Proc. AGILE, 2006, pp. 165-169.

[P47] I. Fronza, A. Sillitti, and G. Succi, "An interpretation of the results of the analysis of pair programming during novices integration in a team," Proc. 3rd Int'l Symposium on Empirical Software Eng. and Measurement (ESEM), 2009, pp. 225-235.

[P48] A. Fruhling, P. McDonald, and C. Dunbar, "A case study: Introducing eXtreme programming in a US government system development project," Proc. 41st Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2008, pp.464.

[P49] A. Fruhling and G.D. Vreede, "Field Experiences with eXtreme Programming: Developing an Emergency Response System," Journal of Management Information Systems, vol. 22, no. 4, 2006, pp. 39-68.

[P50] A.M. Fuqua and J.M. Hammer, "Embracing change: an XP experience report," Proc. 4th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 298-306.

[P51] R. Gittins, J. Bass, and S. Hope, "A Comparison of Software Development Process Experiences," Proc. 5th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2004), 2004, pp. 231-236.

[P52] R. Gittins, S. Hope, and I. Williams, "Qualitative Studies of XP in a Medium Sized Business," Proc. 2nd Int'l Conf. Extreme Programming and Flexible Processes in Software Eng. (XP 2001), 2001, pp. 122-126.

[P53] C.J. Goebel, "How being agile changed our human resources policies," Proc. AGILE, 2009, pp. 101-106.

[P54] B. Greene, "Agile methods applied to embedded firmware development," Proc. Agile Development Conf. 2004, pp. 71-77.

[P55] J. Grenning, "Launching extreme programming at a process-intensive company," IEEE Software, vol. 18, no. 6, 2001, pp. 27-33.

[P56] H. Grewal and F. Maurer, "Scaling agile methodologies for developing a production accounting system for the oil & gas industry," Proc. AGILE, 2007, pp. 309-314.

[P57] F. Grossman, J. Bergin, D. Leip, S. Merritt, and O. Gotel, "One XP experience: introducing agile (XP) software development into a culture that is willing but not ready," Proc. Conf. of the Centre for Advanced Studies on Collaborative Research (CASCON '04), 2004, pp. 242-254.

[P58] E. Gul, T. Sekerci, A.C. Yücetürk, and Ü. Yildirim, "Using XP in telecommunication software development," Proc. 3rd Int'l Conf. Software Eng. Advances (ICSEA), 2008, pp. 258-263.

[P59] J. Haungs, "Pair Programming on the C3 Project," Computer, vol. 34, no. 2, 2001, pp. 118-119.

[P60] J. Higman, T. Mackinnon, I. Moore, and D. Pierce, "Innovation and Sustainability with Gold Cards," Proc. 1st XP Universe Conf. 2001.

[P61] P. Hodgetts, "Refactoring the development process: Experiences with the incremental adoption of agile practices," Proc. Agile Development Conf. 2004, pp. 106-113.

[P62] A. Höfer and M. Philipp, "An empirical study on the TDD conformance of novice and expert pair programmers," Proc. 10th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2010), 2009, pp. 33-42.

[P63] C. Hofer, "Software development in Austria: results of an empirical study among small and very small enterprises," Proc. 28th Euromicro Conf. 2002, pp. 361-366.

[P64] M. Holcombe and C. Thomson, "Seven Years of XP - 50 Customers, 100 Projects and 500 Programmers – Lessons Learnt and Ideas for Improvement," Proc. 9th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2008), 2008, pp. 104-113.

[P65] H. Hulkko and P. Abrahamsson, "A multiple case study on the impact of pair programming on product quality," Proc. 27th Int'l Conf. Software Eng. (ICSE '05), 2005, pp. 495-504.

[P66] P. Ingalls and T. Frever, "Growing an agile culture from value seeds," Proc. AGILE, 2009, pp. 119-124.

[P67] B. Jensen and A. Zilmer, "Cross-continent development using scrum and XP," Proc. 4th Int'l Conf. Extreme Programming and Agile processes in Software Eng. (XP 2003), 2003, pp. 146-153.

[P68] R. Jochems and S. Rodgers, "The rollercoaster of required agile transition," Proc. AGILE, 2007, pp. 229-233.

[P69] K. Johansen, R. Stauffer, and D. Turner, "Learning by Doing: Why XP Doesn't Sell," Proc. 1st XP Universe Conf. 2001,

[P70] S. Johnson, J. Mao, E. Nickell, and I. Smith, "Extreme makeover: bending the rules to reduce risk rewriting complex systems," Proc. 4th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 307-314.

[P71] D. Karlström, "Introducing Extreme Programming – An Experience Report," Proc. 3rd Int'l Conf. XP and Agile Processes in Software Eng. (XP 2002), 2002,

[P72] D. Karlström and P. Runeson, "Decision support for extreme programming introduction and practice selection," Proc. Int'l Conf. Software Eng. and Knowledge Eng. (SEKE '02), 2002, pp. 835-841.

[P73] S.-. Katriou and E. Tolias, "From twin training to pair programming," Proc. 2nd India Software Eng. Conf. (ISEC), 2009, pp. 101-104.

[P74] K. Kautz and S. Zumpe, "Just Enough Structure at the Edge of Chaos: Agile Information System Development in Practice," in Agile Processes in Software Engineering and Extreme Programming, vol. 9, W. Aalst, J. Mylopoulos, N.M. Sadeh, M.J. Shaw, C. Szyperski, P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan and X. Wang Eds. Springer Berlin Heidelberg, 2008, pp. 137-146.

[P75] W.H. Kee, "Future implementation and integration of Agile methods in software development and testing," Proc. Innovations in Information Technology, 2006.

[P76] N. Kini and S. Collins, "Steering the Car: Lessons Learned from an Outsourced XP Project," Proc. 1st XP Universe Conf. 2001.

[P77] O. Kobayashi, M. Kawabata, M. Sakai, and E. Parkinson, "Analysis of the interaction between practices for introducing XP effectively," Proc. 28th Int'l Conf. Software Eng. 2006, pp. 544-550.

[P78] W. Krebs, "Turning the Knobs: A Coaching Pattern for XP through Agile Metrics," Proc. 2nd XP Universe and 1st Agile Universe Conf. Extreme Programming and Agile Methods, 2002, pp. 60-69.

[P79] Y. Kuranuki and K. Hiranabe, "AntiPractices: AntiPatterns for XP practices," Proc. Agile Development Conf. 2004, pp. 83-86.

[P80] M. Lacey, "Adventures in promiscuous pairing: Seeking Beginner's Mind," Proc. AGILE, 2006, pp. 263-269.

[P81] M. Lange, "After the Fact: Introducing XP into an Existing C++ Project," Proc. 1st Int'l Conf. Extreme Programming and Flexible Processes in Software Eng. (XP 2000), 2000,

[P82] T.D. LaToza, G. Venolia, and R. DeLine, "Maintaining mental models: A study of developer work habits," Proc. 28th Int'l Conf. Software Eng. 2006, pp. 492-501.

[P83] A. Law and R. Charron, "Effects of agile practices on social factors," Proc. Workshop on Human and Social Factors of Software Eng. 2005, pp. 1-5.

[P84] A. Law and A. Ho, "A study case: Evolution of co-location and planning strategy," Proc. AGILE, 2004, pp. 56-62.

[P85] R. Lawrence, "XP and junior developers: 7 mistakes (and how to avoid them)," Proc. AGILE, 2007, pp. 234-238.

[P86] R. Lawrence and B. Yslas, "Three-way cultural change: Introducing agile within two non-agile companies and a non-agile methodology," Proc. AGILE, 2006, pp. 255-259.

[P87] L. Layman, L. Williams, and L. Cunningham, "Exploring extreme programming in context: An industrial case study," Proc. Agile Development Conf. 2004, pp. 32-41.

[P88] L. Layman, L. Williams, and L. Cunningham, "Motivations and measurements in an agile case study," Proc. Workshop on Quantitative Techniques for Software Agile Process (QUTE-SWAP), 2004, pp. 14-24.

[P89] M. Lechner, "XP Team Psychology - An Inside View," Proc. 20th Ann. Workshop of the Psychology of Programming Interest Group, 2008, pp. 114-127.

[P90] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May and T. Kähkönen, "Agile software development in large organizations," Computer, vol. 37, no. 12, 2004, pp. 26-34.

[P91] M. Lippert, S. Roock, H. Wolf, and S. Zumpe, "JWAM and XP Using XP for framework development," Proc. 1st Int'l Conf. Extreme Programming and Flexible Processes in Software Eng. (XP 2000), 2000.

[P92] J.A. Livermore, "What elements of XP are being adopted by industry practitioners?" Proc. SoutheastCon, 2006, pp. 149-152.

[P93] G. Lovaasen, "Brokering with eXtreme Programming," Proc. 1st XP Universe Conf. 2001.

[P94] G. Luck, "Subclassing XP: Breaking its rules the right way," Proc. Agile Development Conf. 2004, pp. 114-119.

[P95] K.M. Lui and K.C.C. Chan, "Software process fusion by combining pair and solo programming," IET Software, vol. 2, no. 4, 2008, pp. 379-390.

[P96] K.M. Lui, K.C.C. Chan and J. Nosek, "The effect of pairs in program design tasks," IEEE Trans. Software Eng., vol. 34, no. 2, 2008, pp. 197-211.

[P97] A. Mackenzie and S. Monk, "From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice," Computer Supported Cooperative Work: CSCW: An International Journal, vol. 13, no. 1, 2004, pp. 91-117.

[P98] L. Madeyski and W. Biela, "Capable leader and skilled and motivated team practices to introduce eXtreme programming," Proc. 2nd IFIP TC 2 Central and East European Conf. Software Eng. Techniques (CEE-SET), 2008, pp. 96-102.

[P99] J. Magalhães, A. Von Staa and C.J.P. De Lucena, "Evaluating the recoveryoriented approach through the systematic development of real complex applications," Software - Practice and Experience, vol. 39, no. 3, 2009, pp. 315-330.

[P100] A. Marchenko, P. Abrahamsson, and T. Ihme, "Long-term effects of test-driven development A case study," Proc. 10th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2009), 2009, pp. 13-22.

[P101] V.B. Mišic, "Perceptions of extreme programming: an exploratory study," SIGSOFT Software Eng. Notes, vol. 31, no. 2, 2006, pp. 1-8.

[P102] G. Mueller and J. Borzuchowski, "Extreme embedded a report from the front line," Proc. OOPSLA 2002 Practitioners Reports, 2002, pp. 1-ff.

[P103] O. Murru, R. Deias and G. Mugheddu, "Assessing XP at a European Internet company," IEEE Software, vol. 20, no. 3, 2003, pp. 37-43.

[P104] J.T. Nosek, "The Case for Collaborative Programming," Comm. ACM, vol. 41, no. 3, 1998, pp. 105-108.

[P105] M.J. O'Donnell and I. Richardson, "Problems Encountered When Implementing Agile Methods in a Very Small Company," Proc. 15th European Conf. Software Process Improvement (EuroSPI), in Communications in Computer and Information Science, 2008, pp. 13-24.

[P106] G. Oliphant, "Convincing the inconvincable," Proc. 4th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 369-371.

[P107] A. Pandey, C. Miklos, M. Paul, N. Kameli, F. Boudigou, V. Vijay, A. Eapen, I. Sutedjo, and W. Mcdermott, "Application of Tightly Coupled Engineering Team for Development of Test Automation Software - A Real World Experience," Proc. 27th Ann. Int'l Computer Software and Applications Conf. (COMPSAC), 2003, pp. 56-63.

[P108] D. Parsons, H. Ryu, and R. Lal, "The impact of methods and techniques on outcomes from agile software development projects," Proc. 10th Working Conf. IFIP Working Group 8.6, 2007, pp. 235-249.

[P109] M. Phongpaibul and B. Boehm, "An empirical comparison between pair development and software inspection in Thailand," Proc. Int'l Symp. Empirical Software Eng. (ISESE), 2006, pp. 85-94.

[P110] M. Pikkarainen, O. Salo, and J. Still, "Deploying agile practices in organizations: A case study," Proc. 12th European Conf. Software Process Improvement (EuroSPI), 2005, pp. 16-27.

[P111] C. Poole and J.W. Huisman, "Using extreme programming in a maintenance environment," Software, IEEE, vol. 18; 18, no. 6, 2001, pp. 42-50.

[P112] D. Poon, "A self funding agile transformation," Proc. AGILE, 2006, pp. 342-350.

[P113] A. Puschnig and R.T. Kolagari, "Requirements engineering in the development of innovative automotive embedded software systems," Proc. Int'l Conf. Requirements Eng. 2004, pp. 328-333.

[P114] V. Ramachandran and A. Shukla, "Circle of Life, Spiral of Death: Are XP Teams Following the Essential Practices?" Proc. 2nd XP Universe and 1st Agile Universe Conf. Extreme Programming and Agile Methods, 2002, pp. 166-173.

[P115] J. Rasmusson, "Introducing XP into greenfield projects: Lessons learned," IEEE Software, vol. 20, no. 3, 2003, pp. 21-28.

[P116] H. Robinson and H. Sharp, "Organisational culture and XP: Three case studies," Proc. AGILE, 2005, pp. 49-58.

[P117] H. Robinson and H. Sharp, "The social side of technical practices," Proc. 6th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2005), 2005, pp. 100-108.

[P118] H. Robinson and H. Sharp, "The Characteristics of XP Teams," Proc. 5th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2004), 2004, pp. 139-147.

[P119] H. Robinson and H. Sharp, "XP culture: why the twelve practices both are and are not the most significant thing," Proc. Agile Development Conf. 2003, pp. 12-21.

[P120] J. Rooksby, D. Martin, and M. Rouncefield, "Reading as Part of Computer Programming. An Ethnomethodological Enquiry," Proc. 18th Ann. Workshop of the Psychology of Programming Interest Group, 2006, pp. 198-212.

[P121] M. Rostaher and M. Hericko, "Tracking Test First Pair Programming - An Experiment," Proc. 2nd XP Universe and 1st Agile Universe Conf. Extreme Programming and Agile Methods, 2002, pp. 174-184.

[P122] D. Rowley and M. Lange, "Forming to performing: The evolution of an agile team," Proc. AGILE, 2007, pp. 408-413.

[P123] B. Rumpe and A. Schröder, "Quantitative Survey on Extreme Programming Projects," Proc. 3th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2002), 2002.

[P124] O. Salo and P. Abrahamsson, "Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of Extreme Programming and Scrum," IET Software, vol. 2, no. 1, 2008, pp. 58-64.

[P125] C. Schindler, "Agile software development methods and practices in austrian IT-industry: results of an empirical study," 2008, pp. 321-326.

[P126] P. Sfetsos, L. Angelis and I. Stamelos, "Investigating the extreme programming system---An empirical study," Empirical Software Eng., vol. 11, no. 2, 2006, pp. 269-301.

[P127] S. Shahzad, "Learning from experience: The analysis of an extreme programming process," Proc. 6th Int'l Conf. Information Technology: New Generations (ITNG), 2009, pp. 1405-1410.

[P128] K. Sharifabdi and C. Grot, "Team Development and Pair Programming – tasks and challenges of the XP coach," Proc. 3rd Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2002), 2002,

[P129] H. Sharp and H. Robinson, "Collaboration and co-ordination in mature eXtreme programming teams," Int'l J. Human Computer Studies, vol. 66, no. 7, 2008, pp. 506-518.

[P130] H. Sharp and H. Robinson, "A distributed cognition account of mature XP teams," Proc. 7th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2006), 2006, pp. 1-10.

[P131] H. Sharp and H. Robinson, "Some social factors of software engineering: the maverick, community and technical practices," Proc. Workshop on Human and Social Factors of Software Eng. (HSSE), 2005, pp. 1-6.

[P132] H. Sharp and H. Robinson, "An Ethnographic Study of XP Practice," Empirical Software Eng., vol. 9, no. 4, 2004, pp. 353-375.

[P133] A. Sigfridsson, G. Avram, A. Sheehan, and D.K. Sullivan, "Sprint-driven development: Working, learning and the process of enculturation in the PyPy community," Proc. 3rd Int'l Conf. Open Source Systems (OSS), 2007, pp. 133-146.

[P134] R. Sison and T. Yang, "Use of agile methods and practices in the Philippines," Proc. 14th Asia Pacific Software Eng. Conf. (APSEC), 2007, pp. 462-469.

[P135] J.W. Spence, "There has to be a better way!" Proc. AGILE, 2005, pp. 272-278.

[P136] M. Strömman, K. Thramboulidis, S. Sierla, N. Papakonstantinou, and K. Koskinen, "Incorporating industrial experience to IEC 61499 based development methodologies and toolsets," Proc. 12th IEEE Int'l Conf. Emerging Technologies and Factory Automation (ETFA), 2007, pp. 490-497.

[P137] H. Svensson and M. Host, "Introducing an agile process in a software maintenance and evolution organization," 2005, pp. 256-264.

[P138] A.B. Tedjasaputra, E.R. Sari, and G. Strom, "Sharing and learning through pair writing of scenarios," Proc. 3rd Nordic Conf. Human-Computer Interaction, 2004, pp. 229-232.

[P139] B. Tessem, "Experiences in learning XP practices: A qualitative study," Proc. 4th Int’l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2003), 2003, pp. 131-137.

[P140] P. Tingling and A. Saeed, "Extreme programming in action: A longitudinal case study," Proc. 12th Int'l Conf. Human-Computer Interaction, 2007, pp. 242-251.

[P141] M. Toleman, F. Darroch, and M. Ally, "Web Publishing: An Extreme, Agile Experience," Proc. IFIP Working Conf. Business Agility and IT Diffusion, in IFIP International Federation for Information Processing, 2005, pp. 245-256.

[P142] N. Van Schooenderwoert, "Embedded agile project by the numbers with newbies," Proc. AGILE, 2006, pp. 351-363.

[P143] J. Vanhanen and H. Korpi, "Experiences of Using Pair Programming in an Agile Project," Proc. 40th Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2007, pp. 274b.

[P144] J. Vanhanen and C. Lassenius, "Perceived Effects of Pair Programming in an Industrial Context," Proc. 33rd EUROMICRO Conf. Software Eng. and Advanced Applications, 2007, pp. 211-218.

[P145] J. Vanhanen, C. Lassenius, and M.V. Mäntylä, "Issues and Tactics when Adopting Pair Programming: A Longitudinal Case Study," Proc. Int'l Conf. Software Eng. Advances (ICSEA), 2007, pp. 70.

[P146] J. Vanhanen, J. Itkonen, and P. Sulonen, "Improving the interface between business and product development using agile practices and the cycles of control framework," Proc. Agile Development Conference, 2003. ADC 2003. Proceedings of the, 2003, pp. 71-80.

[P147] C. Vriens, "Certifying for CMM Level 2 and IS09001 with XP@Scrum," Proc. Agile Development Conf. 2003, pp. 120-124.

[P148] T. Walle and J.E. Hannay, "Personality and the nature of collaboration in pair programming," Proc. 3rd Int'l Symp. Empirical Software Eng. and Measurement (ESEM), 2009, pp. 203-213.

[P149] D. Wells and T. Buckley, "The VCAPS Project: An Example of Transitioning to XP," Proc. 1st Int'l Conf. Extreme Programming and Flexible Processes in Software Eng. (XP 2000), 2000,

[P150] R. Wijnands and I. Van Dijk, "Multi-tasking agile projects: The pressure tank," Proc. 8th Int'l Conf. Agile Processes in Software Eng. and Extreme Programming (XP 2007), 2007, pp. 231-234.

[P151] L. Williams, W. Krebs, L. Layman, A.I. Anton, and P. Abrahamsson, "Toward a framework for evaluating extreme programming," Proc. 8th Int'l Conf. Empirical Assessment in Software Eng. (EASE), 2004, pp. 11-20.

[P152] L. Williams, A. Shukla, and A.I. Anton, "An Initial Exploration of the Relationship Between Pair Programming and Brooks' Law," Proc. Agile Development Conf. 2004, pp. 11-20.

[P153] M.A. Wojcicki and P. Strooper, "A state-of-practice questionnaire on verification and validation for concurrent programs," 2006, pp. 1-10.

[P154] W.A. Wood and W.L. Kleb, "Exploring XP for scientific research," IEEE Software, vol. 20, no. 3, 2003, pp. 30-36.

References

[1] M. Ally, F. Darroch, and M. Toleman, A framework for understanding the factors influencing pair programming success, *in Proc. 6th Int'l Conf. Extreme Programming and Agile Processes in Software Eng. (XP 2005),* 2005, pp. 82-91.

[2] J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton, and S. Linkman, Evidence relating to object-oriented software design: A survey, *in Proc. 1st Int'l Symp. on Empirical Software Eng. and Measurement (ESEM),* 2007, pp. 482-484.

[3] K. Beck, *Extreme Programming Explained: Embrace Change,* (Addison-Wesley, 1999).

[4] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, Motivation in Software Engineering: A systematic literature review, *Information and Software Technology,* 50(9-10) (2008) pp. 860-878.

[5] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, Using Mapping Studies in Software Engineering, *in Proc. 22nd Ann. Workshop on Psychology of Programming Interest Group,* 2008, pp. 195-204.

[6] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor, A systematic mapping study on empirical evaluation of software requirements specifications techniques, *in Proc. 3rd Int'l Symp. Empirical Software Eng. and Measurement (ESEM),* 2009, pp. 502-505.

[7] D.S. Cruzes and T. Dybå, Research synthesis in software engineering: A tertiary study, *Information and Software Technology,* 53(5) (2011) pp. 440-455.

[8] T. Dybå and T. Dingsøyr, Empirical studies of agile software development: A systematic review, *Information and Software Technology,* 50(9-10) (2008) pp. 833-859.

[9] H. Gallis, E. Arisholm, and T. Dybå, An Initial Framework for Research on Pair Programming, *in Proc. Int'l Symp. Empirical Software Eng.* 2003, pp. 132-142.

[10] J.E. Hannay, T. Dybå, E. Arisholm, and D.I.K. Sjøberg, The effectiveness of pair programming: A meta-analysis, *Information and Software Technology,* 51(7) (2009) pp. 1110-1122.

[11] E. Hossain, M. Ali Babar, and H.-. Paik, Using scrum in global software development: A systematic literature review, *in Proc. 4th IEEE Int'l Conf. Global Software Eng.* 2009, pp. 175-184.

[12] M. Jørgensen and M. Shepperd, A Systematic Review of Software Development Cost Estimation Studies, *IEEE Trans. Software Eng.,* 33(1) (2007) pp. 33-53.

[13] B. Kitchenham, D. Budgen, and O.P. Brereton, Using mapping studies as the basis for further research – A participant-observer case study, *Information and Software Technology,* 53(6) (2011) pp. 638-651.

[14] B. Kitchenham, What's up with software metrics? - A preliminary mapping study, *J. Systems and Software,* 83(1) (2010) pp. 37-51.

[15] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, Systematic literature reviews in software engineering - A systematic literature review, *Information and Software Technology,* 51(1) (2009) pp. 7-15.

[16] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering (version 2.3)," Keele University and University of Durham., Tech. Rep. Technical Report EBSE-2007-01, 2007.

[17] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, Systematic Mapping Studies in Software Engineering, *in Proc. 12th Int'l Conf. Evaluation and Assessment in Software Eng.* 2008, pp. 71-80.

[18] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide,* (Blackwell Publishing, 2006).

[19] R. Pretorius and D. Budgen, A mapping study on empirical evidence related to the models and forms used in the UML, *in Proc. 2nd Int'l Symp. on Empirical Software Eng. and Measurement (ESEM),* 2008, pp. 342-344.

[20] N. Salleh, E. Mendes, and J. Grundy, Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review, *IEEE Trans. Software Eng.,* 37(4) (2010) pp. 509-525.

[21] L.L. Werner, B. Hanks, and C. McDowell, Pair-programming helps female computer science students, *J. Educational Resources in Computing,* 4(1) (2004) pp. 4.

[22] L. Williams and R. Kessler, *Pair Programming Illuminated,* (Addison-Wesley, 2002).