International Journal of
Information Systems and Project Management
ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X
Available online at ijispm.sciencesphere.org

Software developers reasoning behind adoption and use of software development methods – a systematic literature review

Tanja Elina Havstorm
Centre for Empricial Research on Information Systems (CERIS),
Örebro University School of Business
Fakultetsgatan 1, SE-702 81, Örebro
Sweden
tanja.havstorm@oru.se

Fredrik Karlsson
Centre for Empricial Research on Information Systems (CERIS),
Örebro University School of Business
Fakultetsgatan 1, SE-702 81, Örebro
Sweden
fredrik.karlsson@oru.se

Abstract:
When adopting and using a Software Development Method (SDM), it is important to stay true to the philosophy of the method; otherwise, software developers might execute activities that do not lead to the intended outcomes. Currently, no overview of SDM research addresses software developers' reasoning behind adopting and using SDMs. Accordingly, this paper aims to survey existing SDM research to scrutinize the current knowledge base on software developers' type of reasoning behind SDM adoption and use. We executed a systematic literature review and analyzed existing research using two steps. First, we classified papers based on what type of reasoning was addressed regarding SDM adoption and use: rational, irrational, and non-rational. Second, we made a thematic synthesis across these three types of reasoning to provide a more detailed characterization of the existing research. We elicited 28 studies addressing software developers' reasoning and identified five research themes. Building on these themes, we framed four future research directions with four broad research questions, which can be used as a basis for future research.

Keywords:
systems development method; software development method; systematic literature review; use; adoption.

DOI: 10.12821/ijispm110203

Manuscript received: 17 January 2023
Manuscript accepted: 1 April 2023

1. Introduction

Inefficient Software Development (SD) is a perennial problem in many organizations where SD projects have had difficulties in meeting their targets [1-3]. As a result, organizations have placed reliance on adopting and using Software Development Methods (SDM). An SDM is a set of standardized activities to design, build, implement, and maintain software, including how these activities are accomplished and structured, the frequency of these activities, and the goals and value behind them [4]. Decisions to adopt and use SDMs can occur "at three different levels" [5]. At the individual level, a software developer chooses an SDM, or part thereof, to support specific SD tasks. A wider agreement can be reached at the project level to adopt and use a specific SDM. Finally, at the organizational level, an SDM "can be chosen as the organization-wide "standard" method to be used in all projects" [5]. Managers are involved in coordinating the decisions to achieve the two latter levels. Thus, adopting and using SDMs are non-trivial tasks. Furthermore, it is well-recognized that SDMs are not used literally [6-8], i.e., exactly as described in textbooks. Instead, software developers modify SDMs. Researchers [9, 10] have found that some of these modifications are beneficial, creating a better fit between the SDMs and the tasks at hand. In other situations, misconceptions or malpractices have been reported [11, 12].

Although SDMs are not followed literally, it is still important that the executed activities align with the method's rationale, i.e., goals and values. SDMs are the result of design efforts, where specific design goals are set out to improve effectiveness and efficiency in SD. The Agile Manifesto [13] is one prominent example. It includes "a set of values and associated goals (referred to as principles)" [9] that guide activities found in agile SDMs. Jayaratna et al. [14] stressed the importance of SDMs' rationale, arguing that "if […] the rationale for the action is implicit then by definition the activity set cannot be considered a methodology." Similar thoughts are found in Brinkkemper [15] and Russo and Stolterman [16]. Consequently, a method's rationale is an important part of an SDM, and goals have been suggested as a vehicle to tailor SDMs to actual SD projects [e.g., 5, 17, 18]. That said, Karlsson [7] concluded that "it [the tailored method] should, at the same time, align with the basic philosophy of the original method […]. Otherwise, there is a risk of losing the method's core idea." For example, if an organization's use of an agile SDM, such as Scrum, deviates from the Agile Manifesto, there is a risk of the organization not being agile. Consequently, to understand SDM adoption and use, it is important to trace whether, and how well, executed activities align with the rationale of the SDM. According to Havstorm et al. [19] software developers' reasoning behind SDM adoption and use can be rational, irrational, or non-rational. Rational reasoning means that the intentionally chosen and executed activities align with the method's rationale, while irrational reasoning means that the intentionally chosen and executed activities misalign with the method's rationale. However, activities can also be non-rational, which means software developers have no awareness of why the activities are chosen and executed and how they align with the method's rationale.

As shown from the number of scholarly review articles in Section 2, SDM adoption and use have attracted significant research interest over the years. These reviews include the classification of SDM literature [20] and granular reviews that focus on subsets of the SDM literature, such as the benefits and limitations of agile SDMs [21] and agile smells [22]. Taken as a collective, these reviews do not provide any overview of SDM research that addresses software developers' reasoning about method rationale when adopting and using SDMs. Thus, it is difficult for researchers and practitioners to comprehend the current state of knowledge regarding this part of the SDM domain. Therefore, we submit that it is time to take stock of the SDM literature and provide an overarching representation of the this literature and software developers' reasoning about method rationale. Accordingly, the aim of this paper is to survey existing SDM research to scrutinize the current knowledge base on software developers' type of reasoning behind SDM adoption and use. We are interested in how large the existing knowledge base is in absolute terms and to understand in relative terms the extent to which the reasoning aspect behind SDM adoption and use has received any focus in SDM research. Moving beyond the quantitative measure, we are also interested in the patterns, if any, found in existing research, i.e., research themes, to characterize the knowledge base.

We pose, therefore, the following research questions (RQ):

RQ 1: To what extent has previous research investigated software developers' reasoning behind SDM adoption and use?

RQ 2: What research themes have previous research investigated concerning software developers' different types of reasoning behind SDM adoption and use?

Our results are based on a structured literature review [23] of SDM research published until 2022. The study is based on an initial search consisting of 1,619 unique research papers. Of these, 111 papers were singled out for further analysis, where we finally identified 28 relevant studies. We have developed and used an analytical framework anchored in Weber's [24] work on social actions to analyze these papers, classifying them as analyzing rational, irrational, and/or non-rational reasoning (see Section 3.3.1). Furthermore, we used thematic synthesis [25] to identify research themes across these three types of reasoning. Thus, our review contributes to a systematic understanding of what is known about software developers' type of reasoning behind SDM adoption and use and pinpoints four areas for future research.

This paper is structured as follows. Following the introduction, Section 2 describes existing SD and SDM literature reviews addressing SDM adoption and use. Section 3 presents the research method adopted for our systematic literature review. In Section 4, we present the results of our review. In Section 5, we discuss our findings and their implications for SDM research. We end the paper with a short conclusion in Section 6.

2. Related research

Given the long history of SDM research, it is not surprising that several SD and SDM literature reviews have been carried out. Table 1 summarizes existing reviews, showing the aims and how they have been categorized. For example, already in 1994, Walters et al. [26] examined "the historical development of ISDMs [Information Systems Development Method]." They also analyzed different themes of SDMs and problems with SDMs. One identified type of problem was the use of SDMs. However, as the rightmost column shows, very few reviews provide results that can be related to software developers' reasoning behind the adoption and use of SDMs.

We found five studies whose results border on our study [20-22, 27, 28]. Abrar et al. [27] reviewed research on agile SDM and large-scale development teams to identify the de-motivator factors while scaling agile. A few identified factors, such as "traditional organizational culture" and "lack of agile experts," can lead to reasoning that deviates from the chosen SDM's rationale. However, the authors did not study software developers' reasoning as such. Dybå and Dingsøyr [21] reviewed empirical papers on agile SDMs, reviewing what was currently known regarding the benefits and limitations of agile SDMs in the industry. They categorized reviewed papers into four themes, and one of them included adoption. Although Dybå and Dingsøyr [21] did not give explicit attention to software developers' reasoning behind adopting agile SDMs, their detailed description reveals that some papers have addressed why the methods have been adopted.

Iivari et al. [20] developed a framework for classifying the understanding of SDMs in previous literature and provided a structure of the intellectual core of SDMs. This framework could help practitioners understand the rationale behind their SDM practices. However, they did not compile studies on practitioners' actual understanding of the rationale. Onwujekwe and Weistroffer [28] reviewed SD literature on implementing agile SD projects in the public sector to understand how bureaucratic organizational cultures impact the use of agile SDMs. Although they address the fit between specific cultures and agile SDMs, they do not address software developers' reasoning behind adopting and using agile SDMs. Finally, Telemaco et al. [22] developed a catalog of agile SDM malpractice called "agile smells." Although the catalog is a valuable contribution to identifying malpractices in other contexts, they did not link such malpractices to software developers' reasoning.

Table 1. Overview of existing SD and SDM literature reviews

Study	Study aim	Content addressing software developers' reasoning
Abrar et al. [27]	"[I]dentify the de-motivators while scaling agile at large, from management perspectives."	Some identified factors can lead to differences in reasoning.
Alqudah and Razali [29]	Review and compare scaled agile SDMs, identifying similarities and differences.	No
Campanelli and Parreiras [30]	"[E]valuate, synthesize and present aspects of research on agile methods tailoring."	No
Dingsøyr et al. [31]	"[D]elineate the conceptual structure underlying agile scholarship."	No
Dybå and Dingsøyr [21]	"[I]nvestigates what is currently known about the benefits and limitations of, and the strength of evidence for, agile methods."	Some identified papers have addressed why the SDMs have been adopted.
Gall and Pigni [32]	Develop a conceptual framework for DevOps.	No
Gandomani et al. [33]	Assess the relationship between agile SDMs and open-source SDMs.	No
Goldkuhl and Karlsson [34]	Investigate to what extent "research literature claiming to develop or adapt ISDMs" has adopted empirical inquiries.	No
Gutiérrez-Ríos et al. [35]	Suggest factors to consider when adopting agile and/or lean practices.	No
Hassan and Mathiassen [36]	Identify the field traditions by mapping canonical SD literature.	No
Hirschheim et al. [37]	Developed a framework showing that the SD field was rightly called a "fragmented adhocracy."	No
Iivari et al. [20]	Propose a framework for classifying SDMs in literature.	The framework could help to understand the rationale behind the SDM practices used.
Matalonga et al. [38]	Identify "factors that affect the adoption of agile practices in distributed projects."	No
Mihailescu and Mihailescu [39]	"Identify theoretical perspectives applied in the conceptualization of ISDM."	No
Moloto et al. [40]	Investigate "the impact that agile method use has on project success in organizations."	No
Onwujekwe and Weistroffer [28]	Survey literature on SD projects in the public sector and how the bureaucratic nature impacts the use of agile SDMs.	Addresses the fit between specific cultures and agile SDMs.
Sambamurthy and Kirsch [41]	"This paper reviews prior research on ISD processes and identifies the different types of contributions that have been made to our growing knowledge."	No
Telemaco et al. [22]	Define a catalog of agile smell "to denote the issues and practices that may impair the adoption of the agile approach."	They catalog malpractices.
Walters et al. [26]	Examine "the historical development of ISDMs."	No

Based on the above, we conclude that there is no existing literature review on software developers' type of reasoning behind SDM adoption and use. Consequently, we have limited knowledge about the frontline of research on this aspect of SDMs.

3. Research method

Our research method followed the systematic literature review [23, 42], where we surveyed existing SDM research referred to as primary studies. The identified need for a review is accounted for in the Introduction and Related research sections, and the specified research questions are found in the Introduction. Below, we explain how we identified research, selected primary studies, and extracted data from these studies.

3.1 Identification of research

Research on SDM adoption and use appears in conference proceedings and international journals. Therefore, we aimed for an inclusive selection of papers. We used the Scopus database, the largest "database of peer-reviewed literature" [43], which includes IEEE and ACM papers. The search included papers published until April 2022. In practice, the search started in 1972 when the earliest article was published in Scopus. Our search included journal papers, conference papers, and book chapters, regardless of the geographic region. Search fields include the paper title, abstract, and keywords. Appendix A shows the total list of search queries (72 combinations of keywords). The following is an example of one of the search queries: "Software Development Method" AND "Implementation." Our combination of keywords evolved based on our reading of papers, where we identified the need for additional keywords. For example, we noticed that some well-referenced papers on SDMs, which appeared in the reference list of retrieved papers, did not appear in our search results. We decided, therefore, to add keywords used in these missing papers to extend our search. One such example is adding the keyword "diffusion."

3.2 Selection of primary studies

We did not filter any results in Scopus, which meant that all generated results were included in our selection process. As shown in Figure 1, the selection process includes four stages. The first stage employed the search queries described above and resulted in a gross list of 3,227 papers, including duplicates. In the second stage, we removed duplicates, resulting in 1,619 unique papers. In the third stage, the list of papers was divided between the authors for a screening based on reading the abstract. We used the inclusion and exclusion criteria in Table 2, and papers were classified as "potentially relevant," "need to discuss with co-author," or "not relevant." The "potentially relevant" papers were fed into the next stage without further assessment. The "need to discuss with co-author"-papers were read once more by both authors, and they made decisions on the papers based on the criteria in Table 2. All conflicts were satisfactorily resolved during the joint assessment. We had a net list of 516 papers at the end of this stage.

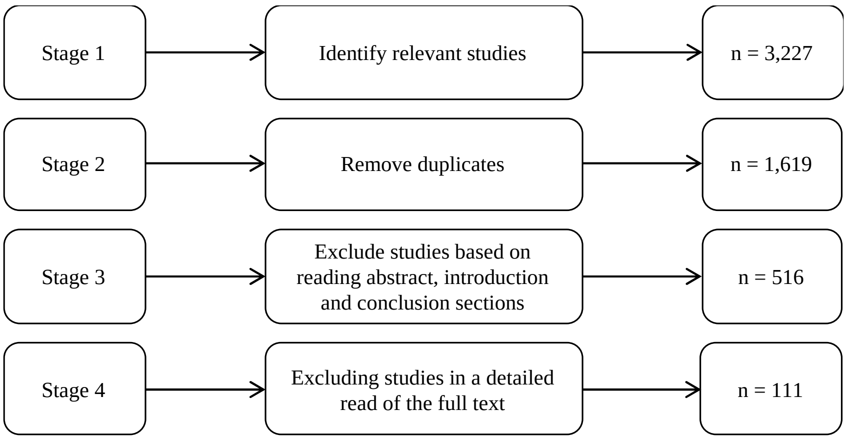

Fig. 1. Selection process

Table 2. Inclusion and exclusion criteria for primary studies

Inclusion criterion	Exclusion criterion
I1. The paper is written in English.	E1. The paper is an editorial.
I2. The paper is a peer-reviewed journal paper, conference paper, or book chapter.	E2. The paper is an introduction to a conference proceeding.
I3. The paper focuses on SDM as a study object.	E3. The paper is a literature review.
I4. The paper provides an empirical account, i.e., data, measurements, observations, or descriptions about adoption and/or use of SDMs.	

Papers classified as "potentially relevant" entered stage four, where both authors read and analyzed the full text of each paper. The details are provided in Section 3.3. If we found during this detailed analysis that a paper did not meet the inclusion and exclusion criteria in Table 2, it was excluded. As shown in Figure 1, we ended up with 111 papers included in the detailed analysis. The sharp reduction in potentially relevant papers was due to our inclusive search strategy. Thus, papers were included in the first dataset if they were related to SDMs. For example, an extensive number of non-empirical papers were included, papers that fall short of the inclusion criteria. Still, we chose the abovementioned search strategy to ensure we would not miss any papers by setting the search parameters too narrow.

3.3 Data extraction and analysis

The data extracted from each study were (1) Full reference (author, year, title, source name), (2) Source (journal, conference, book chapter), (3) Research question/aim, (4) Research method as classified by the authors of the paper, (5) Phases addressed (adoption, use), (6) Type of reasoning (rational, irrational, non-rational, unable to identify), and (7) Research themes addressed. The classification framework and procedures for extracting the more complex data (5 to 7) are explained in Sections 3.3.1 and 3.3.2. An overview of the assessed papers and the data extraction (1, 5, 6, and 7) is found in Appendix B; a summary is presented in Section 4.

We carried out an integrated thematic synthesis [25] to extract and analyze the data, i.e., we employed both a deductively generated framework and inductively generated codes to organize the extracted data. Cruzes and Dybå [25] argue that the integrated approach "is the most relevant in systematic reviews as they tend to be done on the basis of the theoretical interests guiding the research questions of the review." In our case, the integrated approach enabled us to be guided by our interest in software developers' reasoning and at the same time explore different directions that existing research has taken.

3.3.1 Framework to extract phase and type of reasoning addressed in primary studies

To address to what extent software developers' reasoning behind SDM adoption and use has been investigated in previous research, we need to classify the phases and the type of reasoning addressed in the primary studies. Therefore, both authors actively devised the framework shown in Table 3; the framework was reviewed informally by an external reviewer before we started our data extraction [44].

A paper could address the SDM adoption, SDM use phases, or both. Vavpotič and Hovelja [45] argue that research on SDM adoption often focuses on how and why the SDM or some of its parts are spread among potential software developers. Aligning with this view, we define SDM adoption as one or several SDMs (or parts thereof) that have been introduced into an organization or an SD team as a new way of working with SD. Regarding SDM use, we draw on Fitzgerald et al. [46] and the focus on how the software developers apply the SDM. Thus, we define SDM use as a selection or all activities of one or several SDMs that software developers have applied to develop an information system. These two definitions are ideal phases related to SDMs in organizations, although the phases often overlap in practice. An organization's adoption of an SDM means that software developers gradually start to use the method, and eventually, the adoption ends, and regular use takes over. Also, we use an inclusive definition of software developer, referring to individual software developers', an SD team's or an organization's adoption and use of SDM.

Our classification of types of reasoning takes Weber's [24] work on social actions as a point of departure. According to Weber [24], it is possible to distinguish between two types of social actions: rational and non-rational. Still, the existence of rational actions also makes it relevant to discuss irrational actions [47]. Compared to rational social action, irrational ones are flawed reasoning, resulting in, for example, misconceptions or miscalculations. In both rational and irrational reasoning, it means that such primary studies analyze arguments behind the executed activities and compare them with a reference point, i.e., the claimed method rationale. Finally, non-rational social actions are not anchored in any method rationale. Instead, these actions are based on deeply rooted habits where the software developer no longer considers rational reasons or where software developers are driven by emotional conditions [24].

Table 3. Classification of the type of reasoning addressed in the primary studies

	Phase	
Type of reasoning	**SDM adoption**	**SDM use**
Rational	Researchers in the primary study analyze whether the choice and execution of adoption activities align with the rationale (e.g., goals) of the chosen adoption approach and the adopted SDM.	Researchers in the primary study analyze whether the choice and execution of SD activities align with the rationale (e.g., goals) of the chosen SDM or some parts of this method.
Irrational	Researchers in the primary study analyze whether the choice and execution of adoption activities deviate from the rationale (e.g., goals) of the chosen adoption approach and the adopted SDM.	Researchers in the primary study analyze whether the choice and execution of SD activities deviate from the rationale (e.g., goals) of the chosen SDM or some parts of this method.
Non-Rational	Researchers in the primary study analyze whether the adoption activities are chosen and executed without awareness of the rationale (e.g., goals) of the chosen adoption approach and the adopted SDM.	Researchers in the primary study analyze whether the SD activities are chosen and executed without awareness of the rationale (e.g., goals) of the chosen SDM or some parts of this method.
Unable to identify	Researchers in the primary study analyze activities to adopt an SDM. However, the analysis does not attempt to connect these activities to the rationale (e.g., goals) of the chosen adoption approach and the adopted SDM.	Researchers in the primary study analyze SD activities. However, the analysis does not attempt to connect these activities to the rationale (e.g., goals) of the chosen SDM or some parts of this method.

The first row of Table 3 focuses on rational reasoning, where the social actions are based on rational reasoning and result, in our case, in adhering to the intended goals with either the SDM adoption or SDM use. Rational reasoning means the primary study has investigated whether the reasons behind the executed activities align with the claimed rationale. For example, if software developers claim to use Daily Scrum meetings, the claimed method rationale is "to create a shared understanding of the project status, create short term planning and achieve better interaction within the ISD team" [48]. Thus, rational reasoning is about choosing and executing activities that support these ends, such as discussing what the team members will do today. The second row in Table 3 addresses irrational reasoning. In these cases, the primary studies have investigated software developers' misconceptions or miscalculations in the SD adoption or SDM use and the misalignment with the claimed method rationale. For example, identifying the use of the Daily Scrum Meeting as a problem-solving meeting and software developers' arguing that this activity supports the ends of such a meeting.

The third row in Table 3 addresses non-rational reasoning. Primary studies are classified as non-rational when they have addressed SDM adoption or SDM use without the software developers being aware of the reasons for their activities. For example, in the case of SDM adoption, the researchers of the primaru studies have addressed whether the adoption activities are chosen and executed without awareness of the rationale of the chosen adoption approach and the adopted SDM. Finally, the fourth row in Table 3 contains the category "Unable to identify." Studies fall in this category when the analyzed primary study has addressed SDM adoption and use without investigating any connection between executed activities and any type of reasoning (rational, irrational, or non-rational).

3.3.2 Extraction of phases, type of reasoning, and research themes in primary studies

We carried out the integrated thematic synthesis using the five steps put forth by Cruzes and Dybå [25] to extract phases, types of reasoning, and research themes in the existing research. Below we discuss our tailored version of these steps, highlighting the deductive and inductive parts.

Step 1: Extract data. We familiarized ourselves with the existing research by individually reading the papers. We focused on the context descriptions (SDM and settings) and what type of analysis of the SDM adoption and/or use that existing research addressed. We made brief notes about these aspects. We also captured data extracts, i.e., quotes or text sections in the papers, that contained ideas for our characterization of the research themes. All extracted data and notes were organized in a spreadsheet.

Step 2a: Code data – deductively: We individually read the papers once more to extract the type of reasoning the primary studies addressed (if any) when investigating software developers' adoption and use of SDMs. We used an iterative approach to classify shared subsets of papers using the framework in Table 3. We discussed our individual classifications and together made a final classification. By executing the analysis on subsets of papers, we could learn from each other and develop a shared understanding of how to use the classification framework [44].

Step 2b: Code data – inductively. When reading the papers in Step 2a, we also created initial inductive codes for each of the papers included in the three topmost rows in Table 3. The reason for not creating inductive codes for papers classified as "Unable to identify" is our specific interest in the type of reasoning behind software developers' adoption and use of SDMs. One example of our inductive coding is the paper by Häggmark and Ågerfalk [10], which received the codes "Different rationalities behind method use" and "How method has been adjusted." Thus, we allowed ourselves to associate a paper with one or more codes because papers can have multiple purposes. Like step 2a, our work was highly iterative, comparing the codes generated by our individual reads of the papers. The purpose was to improve our reading and develop a shared understanding of the initial codes for the upcoming search for research themes [44]. The spreadsheet documented our collection of inductive codes, linking them to the extracted data. We also added additional quotes or summaries when necessary.

Step 3: Translate inductive codes into themes. We used our collection of inductive codes from Step 2b as input for workshops in which both authors participated. These workshops aimed to develop a set of themes, so we iterated our codes and searched for what they had in common. We organized our research themes using mind maps and traced how the inductive codes could potentially be linked to research themes. These mind maps allowed us to scrutinize the congruence of the research themes and move codes when necessary. For example, the inductive code "Different rationalities behind method use" introduced above was included in the research theme "Competing rationalities." It again meant an iterative work pattern, constantly comparing inductive codes, data extracts, and candidate research themes. The iterative work resulted in both moving inductive codes and collapsing research themes into each other to better reflect the underlying papers.

Step 4: Create a model of higher-order themes. Given our interest in different types of reasoning behind SDM adoption and use, we combined the identified research themes with our deductive classification of papers in Step 2b. Consequently, this meant identifying if and to what extent different types of reasoning had been studied for the identified research themes. Thus, we created higher-order themes, or patterns, for each research theme. Furthermore, we also settled on a description of each research theme that captured its essence; it meant summarizing the papers and providing coherent and internally consistent accounts. For the research theme exemplified above, we decided to keep the name "Competing rationalities," which we defined as how software developers deal with and prioritize conflicting rationalities during the adoption and use of SDMs.

Step 5: Assess the trustworthiness of the synthesis. Both authors reviewed the research themes during a second set of workshops. First, we analyzed the fit between the claims made about the research themes and "what the evidence shows" [25], i.e., what the data extracts show. Sometimes, it also meant returning to the actual papers and rereading paragraphs. Second, we analyzed the fit between the research themes and the overall focus of our analysis. We made minor adjustments to the synthesis; a couple of papers were moved to create a better fit. Completing our running example discussed above, we identified eight papers for the candidate research theme "Competing rationalities."

4. Results

In this section, we present a summary of our systematic literature review, structured according to our two research questions. The first subsection addresses to what extent the reasoning behind SDM adoption and SDM use has been investigated in previous research. In the second subsection, we detail the research themes and how reasoning – rational, irrational, and non-rational – behind SDM adoption and use has been addressed in existing research. The overview of the analysis is found in Appendix B.

4.1 RQ1 – To what extent has previous research investigated software developers' reasoning behind SDM adoption and use?

The analysis of the extent to which research has addressed the reasoning behind SDM adoption and use is presented in Figure 2 using a bubble chart. A bubble chart shows three dimensions of the data. The horizontal axis contains the two phases, SDM adoption and use. The vertical axis shows whether a primary study has addressed any type of reasoning behind the studied activities. The bubble size is proportional to the frequency of primary studies in the pair of categories corresponding to the bubble coordinates.

Starting on the horizontal axis in Figure 2, we see that SDM adoption has received more research attention than SDM use, 54.5 percent versus 45.4 percent. In addition, it should be noted that eight primary studies address both SDM adoption and use. The vertical axis in Figure 2 shows that most primary studies (76.4 percent) do not investigate software developers' reasoning behind SDM adoption or SDM use. Thus, 23.6 percent of all studies addressed software developers' reasoning behind their SDM adoption or SDM use.

The intersections in Figure 2 show that the largest share of studies addresses SDM adoption without focusing on the reasoning behind these activities. These primary studies give accounts of activities and experiences from adopting one or more SDMs without any inquiry into the reasoning during the adoption. Thus, these studies do not analyze how the software developers' reasoning aligns with the rationale of the adoption approach and the adopted SDM. We identified studies addressing, for example, diffusion of SDMs [e.g. 45, 49, 50], challenges with SDM adoption [e.g. 51, 52, 53], factors that affect SDM adoption [e.g. 54, 55, 56] and experiences of SDM adoption [57-59].

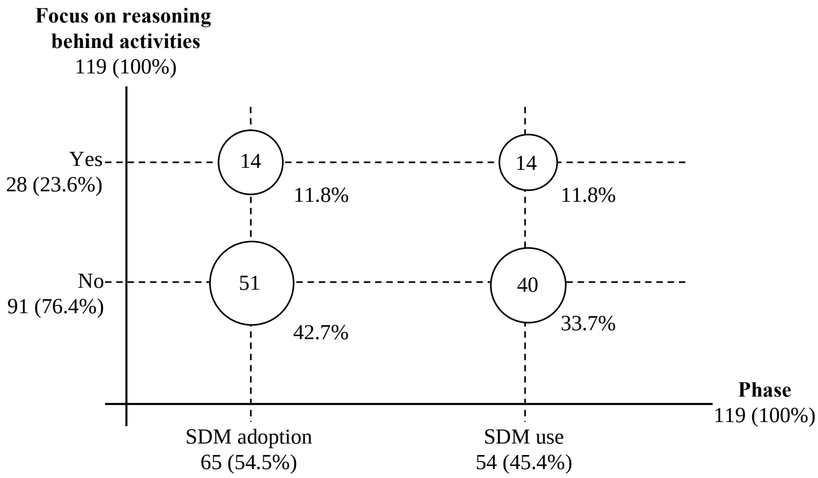

Fig. 2. The focus of existing research on the reasoning behind SDM adoption and use

Note: Please note that papers can address both SDM adoption and use, which means that the total number of primary studies that address the phases exceeds the actual number of primary studies.

The next largest share of primary studies addresses SDM use without focusing on the reasoning behind the SD activities. Topic-wise, these studies are similar to the SDM adoption papers; they give accounts of activities and experiences from using one or more SDMs. We identified studies addressing what SDMs are used [e.g. 60, 61, 62], the experience of SDM use [e.g. 63, 64, 65], non-rigors use of SDMs [e.g. 66, 67, 68], and factors that affect SDM use [69, 70]. Yet again, these studies do not include any analysis of how the software developers have reasoned about the SDM during use and how their reasoning aligns with the rationale of the used SDM.

Turning our attention to the research that has addressed the reasoning behind SDM adoption and use, as the vertical axis in Figure 2 shows, we identified 28 primary studies. The number of primary studies that analyzed software developers' reasoning behind SDM adoption and use is similar; 14 papers addressed SDM adoption, and 14 papers addressed SDM use. In these primary studies, researchers have identified some method rationale that has been used as a reference point for analyzing SDM adoption and/or SDM use. Thus, these reference points made it possible for us to analyze what types of reasoning have been addressed. We scrutinize these papers in more detail in the next sub-section.

4.2 *RQ2 – What research themes have previous research investigated concerning software developers' different types of reasoning behind SDM adoption and use?*

Table 4 presents an overview of how the 28 primary studies were distributed across the research themes, two phases, and types of reasoning. It is important to note that primary studies can address multiple research themes and types of reasoning; however, we did not find any primary studies that addressed both phases. The bottom row in Table 4 shows that we identified 21 primary studies addressing rational reasoning, 11 primary studies addressing irrational reasoning, and three primary studies addressing non-rational reasoning. Below we present our analysis using the identified research themes. We present how the different types of reasoning have been addressed in combination with SDM adoption and use for each research theme.

Table 4. Overview of research themes and types of reasoning

		Type of reasoning			
Research theme	**Phase**	**Rational**	**Irrational**	**Non-rational**	**Total**
Choice of SDM	Adoption	5	0	0	5
Competing rationalities	Adoption	1	3	0	8
	Use	4	1	0	
Method deviation	Adoption	0	0	2	10
	Use	6	2	0	
Understanding operationalization	Adoption	2	1	0	7
	Use	1	2	1	
Achieved contribution	Adoption	0	3	0	5
	Use	2	0	0	
Total		21	11	3	35

4.2.1 *Choice of SDM*

The research theme, Choice of SDM, captures existing research focusing on how the rationale of the chosen SDMs fits the situation at hand. Each SDM comes with its own rationale, and this rationale can align with what the software developers want to achieve to varying degrees. A well-suited choice means an SDM that provides support, whereas an ill-suited SDM provides less support or, in the worst case, creates obstacles. Table 4 shows that five primary studies belong to this theme. All these studies address SDM adoption and rational reasoning behind the choice of SDMs [71-75]. Thus, existing research has not focused on investigating misconceptions made when choosing methods or, if choices are made, without awareness of the chosen SDM's rationale.

SDM adoption – rational reasoning. Vejandla and Sherrell [71] described practitioners' experiences from adopting eXtreme Programming and test-driven development. For example, they described how the goals of test-driven development fit the situation at hand, i.e., providing means-end reasoning. Mishra and Mishra [72] illustrated how agile methods could be adopted in complex SD projects. They gave an account of the choices made regarding the combination of method parts and how these choices were consciously anchored in the goals behind the SDMs. Taylor et al. [73] illustrated how risk assessment could be combined with the rationale behind agile practices to guide the adoption of an SDM and to create a situational fit. Tripp and Amstrong [74] carried out a high-level analysis of how agile practices were matched with organizations' intentions to improve software quality and improve SD efficiency and effectiveness. They argued that a misfit between intentions and the agile practices utilized might decrease the overall success of the adoption. Finally, Alt et al. [75] developed an agile DevOps framework and described the goals they wanted to reach. They described a transformation plan, consistent with these goals, and provided lessons learned from the method transformation. Based on the transformation process, they identified both success factors and challenges, which includes an awareness of why the different practices were selected and adopted.

4.2.2 Competing rationalities

The research theme, Competing rationalities, includes primary studies that address how software developers deal with and prioritize conflicting rationalities during the adoption and use of SDMs. These studies show that SDM adoption and use take place in a context. Software developers interact with other parts of the organization or with other organizations, which means that there are other goals and values at play at the same time. More importantly, sometimes, different goals and values compete, and software developers must choose between them. Thus, software developers' choices can be rational, irrational, or non-rational in relation to the goals and values they prioritize. As shown in Table 4, eight primary studies are found in this research theme, divided across rational and irrational reasoning behind SDM adoption and rational and irrational reasoning behind SDM use.

SDM adoption – rational reasoning. We identified one primary study analyzing rational reasoning when competing rationalities were encountered during SDM adoption. Sauer and Lau [76] analyzed how different rationalities come into play in SDM adoption. They highlight the role of business managers in bringing the adoption of the Structured Systems Analysis and Design Method (SSADM) to a halt, where business pressure influenced and overrode the SDM goals. Although software developers were aware of the potential contribution that the SDM could make, they did not persist in pursuing the started adoption activities. The study shows software developers' awareness of the reference point and activities that indicated an understanding of the SDM and its potential contributions. However, because software developers and business managers have competing rationalities, the SDM adoption did not unfold as planned.

SDM adoption – irrational reasoning. We found two primary studies that focus on irrational reasoning concerning competing rationalities and SDM adoption [77, 78]. McAvoy and Butler [77] examined the failure to change from a traditional SDM to an agile SDM. They analyzed how software developers' initial commitment to user stories changed over time, going from commitment to resisting the change. They found that loyalties to the team, in this case, hierarchical groupthink and a need to please the project managers, made the team members deviate from the rationale of user stories. In the end, the team abandoned its adoption of user stories. McAvoy and Butler [78] studied factors needed to succeed with the implementation of method values and how learning occurred in the introduction of an SDM. They found that even though software developers individually understood and agreed on the values of a selected method part, power factors in the team made them deviate from the SDM when introduced. Thus, these power factors resulted in executed activities not being in line with the stated goals.

SDM use – rational reasoning. We found four primary studies that address rational reasoning when competing rationalities were encountered during SDM use [10, 79-81]. Strode et al. [79] studied the extent to which the organizational culture (values and assumptions) matched the values of used agile methods. They built on the idea "that the organizational culture in which the agile method is embedded could have an impact on its use." Their study suggested that organizations with an external focus and preferences for flexibility align with the rationale of agile SDMs. Dennehy and Conboy [80] showed how different interests can be a source of tension when software developers are using an SDM and when introducing tools to support features of the SDM. Häggmark and Ågerfalk [10] did an "empirical enquiry into why engineers do not keep to the principle of separating business logic from display." These software developers provided arguments for the tailored version of the assessed SD principle, prioritizing other goals than prescribed by the SDM. Mohan et al. [81] have studied individuals' attachment to and identification and involvement with prescribed SDM activities, i.e., an inquiry into software developers' buy-in to the rationale behind SDMs. They divided SDM use into three categories, committed use, compliant use, and resistant use, and investigated how an SDM's relative advantage, complexity, and combability affect these use categories. For example, they found a positive effect of combability, i.e., that the SDM is consistent with the "social cultural values, and past experiences of potential users" on committed use. Thus, if the rationale of the SDM fits the cultural values of the organization, the likelihood increases that actual execution will build on and be in line with the rationale of the SDM.

SDM use – irrational reasoning. We identified one primary study that analyzed irrational reasoning. Mohan et al. [81] found that software developers' attachment to and identification and involvement with the prescribed SDM activities could result in committed use and resistance use. When software developers feel inequity in executing the prescribed SDM activities, they oppose them. They found that the less the compliance between the chosen SDM and the cultural values in the organization, the more likely is resistance use. In other words, it might result in software developers resorting to other ways of working that are not in line with the rationale of the chosen SDM.

4.2.3 Method deviation

The research theme, Method deviation, organizes studies that identify and explain deviations from the adoption approach and the adopted SDMs, or the used SDM. These studies show that software developers do not introduce or apply SDMs as they are presented in the textbox. Thus, the enacted versions deviate from what is described. Studies in this research theme go beyond identifying the deviations as such; they address how the deviations relate to the rationale of the SDM. As shown in Table 4, ten primary studies belong to this research theme, covering non-rational SDM adoption, rational SDM use, and irrational SDM use.

SDM adoption – non-rational. We found two primary studies that addressed non-rational reasoning [82, 83]. Patanakul and Rufo-McCarron [82] identified six challenges in an agile implementation program. One of these challenges was change management. They concluded that effective change management is needed to avoid actors sticking to a specific mindset that can lead to destructive behaviors where the rationale of the practices is not met. Berger and Beynon-Davies [83] studied the unbundling, i.e., adoption, of an SDM in a large-scale development project. In this public-sector project in the UK, an external company suggested using "their own in-house commercial RAD-like development approach" during SD work in the client organization. Although, over time, the adoption resulted in the successful diffusion of "certain SDM practices with the case organisation," the adoption itself was problematic. Berger and Beynon-Davis found that the actors paid limited attention to how the SDM rationale differed from culture in the case organization, showing an unawareness of the fit of the SDM and its rationale. For example, the organizational culture stressed "individuality and individual accountability," which made organizational stakeholders unwilling to commit to fast decision-making in the iterative development process prescribed by the adopted SDM.

SDM use – rational reasoning. We identified six papers that addressed rational reasoning behind method deviation during SDM use [10, 11, 84-86]. Heikkilä et al. [11] studied the phenomenon of "ScrumBut," which refers to mismatches between the prescribed and the actual use of Scrum. They identified a couple of deviations from how Scrum is prescribed, and some of these were conscious improvements to Scrum in line with the goals of the SDM. Häggmark and Ågerfalk [10], discussed above, studied how software developers deviated from the SD principle of separating business logic from display. They found that some of the software developers "had a more conscious departure from the strict tier-separation, with clear arguments of why they did what they did." Thus, the software developers showed awareness of the principle and its goals. At the same time, they provided arguments for their deviation showing a conscious means-end decision to tailor the SDM.

In their study of SSADM, Middleton [84, 85] and Middleton and McCollum [86] concluded that, in most of the projects they investigated, the SDM was tailored "beyond recognition" and altered the SDM's foundation. However, in some of these projects, the modifications resulted from limitations in the SSADM and not a lack of understanding of the method rationale of the SDM. For example, Middleton [84, 85] described SSADM's type of user involvement as superficial, and, therefore, modifications were made to the SDM to reach the user involvement goals. Thus, these modifications were based on rational reasoning, where the software developers moved beyond the rigid use of the SDM. Chen et al. [87] analyzed the use of a modified version of Scrum that implemented unified modeling language diagrams. Thus, they give an account of a planned method deviation and how the used SDM aligned with the modified rationale.

SDM use – irrational reasoning. We found two papers that captured irrational reasoning behind method deviation during SDM use [11, 88]. In their study of ScrumButs, discussed above, Heikkilä et al. [11] identified changes to two Scrum practices that could not be fully explained by a real need and a deep understanding of what makes Scrum work. For example, they found user stories that were active over several sprints because of competence issues and not planning according to the sprint's actual resources. Thus, such activities contradict the intended goal of the SDM. Saarinen [88] studied how prototyping strategies had been used and whether use in actual SD projects was in line with the high-level recommendations of the method. He found, for example, that prototyping strategies had been employed when requirements were easy to specify. He also found that prototyping strategies were used regardless of the uncertainty level, which is inconsistent with high-level recommendations.

4.2.4 Understanding operationalization

The research theme, Understanding operationalization, captures primary studies that address the driving actors' understanding of how the SDM activities implement the rationale of the SDM. These studies focus on software developers' awareness and understanding of the method rationale plays a role in how the SDM adoption and use unfolds. It is shown that software developers both show awareness of the chosen SDM's rationale, how the chosen activities align with the rationale, and where misconceptions result in misalignment. Furthermore, existing research on this theme also shows that software developers sometimes are unaware of why they carry out activities and how they relate to the claimed SDM's rationale. As shown in Table 4, this theme consists of seven primary studies, which address rational, irrational behind SDM adoption and rationale, irrational, and non-rational reasoning behind SDM and use.

SDM adoption – rational reasoning. In total, we identified two primary studies that analyzed rational reasoning during SDM adoption [89, 90]. Andersson and Nilsson [89] studied several SDM adoption projects in a large SD organization. They assessed whether the adoption projects paid attention to the rationale of the chosen SDM and to what extent the rationale was used for monitoring and evaluation. The authors drew lessons learned from the mixed results in the studied projects. They found that projects defined clear goals but differed in communicating and evaluating them. Consequently, they concluded that these latter activities are important to improve future SDM adoptions. To improve adoption, Hazzan and Dubinsky [90] studied how the awareness of the principles behind an SDM was used by a "task force" to introduce an SDM lifestyle in an organization. Thus, the task force used the underlying goals and values of the SDM in a conscious design of the adoption process, showing an understanding of the method rationale.

SDM adoption – irrational reasoning. We found one primary study that analyzed irrational reasoning during SDM adoption. The previously discussed study by Andersson and Nilsson [89] on SDM adoption came across irrational activities. In their assessment of how adoption projects' paid attention to the rationale of the chosen SDM, they found that two out of the four studied projects failed due to deviations from stated goals, i.e., by executing irrational activities. Although these failed projects had clear goals, they were not communicated and resulted in deviation from the rationale of the SDM adoption.

SDM use – rational reasoning. We found one primary study that analyzed rational reasoning behind SDM use. Mäki-Runsas [91] studied software developers to identify potential SDM cargo cult behavior. She investigated the extent to which the executed activities in an SD team's Daily Scrum Meetings corresponded with the intended goals of such meetings. She found that the software developers focused on what each developer had done, current hindrances, and what they would do in the current day. Thus, these activities were in line with the SDM's rationale.

SDM use – irrational reasoning. We identified two primary studies that analyzed irrational reasoning during SDM use [91, 92]. Mnkandla and Dwolatzky [92] studied an SD organization that tried to adhere to a set of agile principles, although they did not follow a specific SDM. Although the project was, to some extent, found successful, it was not due to these principles. They found that the organization's way of working, in some cases, contradicted the agile principles. Thus, the analysis showed a lack of understanding of the agile principles; that is, the actual implementation of these principles can be viewed as irrational. In Mäki-Runsas' [91] study about potential SDM cargo cult behavior, discussed above, she also addressed irrational activities. In her analysis of an SD team's Daily Scrum Meetings, she found irrational activities due to malpractices. It means that the SD team carried out these meetings incorrectly in relation to the goals. For example, the SD team used Daily Scrum Meetings to "discuss reports from a long-term perspective or discuss technical solutions."

SDM use – non-rational reasoning. We found one primary study that addresses non-rational reasoning. Mäki-Runsas [91] identified activities in an SD team's Daily Scrum Meetings that were non-rational. These non-rational activities were old habits that the SD team had kept after the studied organization had adopted Scrum as their way of working. For example, the Scrum master informed about and introduced new requirements from upper management during Daily Scrum Meetings. Thus, these new requirements changed priorities "in the middle of an ongoing sprint"; the activity could be traced back to unawareness of continuing old practices.

4.2.5 Achieved contribution

The research theme, Achieved contribution, includes primary studies that focus on to what extent the potential contribution of a chosen SDM has been reached during adoption and use. These studies relate the rationale of the SDM to the achieved results and assess to what extent and how the introduction or applied activities have contributed or not. As shown in Table 4, this research theme includes five primary studies that addressed irrational reasoning behind SDM adoption and rational reasoning behind SDM use. Thus, studies on SDM adoption have focused on why goals are not reached, whereas studies on SDM use have focused on the customization of the SDM and its method rationale.

SDM adoption – irrational reasoning. We identified three primary studies that analyzed irrational reasoning behind SDM adoption [82, 93, 94]. Roberts Jr et al. [93] found SDM practices that contributed less than expected. They argued that organizational actors do not always align with the stated goals. Instead, passive resistance misuse was mixed into the SDM adoption, which means deviating from the rationale of the SDM. Therefore, they called for more research that "move[s] beyond the rational model" of SDM adoption. Mohallel and Bass [94] found that some SD teams had difficulties adhering to the adopted practices' agile principles, which resulted in the SDM not contributing as expected. For example, they identified a "lack of cadence in sprint planning meetings, inadequate use of effort estimation and product quality issues." Thus, they found signs of irrational SDM adoption, where teams chose dynamic sprint lengths without understanding the consequences of such a choice, for example, its impact on estimates. Patanakul and Rufo-McCarron [82] identified six challenges in the transition from traditional SDMs to agile SDMs. One reason present in several of these challenges is that those project actors did not fully understand what the introduced practices could contribute to, i.e., the goals. They conclude that it can lead to destructive behaviors, where the intentions with the practices are not met and executed activities do not follow the SDM's rationale.

SDM use – rational reasoning. We identified two primary studies that addressed rational reasoning behind SDM use [95, 96]. Soundararajan et al. [95] investigated how a customized agile SDM contributes to organizational objectives. Therefore, they proposed a framework for this type of analysis using three concepts: adequacy, capability, and effectiveness. Regarding rational reasoning, two of these concepts, adequacy and effectiveness, are interesting. Adequacy means an assessment of the "sufficiency of the method with respect to meeting its stated objectives," i.e., a possibility to analyze how tailored instances of SDMs meet its stated objectives. Effectiveness addresses "how well the adopted method actually achieves its stated objectives," and the concept was used to assess the achievements of stated objectives. The framework has been implemented as a survey, and preliminary findings were presented on the adequacy of SDMs. Clark et al. [96] presented a modified version of Scrum to infuse medical research into the SD process. They provided a rationale for the added method steps, showing what type of additional understanding the new steps contribute. Furthermore, they provided an account for their use of the added method steps that align with the provided rationale.

5. Discussion

SDMs have been on the research agenda for several decades [97]; however, there is no systematic literature review on software developers' types of reasoning behind adopting and using SDMs. Although some existing literature reviews have resulted in different types of SDM frameworks [e.g., 20, 39, 98], none include or address the reasoning aspect. Thus, our study contributes to a systematic overview of primary studies on this part of the SDM domain.

5.1 RQ 1 - To what extent has previous research investigated software developers' reasoning behind SDM adoption and use?

In our analysis of existing research on SDM adoption and use, we identified 111 empirical SDM studies. Of these studies, 28 papers addressed software developers' reasoning behind SDM adoption and use. Consequently, as shown in Figure 2, this group of papers constitutes a significant part of the knowledge base, with an equal focus on adoption and use. Still, the details in Table 4 show that most of these studies have investigated rational SDM adoption and use; much less attention has been given to irrational types of activities, and almost no attention has been given to non-rational activities. This finding is somewhat surprising and unexpected since SDM research has such a long history, and SDM use was identified as a challenging area very early [26].

In most cases, the choice to adopt and use SDMs is associated with significant organizational investments. Such investments should lead to interest among researchers and practitioners in understanding software developers' irrational and non-rational SDM adoption and use. In addition, it is a well-known fact that software developers use their situational knowledge when adopting and using SDMs [46], which means that SDMs are not followed literally. However, research on the adoption and use of SDMs seems to have considered rational activities predominantly, something that has not been evident from existing literature reviews on SDMs. For example, Abrahamsson et al.'s [99] review focused on successful stories, making it reasonable that they did not address irrational and non-rational SDM adoption and use activities.

Taken collectively, our results show that existing empirical research on SDM adoption and use provides limited advice and guidance on identifying and dealing with irrational and non-rational adoption and use activities. The rather limited attention to these types of activities indicates that previous research might have missed one important aspect when analyzing successful as well as flawed SDM adoption and use. Indeed, at the general level, this finding points toward a knowledge gap in empirical SDM research and provides an important opportunity for future research: to explore irrational and non-rational SDM adoption and use activities. Such research is important for understanding and improving future advice to practitioners on how to adopt and use SDMs.

We do not claim that stressing the importance of understanding the rationale behind SDMs is new in SDM research. In their literature review of SDMs, Iivari et al. [20] discussed the importance of understanding the philosophy of the chosen SDM to be able to reach the method's intended goals and values. In addition, they discussed how their framework could help practitioners understand the rationale behind the SDM practices they are using. Otherwise, the software developers will have a "professional blind spot" [20] and thus run the risk of irrational or non-rational SDM adoption and use. However, our results show that researchers have made limited contributions to knowledge about the existence of this professional blind spot in practice and its consequences on SDM adoption and use. Of course, SDMs have advanced since Iivari et al. [20] compiled their framework, making the covered SDM content outdated. Nevertheless, the fundamental idea of understanding and working in line with the chosen SDM's rationale is still valid.

5.2 RQ2 – What research themes have previous research investigated concerning software developers' different types of reasoning behind SDM adoption and use?

Our thematic synthesis of existing research revealed five research themes in previous research: 1) Choice of SDM, 2) Competing rationalities, 3) Method deviation, 4) Understanding operationalization, and 5) Achieved contribution. As shown in Table 4, four of these research themes – Competing rationalities, Method deviation, Understanding operationalization, and Achieved contributions – span both the adoption and use of SDM. Table 4 also shows a detailed view of how the three types of reasoning have been investigated across the identified research themes. We found that all the research themes have covered rational reasoning, which is not that surprising given our discussion about SDM adoption and use being predominantly treated as rational activities. A more scattered pattern is found in analyses of software developers' irrational and non-rational reasoning. Thus, these results indicate that the research community knows less about software developers' irrational and non-rational reasoning behind SDM adoption and use, where a knowledge gap in one research theme can impact the understanding of another research theme.

To identify gaps and develop them into tangible opportunities for future research, we outline a series of theoretical relationships in Figure 3. Thus, the model in Figure 3 proposes new links between the research themes and highlights specific future research directions. To aid in framing these directions for researchers, we further develop a series of possible broad research questions that could be explored.

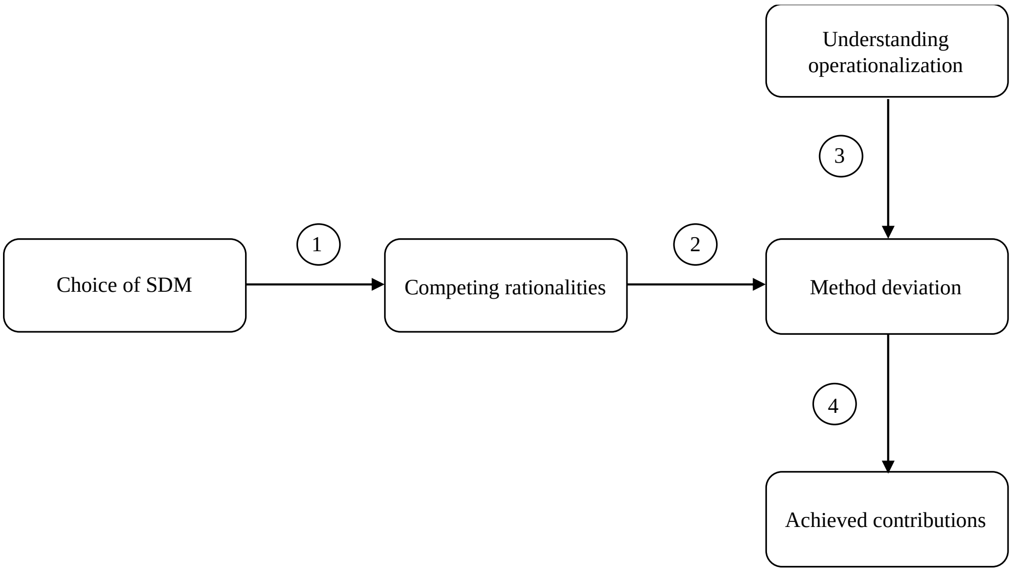

Fig. 3. Identified relationships between the research themes

Relationship #1: Consider how the rational, irrational, or non-rational choice of SDM impacts competing rationalities in SDM adoption and use. The research theme, Choice of SDM, focuses on how software developers connect the SDM rationale to the organizations' goals, which becomes the motives for choosing a particular SDM or part thereof. When choosing an SDM to adopt, the rationale of the method must fit the organization and the situation at hand. Thus, existing research has focused on investigating these choices as rational decisions [71-75]. However, if the rationale of the chosen SDM does not align with organizational goals, the method will add conflicting rationalities that have to be resolved later [e.g., 76, 79]. Our results show that we lack knowledge of whether and to what extent software developers make irrational choices, where the SDM rationale will not contribute to the organizational goals, or make non-rational choices, where there is now awareness if the SDM rationale aligns with the organizational goals. An opportunity exists to develop an increased understanding of how rational, irrational, or non-rational SDM choices impact competing rationalities in SDM adoption and use. Future research could, therefore, consider questions such as: To what extent do rational, irrational, or non-rational SDM choices lead to competing rationalities in organizations?

Relationship #2: Consider how competing rationalities in SDM adoption and use drive rational, irrational, or non-rational method deviation. Existing studies in the research theme of Competing rationalities show how software developers address and prioritize conflicting goals and values during the SDM adoption [76] and use [10, 79-81]. In order to manage such conflicts, first of all, it requires awareness of them, and second, that chosen adoption and use activities align with the prioritized rationale, i.e., to implement the choices rationally. Moreover, when software developers are placed in situations where they must prioritize between different goals, the result might not be what the organization expects [77, 78, 81]. Our results show that the Competing rationalities research theme focuses on how to deal with conflicting rationalities as rational and irrational choices. Thus, less attention has been given to unawareness of these conflicts that result in non-rational choices. At the same time, we found studies in the Method deviation research theme showing that rational [10, 11, 84-86], as well as irrational [11, 88] and non-rational [82, 83], deviations from the chosen SDM are made. We do not argue that these two research themes have been researched independently; the opposite is shown in, for example, Häggmark and Ågerfalk [10]. Still, there is a need to develop a deeper understanding of competing rationalities as a driver for SDM deviations, in particular, related to irrational and non-rational method deviations. Therefore, future research could pose questions such as: How do software developers deal with competing rationalities doing rational, irrational, and non-rational tailoring of SDMs?

Relationship #3: Consider how software developers understand SDM impacts method deviations. Existing studies in the research theme Understanding operationalization contribute to an increased understanding of how software developers operationalize SDM activities to achieve claimed goals. Our findings show that, in the research theme Understanding operationalization, existing research has covered all three types of reasoning, i.e, rational [89-91], irrational use [91, 92], and non-rational [91] reasoning. Thus, such results provide an important understanding of why software developers deviate from SDMs [10, 11, 84-86]; that is, there is an important relation to the Method deviation research theme. Although research in Understanding operationalization covers all three types of reasoning, we still see opportunities to research how the software developers understanding of the SDMs impacts method deviations. Because if software developers can avoid misconceptions, they can potentially reduce irrational method deviations. Such research could pose questions such as: How does software developers' understanding of SDM contribute to rational, irrational, and non-rational method deviations during adoption and use?

Relationship #4: Consider how consequences of rational, irrational, and non-rational method deviations impact achieved contributions. At the end of the day, the achieved contributions from adopting and using SDMs are important since organizations adopt and use SDMs to improve their SD. We found that studies in the Achieved contribution research theme covered rational [82, 93, 94] and irrational [95, 96] reasoning. We could not identify research investigating non-rational aspects of the achieved contribution. At the same time, it is not an extensively researched theme. Moreover, the achieved results should depended on the method deviations made [e.g., 10, 11, 88]. Of course, we do not argue that all method deviations result in negative contributions and that SDMs should be used rigorously. Such a view would be at odds with existing research on how SDMs are used [6-8]. Instead, and as shown in Figure 3, we suggest that more complex mechanisms are at play – competing rationalities and understanding operationalization – leading to (lack of) the achieved contributions; these mechanisms are materialized in method deviations that can be rational, irrational, or non-rational.

An increased understanding of how method deviations impact achieved contributions can be reached by posing questions such as: How do software developers' rational, irrational, and non-rational method deviations impact achieved contributions of SDM adoption and use?

5.3 Limitations

As with all studies, our study design has limitations. First, our findings rely on our search strategy and our selection of papers. We have been transparent with our selection of papers based on searches in the Scopus database. Of course, other search strategies could have been possible, such as the ones used by Goldkuhl and Karlsson [34] or Dybå and Dingsøyr [21]. Thus, we do not claim to have identified all empirical studies on SDM adoption and use.

Second, using our analytical framework in Tables 2 and the thematic synthesis involved subjective judgment. Classifying papers into different research themes and types of analyzed reasoning was not always an instrumental task. Due to the iterative nature of our coding process, we achieved reliability in our results by comparing and combining the author's individual analyses. Furthermore, we have tried to make the analysis as transparent as possible by providing the complete classification of papers in Appendix B, making it possible to scrutinize the analysis in detail.

Third, using an inductive categorization of existing research means a limitation when it comes to identifying avenues for future research. We have used the rear mirror to explore how much attention the identified research themes have received. Naturally, our strategy provides limited possibilities to identify blind spots in existing research, i.e., problems in practice that have not been addressed. Thus, we do not claim that the less researched areas identified in Table 4 and the research directions outlined above are the only potential avenues for future research on software developers' type of reasoning behind the adoption and use of SDMs.

6. Conclusion

This paper aimed to survey existing Software Development Method (SDM) research to scrutinize the current knowledge base on software developers' type of reasoning behind SDM adoption and use. To this end, we have used an analytical framework anchored in Weber's [24] work on social actions, classifying studies as analyzing rational, irrational, and/or non-rational reasoning. We identified 111 empirical studies on SDM adoption and SDM use, of which 28 addressed software developers' reasoning behind these activities; 14 addressed SDM adoption, and 14 addressed SDM use. Consequently, research on software developers' reasoning constitutes a significant part of the knowledge base. We also conclude that research on software developers' reasoning has predominantly considered SDM adoption and use as rational activities.

Our inductive analysis of empirical studies that addressed software developers' reasoning behind SDM adoption and SDM use resulted in five research themes: 1) Choice of SDM, 2) Competing rationalities, 3) Method deviation, 4) Understanding operationalization, and 5) Achieved contribution. More importantly, we used these themes to frame four future research directions together with a set of broad research questions summarized in Table 5. Although our results do not highlight an exhaustive listing of future research directions, they represent a series of notable examples of gaps that currently exist in the SDM field.

Table 5. Future research directions and research questions

No	Research direction	Research question
1	Consider how the rational, irrational, or non-rational choice of SDM impacts competing rationalities in SDM adoption and use	To what extent do rational, irrational, or non-rational SDM choices lead to competing rationalities in organizations?
2	Consider how competing rationalities in SDM adoption and use drive rational, irrational, or non-rational method deviation	How do software developers deal with competing rationalities when doing rational, irrational, and non-rational tailoring of SDMs?
3	Consider how software developers understand SDM impacts method deviations	How does software developers' understanding of SDM contribute to rational, irrational, and non-rational method deviations during adoption and use?
4	Consider how consequences of rational, irrational, and non-rational method deviations impact achieved contributions	How do software developers' rational, irrational, and non-rational method deviations impact achieved contributions of SDM adoption and use?

References

[1] The Standish Group, "CHAOS Report 2012," 2012.

[2] The Standish Group, "CHAOS Report 2015," 2015.

[3] The Standish Group, "CHAOS 2020: Beyond Infinity," 2020.

[4] K. Conboy, "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research,* vol. 20, no. 3, pp. 329-354, 2009.

[5] F. Karlsson and P. J. Ågerfalk, "Towards structured flexibility in information systems development: devising a method for method configuration," *Journal of Database Management,* vol. 20, no. 3, pp. 51-75, 2009.

[6] B. Fitzgerald, "The use of systems development methodologies in practice: a field study," *Information Systems Journal,* vol. 7, no. 3, pp. 201-212, 1997.

[7] F. Karlsson, "Longitudinal use of method rationale in method configuration: an exploratory study," *European Journal of Information Systems,* vol. 22, no. 6, pp. 690-710, 2013.

[8] B. Henderson-Sellers, J. Ralyté, P. J. Ågerfalk, and M. Rossi, *Situational Method Engineering*. Berlin Heidelberg: Springer-Verlag, 2014.

[9] F. Karlsson and P. J. Ågerfalk, "Exploring Agile Values in Method Configuration," *European Journal of Information Systems,* vol. 18, no. 4, pp. 300-316, 2009.

[10] M. Häggmark and P. Ågerfalk, "Why software engineers do not keep to the principle of separating business logic from display: A method rationale analysis," in *18th International Conference on Advanced Information Systems Engineering, CAiSE 2006*, E. Dubois and K. Pohl Eds. Berlin: Springer, 2006, pp. 399-413.

[11] V. T. Heikkilä, M. Paasivaara, and C. Lassenius, "ScrumBut, But Does it Matter? A Mixed-Method Study of the Planning Process of a Multi-team Scrum Organization," presented at the IEEE International Symposium on Empirical Software Engineering and Measurement 2013, Baltimore, MD, USA, 10-11 October, 2013, 2013.

[12] V. Stray, D. I. K. Sjoberg, and T. Dybå, "The daily stand-up meeting: A grounded theory study," *The Journal of Systems and Software,* vol. 114, no. April 2016, pp. 101-124, 2016.

[13] K. Beck *et al.* "The agile manifesto." www.agilemanifesto.org (accessed 3 March, 2012).

[14] N. Jayaratna, P. Holt, and T. Wood-Harper, "Criteria for Methodology Choice in Information System Development," *The Journal of Contemporary Issues in Business and Government,* vol. 5, no. 2, pp. 30-34, 1999.

[15] S. Brinkkemper, "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology,* vol. 38, pp. 275-280, 1996.

[16] N. L. Russo and E. Stolterman, "Exploring the assumptions underlying information systems methodologies: their impact on past, present and future ISM research," *Information Technology & People,* vol. 13, no. 4, pp. 313–327, 2000.

[17] C. Rolland and N. Prakash, "A Proposal For Context-Specific Method Engineering," in *Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering,* S. Brinkkemper, K. Lyytinen, and R. Welke Eds. Atlanta, United States: Chapman & Hall, 1996, sec. Atlanta, USA, pp. 191–208.

[18] C. Gonzalez-Perez, P. Giorgini, and B. Henderson-Sellers, "Method Construction by Goal Analysis," in *16th International Conference on Information Systems Development (ISD2007),* C. Barry, K. Conboy, M. Lang, W. Wojtkowski, and G. Wojtkowski Eds. Galway, Ireland: Springer-Verlag, 2007, pp. 79-92.

[19] T. E. Havstorm, F. Karlsson, and K. Hedström, "Uncovering Situations of Cargo Cult Behavior in Agile Software Development Method Use," in *The 56th The Hawaii International Conference on System Sciences (HICSS 2023)*, 2023, pp. 6486-6495.

[20] J. Iivari, H. Rudy, and H. K. Klein, "A dynamic framework for classifying information systems development methodologies and approaches," *Journal of Management Information Systems,* vol. 17, no. 3, pp. 179-218, 2001.

[21] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology,* vol. 50, no. 9-10, pp. 833-859, 2008.

[22] U. Telemaco, T. Oliveira, P. Alencar, and D. Cowan, "A Catalogue of Agile Smells for Agility Assessment," *IEEE Access,* vol. 8, pp. 79239-79259, 2020, doi: doi: 10.1109/ACCESS.2020.2989106.

[23] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Information and Software Technology,* vol. 51, no. 1, pp. 7-15, 2009.

[24] M. Weber, *Economy and society*. Berkeley, CA: University of California Press, 1978.

[25] D. S. Cruzes and T. Dybå, "Recommended Steps for Thematic Synthesis in Software Engineering," presented at the 2011 International Symposium on Empirical Software Engineering and Measurement, Banff, AB, Canada, 22-23 September, 2011, 2011.

[26] S. A. Walters, J. E. Broady, and R. J. Hartley, "A Review of Information Systems Development Methodologies," *Library Management,* vol. 15, no. 6, pp. 5-19, 1994.

[27] M. F. Abrar *et al.*, "De-motivators for the adoption of agile methodologies for large-scale software development teams: An SLR from management perspective," *Journal of Software: Evolution and Process,* vol. 32, no. 12, pp. 1-20, 2020.

[28] G. Onwujekwe and H. Weistroffer, "Agile Development in Bureaucratic Environments: A Literature Review," in *15th European, Mediterranean, and Middle Eastern Conference, EMCIS 2018, Limassol, Cyprus, October 4-5, 2018, Proceedings*, M. Themistocleous and P. Rupino da Cunha Eds. Cham: Springer, 2019, pp. 316-330.

[29] M. Alqudah and R. Razali, "A Review of Scaling Agile Methods in Large Software Development," *International Journal on Advanced Science, Engineering and Information Technology,* vol. 6, no. 6, pp. 828-837, 2016.

[30] A. S. Campanelli and F. S. Parreiras, "Agile methods tailoring–A systematic literature review," *The Journal of Systems and Software,* vol. 110, no. December 2015, pp. 85-100, 2015.

[31] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software,* vol. 85, no. 6, pp. 1213-1221, 2012.

[32] M. Gall and F. Pigni, "Taking DevOps mainstream: a critical review and conceptual framework," *European Journal of Information Systems,* pp. 548-567, 2021, doi: DOI: 10.1080/0960085X.2021.1997100.

[33] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, and A. B. M. Sultan, "A Systematic Literature Review on Relationship between Agile Methods and Open Source Software Development Methodology," *International Review on Computer and Software,* vol. 7, no. 4, pp. 1602-1607, 2012.

[34] G. Goldkuhl and F. Karlsson, "Method Engineering as Design Science," *Journal of the Association for Information Systems,* vol. 21, no. 5, pp. 1237-1278, 2020.

[35] S. Gutiérrez-Ríos, I. Morales, and M. Vargas-Lombardo, "A Study of New Decade of Agile and Lean Methodologies for Development Software," presented at the 4th International Conference on Information and Computer Technologies (ICICT 2021), Kahului, Maui Island, HI, USA, 11-14 March, 2021, 2021.

[36] N. R. Hassan and L. Mathiassen, "Distilling a body of knowledge for information systems development," *Information Systems Journal,* vol. 28, no. 1, pp. 175-226, 2018.

[37] R. Hirschheim, H. K. Klein, and K. Lyytinen, "Exploring the intellectual structures of information systems development: a social action theoretic analysis," *Accounting, Management and Information Technologies,* vol. 6, no. 1–2, pp. 1–64, 1996/1/6 1996. [Online]. Available: http://www.sciencedirect.com/science/article/B6VFY-3VV41VP-1/1/bc8fd06e5b8808c6a697ab94ef5e3584.

[38] S. Matalonga, M. Solari, and G. Matturro, "Factors affecting distributed agile projects: A systematic review," *International Journal of Software Engineering and Knowledge Engineering,* vol. 23, no. 9, pp. 1289-1301, 2013.

[39] D. Mihailescu and M. Mihailescu, "Exploring the Nature of Information Systems Development Methodology: A Synthesized View Based on a Literature Review," presented at the 15th Americas Conference on Information Systems (AMICS 2009), San Francisco, California, 2009.

[40] M. Moloto, A. Harmse, and T. Zuva, "Impact of Agile Methodology Use on Project Success in Organizations - A Systematic Literature Review," in *Software Engineering Perspectives in Intelligent Systems. CoMeSySo 2020. Advances in Intelligent Systems and Computing*, R. Silhavy, P. Silhavy, and Z. Prokopova Eds. Cham: Springer, 2020, pp. 267-280.

[41] V. Sambamurthy and L. J. Kirsch, "An Integrative Framework of the Information Systems Development Process," *Decision sciences,* vol. 31, no. 2, pp. 391-411, 2000, doi: 10.1111/j.1540-5915.2000.tb01628.x.

[42] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering (Vol. 5)," in "EBSE Technical Report," School of Computer Science and Mathematics, Keele University, Keele, UK, 2007.

[43] U. Franke and J. Brynielsson, "Cyber situational awareness–a systematic review of the literature," *Computers & Security*, vol. 46, pp. 18-31, 2014.

[44] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *The Journal of Systems and Software*, vol. 80, no. 4, pp. 571-583, 2007.

[45] D. Vavpotič and T. Hovelja, "Improving the evaluation of software development methodology adoption and its impact on enterprise performance," *Computer Science and Information Systems*, vol. 9, no. 1, pp. 165-187, 2012.

[46] B. Fitzgerald, N. L. Russo, and E. Stolterman, *Information Systems Development - Methods in Action*. Berkshire, UK: McGraw-Hill, 2002.

[47] S. Kalberg, "Max Weber's Types of Rationality: Cornerstones for the Analysis of Rationalization Processes in History," *American Journal of Sociology*, vol. 85, no. 5, pp. 1145–1179, 1980.

[48] T. E. Mäki-Runsas, K. Wistrand, and F. Karlsson, "Cargo Cults in Information Systems Development: A Definition and an Analytical Framework," in *Advances in Information Systems Development. Lecture Notes in Information Systems and Organisation*, vol. vol. 34, B. Andersson, B. Johansson, C. Barry, M. Lang, H. Linger, and C. Schneider Eds. Cham: Springer, 2019, pp. 35-53.

[49] B. Bygstad, G. Ghinea, and E. Brevik, "Systems development methods and usability in Norway: An industrial perspective," in *Usability and Internationalization - HIC and Culture. Second International Conference on Usability and Internationalization, UI-HCII 2007. Held as Part of HCI International 2007, Beijing China, July 2007, Proceedings, Part I*, N. Aykin Ed. Berlin: Springer, 2007, pp. 258-266.

[50] T. Jamil and H. Al-Hajry, "Information System Development Methodologies Adoption in the Sultanate of Oman – An Exploratory Study," presented at the 5th International Conference on Information Technology and Applications (ICITA 2008), Cairns, Australia, 2008.

[51] J. S. Babb, R. Hoda, and J. Nørbjerg, "Barriers to learning in agile software development projects," in *Agile Processes in Software Engineering and Extreme Programming - 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*, H. Baumeister and B. Weber Eds. Berlin, Heidelberg: Springer, 2013, pp. 1-15.

[52] H. Hajjdiab, A. S. Taleb, and J. Ali, "An Industrial Case Study for Scrum Adoption," *An Industrial Case Study for Scrum Adoption*, vol. 7, no. 1, pp. 237-242, 2012.

[53] R. Poston, J. Patel, and Y. Tymchenko, "Making Sense of Resistance to Agile Adoption in Waterfall Organizations: Social Intelligence and Leadership," presented at the 22nd Americas Conference on Information Systems: Surfing the IT Innovation Wave, AMCIS 2016, San Diego; United States, 11-14 August, 2016, 2016.

[54] M. Laanti, O. Salo, and P. Abrahamsson, "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation," *Information and Software Technology*, vol. 53, no. 3, pp. 276-290, 2011.

[55] T. J. Gandomani, H. Zulzalil, A. A. Abdul Ghani, and A. B. M. Sultan, "Important considerations for agile software development methods governance," *Journal of Theoretical and Applied Information Technology*, vol. 55, no. 3, pp. 345-351, 2013.

[56] J. McAvoy, I. Owens, and D. Sammon, "Towards the development of a simple tool to assist in agile methodology adoption decisions: agile adoption matri," in *14th European Conference on Information Systems (ECIS 2006)*, Gothenburg, Sweden, 12-14 June, 2006 2006: AIS Electronic Library (AISeL), p. Paper 16.

[57] M. Toleman, F. Darroch, and M. Ally, "Web Publishing: An Extreme, Agile Experience," in *Business Agility and Information Technology Diffusion. TDIT 2005. IFIP International Federation for Information Processing*, R. L. Baskerville, L. Mathiassen, J. Pries-Heje, and J. I. DeGross Eds. Boston, MA: Springer, 2005, pp. 245-256.

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

[59] A. Fruhling, K. Tyser, and G.-J. De Vreede, "Experiences with Extreme Programming in telehealth: Developing and implementing a biosecurity health care application," presented at the The 38th Hawaii International Conference on System Sciences (HICSS, 2005), Big Island, HI, USA, 2005.

[60] N. Joseph, C. Marnewick, and M. J. Santana, "Agile software development and it project performance in South Africa: A positive relationship?," in *IAMOT 2016 - 25th International Association for Management of Technology Conference, Proceedings: Technology - Future Thinking 2016*, Orlando, FL, USA, L. Pretorius, G. A. Thopil, and Y. Hosni, Eds., 2016: International Association for Management of Technology Conference, pp. 338-358.

[61] C. J. Hardy, J. B. Thompson, and H. M. Edwards, "The use, limitations and customization of structured systems development methods in the United Kingdom," *Information and Software Technology,* vol. 37, no. 9, pp. 467-477, 1995.

[62] J. Iivari and J. Maansaari, "The Usage of Systems Development Methods: are we stuck to old practice?," *Information and Software Technology,* vol. 40, pp. 501–510, 1998.

[63] P. Abrahamsson, "Extreme programming: First results from a controlled case study," in *29th EUROMICRO Conference*, G. Chroust and C. Hofer Eds. Los Alamitos, California, USA: IEEE Computer Society, 2003, pp. 259-266.

[64] R. Procter, M. Rouncefield, M. Poschen, Y. Lin, and A. Voss, "Agile Project Management: A Case Study of a Virtual Research Environment Development Project," *Computer Supported Cooperative Work,* vol. 20, no. 3, pp. 197-225, 2011.

[65] 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, pp. 58-64, 2008.

[66] D. E. Avison, M. H. Shah, R. S. Powell, and P. S. Uppal, "Applying methodologies for information systems development," *Journal of Information Technology,* vol. 7, no. 3, pp. 127-140, 1992.

[67] B. Fitzgerald, "The Use of Systems Development Methodologies in Practice: A Field Study," *The Information Systems Journal,* vol. 7, no. 3, pp. 201-212, 1997.

[68] B. Fitzgerald, "An empirical investigation into the adoption of systems development methodologies," *Information and Management,* vol. 34, no. 6, pp. 317-328, 1998.

[69] J. Iivari and M. Huisman, "The relationship between organisational culture and the deployment of systems development methodologies," in *Advanced Information Systems Engineering - 13th International Conference, CAiSE 2001 Interlaken, Switzerland, June 4–8, 2001 Proceedings,* K. R. Dittrich, A. Geppert, and M. C. Norrie Eds. Berlin: Springer Verlag, 2001, pp. 234-250.

[70] L. Vijayasarathy and D. Turk, "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," *Information and Software Technology,* vol. 54, no. 2, pp. 137-148, 2012.

[71] P. K. Vejandla and L. B. Sherrell, "Why an AI research team adopted XP practices," presented at the ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference, Clemson, SC, March 19-21, 2009, 2009.

[72] D. Mishra and A. Mishra, "Complex software project development: agile methods adoption," *Journal of Software Maintenance and Evolution: Research and Practice,* vol. 23, no. 8, pp. 549-564, 2011.

[73] P. S. Taylor *et al.*, "Applying an Agility/Discipline Assessment for a Small Software Organisation," in *Product-Focused Software Process Improvement 7th International Conference, PROFES 2006 Amsterdam, the Netherlands, June 2006 Proceedings,* J. Münch and M. Vierimaa Eds. Berlin: Springer, 2006, pp. 290-304.

[74] J. F. Tripp and D. J. Amstrong, "Exploring the Relationship between Organizational Adoption Motives and the Tailoring of Agile Methods," presented at the 47th Hawaii International Conference on System Science (HICSS 2014), Waikoloa, HI, USA 2014.

[75] R. Alt, G. Auth, and C. Kögler, "Transformation of Consulting for Software-Defined Businesses: Lessons from a DevOps Case Study in a German IT Company," in *Advances in Consulting Research - Recent Findings and Practical Cases*, V. Nissen Ed. Cham, Switzerland: Springer International Publishing, 2019, pp. 385-404.

[76] C. Sauer and C. Lau, "Trying to adopt systems development methodologies — a case-based exploration of business users' interests," *Information Systems Journal,* vol. 7, no. 255-275, 4, 1997.

[77] J. McAvoy and T. Butler, "A Paradox of the Change to User Stories: The Application of the Theory of Competing Commitments," in *13th European Conference on Information Systems (ECIS 2005)*, Regensburg, Germany, May 26-28, 2005 2005: AIS eLibrary, p. Paper 81.

[78] J. McAvoy and T. Butler, "The impact of the Abilene Paradox on double-loop learning in an agile team," *Information and Software Technology,* vol. 49, no. 6, pp. 552-563, 2007.

[79] D. E. Strode, S. L. Huff, and A. Tretiakov, "The impact of organizational culture on agile method use," in *42nd Annual Hawaii International Conference on System Sciences (HICSS 2009)*, Waikoloa, HI; United States, 2009: IEEE.

[80] D. Dennehy and K. Conboy, "Breaking the flow: a study of contradictions in information systems development (ISD)," *Information Technology & People,* vol. 33, no. 2, pp. 477-501, 2020.

[81] K. Mohan, F. Ahlemann, and M. Kugler, "Understanding the Nature of Use Regarding System Development and Management Methodologies - Using Psychoanalysis to Understand the Influences of Methodology Attributes," in *46th Hawaii International Conference on System Sciences (HICSS 2013)*, Wailea, Maui, HI USA, 2013: IEEE, pp. 4384-4393.

[82]P. Patanakul and R. Rufo-McCarron, "Transitioning to agile software development: Lessons learned from a government-contracted program," *Journal of High Technology Management Research,* vol. 29, no. 2, pp. 181-192, 11/01/November 2018 2018.

[83] H. Berger and P. Beynon-Davies, "The utility of rapid application development in large-scale, complex projects," *Information Systems Journal,* vol. 19, no. 6, pp. 549-570, 2009.

[84] P. Middleton, "Software quality by administration," *Software Quality Journal,* vol. 7, no. 3-4, pp. 261-275, 1998.

[85] P. Middleton, "Barriers to the efficient and effective use of information technology," *International Journal of Public Sector Management,* vol. 13, no. 1, pp. 85-99, 2000.

[86] P. Middleton and B. McCollum, "Management of process improvement by prescription," *Journal of Systems and Software,* vol. 57, no. 1, pp. 9-19, 2001.

[87] P.-S. Chen, G. Y.-H. Chen, S.-F. Lien, and W.-T. Huang, "Using Scrum and unified modelling language to analyze and design an automatic course scheduling system," *Information Engineering,* vol. 42, no. 6, pp. 534-543, 2019.

[88] T. Saarinen, "System development methodology and project success," *Information & Management,* vol. 19, no. 3, pp. 183-193, 1990.

[89] I. Andersson and K. Nilsson, "Diagnosing Diffusion Practices Within a Software Organization," in *Diffusing Software Product and Process Innvovations - IFIP TC8 WG8.6 Fourth Working Conference on Diffusing Software Product and Process Innovations April 7–10, 2001, Banff, Canada*, M. A. Ardis and B. L. Marcolin Eds. Boston, MA: Springer, 2001, pp. 111-130.

[90] O. Hazzan and Y. Dubinsky, "The Concept of Change in Technology Transfer," in *TT '06: Proceedings of the 2006 international workshop on Software technology transfer in software engineering*. New York, NY, United States: Association for Computing Machinery, 2006, pp. 29-34.

[91] T. E. Mäki-Runsas, "Towards Identifying Information Systems Development Method (ISDM) Cargo Cult Behavior," presented at the 25th Americas Conference on Information Systems (AMCIS 2019), Cancun, Mexico, August 15-17, 2019, 2019.

[92] E. Mnkandla and B. Dwolatzky, "Balancing the human and the engineering factors in software development," presented at the 7th Africon Conference in Africa, Gaborone, Botswana, 15-17 September, 2004, 2004.

[93] T. L. Roberts Jr, W. Leigh, and R. L. Purvis, "Perceptions on Stakeholder Involvement in the Implementation of System Development Methodologies," *Journal of Computer Information Systems,* vol. 40, no. 3, pp. 78-83, 2000.

[94] A. A. Mohallel and J. M. Bass, "Agile Software Development Practices in Egypt SMEs: A Grounded Theory Investigation," in *Information and Communication Technologies for Development. Strengthening Southern-Driven Cooperation as a Catalyst for ICT4D*, P. Nielsen and H. C. Kimaro Eds. Cham, Switzerland: Springer International Publishing, 2019, pp. 355-365.

[95] S. Soundararajan, J. D. Arthur, and O. Balci, "A Methodology for Assessing Agile Software Development Methods," in *2012 Agile Conference*, Dallas, TX, USA, 2012: IEEE, pp. 51-54.

[96] K. D. Clark, T. T. Woodson, R. J. Holden, R. Gunn, and D. J. Cohen, "Translating Research into Agile Development (TRIAD): Development of Electronic Health Record Tools for Primary Care Settings," *Methods of Information in Medicine,* vol. 58, no. 1, pp. 1-8, 2019.

[97] D. Truex, R. Baskerville, and J. Travis, "Amethodical systems development: the deferred meaning of systems development methods," *Accounting, Management and Information Technologies,* vol. 10, no. 1, pp. 53-79, 2000.

[98] V. Sambamurthy and L. J. Krisch, "An integrative framework of the information systems development process," *Decision Sciences,* vol. 31, no. 2, pp. 391-411, 2000.

[99] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods. Review and analysis," VTT Technical Research Centre of Finland, Espoo, 2002.

[100] A. Abdul, J. M. Bass, H. Ghavimi, and P. Adam, "Product innovation with scrum: A longitudinal case study," presented at the 2017 International Conference on Information Society (i-Society), Dublin, Ireland, 17-19 July, 2017, 2017.

[101] S. Ahmadzai and M. Bakhsh, "An Empirical Investigation on Lean Method Usage: Issues and Challenges in Afghanistan," in *AI and IoT for Sustainable Development in Emerging Countries*, Z. Boulouard, M. Ouaissa, M. Ouaissa, and S. El Himer Eds. Cham: Springer, 2022, pp. 255-264.

[102] F. S. Altuwaijri and M. A. Ferrario, "Awareness and Perception of Agile in Saudi Software Industry," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, Madrin, Spain, 25-28 May, 2021 2021: IEEE, pp. 10-18.

[103] H. Ayed, B. Vanderose, and N. Habra, "Agile cultural challenges in Europe and Asia: insights from practitioners," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, Buenos Aires, Argentina, 20-28 May, 2017 2017: IEEE, pp. 153-162.

[104] P. L. Bannerman, E. Hossain, and R. Jeffery, "Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?," presented at the 45th Hawaii International Conference on System Sciences (HICSS 2012), Maui, HI, USA, 4-7 January, 2012, 2012.

[105] R. Barrow, J. Carroll, and R. Smith, "Towards an integrated model of ISDM tailoring," in *20th Australasian Conference on Information Systems, ACIS 2009*, 2009, Paper 92.

[106] R. Barrow, J. Carroll, R. Smith, and K. Frampton, "The Application of an Information Systems Development Methodology in Practice – An Exploratory Study," in *PACIS 2007 Proceedings*, 2007, Paper 95.

[107] D. A. Chevers and C. C. Whyte, "The adoption and benefits of Agile software development methods in Jamaica," in *21st Americas Conference on Information Systems (AMCIS 2015)*, Fajardo, Puerto Rico, 2015: AIS eLibrary, pp. 1-9.

[108] G. Coleman and R. Verbruggen, "A quality software process for rapid application development," *Software Quality Journal,* vol. 7, no. 2, pp. 107-122, 1998.

[109] O. A. Dada and I. T. Sanusi, "The adoption of Software Engineering practices in a Scrum environment," *African Journal of Science, Technology, Innovation and Development,* 2021, doi: https://doi.org/10.1080/20421338.2021.1955431.

[110] T. L. De Sousa, E. Venson, R. M. D. C. Figueiredo, R. A. Kosloski, and L. C. M. Ribeiro, "Using Scrum in Outsourced Government Projects: An Action Research," presented at the 49th Hawaii International Conference on System Sciences (HICSS 2016), Koloa, HI, USA, 5-8 January, 2016, 2016.

[111] A. Fruhling and J. Zhang, "An Empirical Study Examining the Usage and Perceived Importance of XP Practices," in *13th Americas Conference on Information Systems (AMCIS 2007)*, Keystone, Colorado, USA, 9-12 August 2007: AIS eLibrary, p. Paper 138.

[112] T. J. Gandomani, H. Zulzalil, A. A. Abdul Ghani, A. B. M. Sultan, and K. Y. Sharif, "How human aspects impress Agile software development transition and adoption," *International Journal of Software Engineering and its Applications,* vol. 8, no. 1, pp. 129-148, 2014.

[113] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and K. Y. Sharif, "An Exploratory Study on Managing Agile Transition and Adoption," in *Recent Advances in Information and Communication Technology - Proceedings of the 10th International Conference on Computing and Information Technology (IC2IT2014)*, S. Boonkrong, H. Unger, and P. Meesad Eds. Cham: Springer, 2014, pp. 177-188.

[114] M. Ganis, E. M. Maximilien, and T. Rivera, "A brief report on working smarter with Agile software development," *IBM Journal of Research and Development* vol. 54, no. 4, pp. 1-10, 2010.

[115] M. Gannon, "An agile implementation of SCRUM," presented at the 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2-9 March, 2013, 2013.

[116] I. Ghani and M. Bello, "Agile Adoption in IT Organizations," *KSII Transactions on Internet and Information Systems,* vol. 9, no. 8, pp. 3231-3248, 2015.

[117] H. Hajjdiab and A. S. Taleb, "Agile adoption experience: A case study in the U.A.E," presented at the 2nd International Conference on Software Engineering and Service Science (ICSESS 2011), Beijing, China, 15-17 July, 2011, 2011.

[118] J. A. V. M. K. Jayakody and W. M. J. I. Wijayanayake, "Challenges for adopting DevOps in information technology projects," in *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, 16 September, 2021 2021: IEEE, pp. 203-210.

[119] L. H. Jones and C. T. Kydd, "An information processing framework for understanding success and failure of MIS development methodologies," *Information & Management,* vol. 15, no. 5, pp. 263-271, 1988.

[120] F. Karlsson, "A wiki-based approach to method tailoring," presented at the The 3rd International Conference on the Pragmatic Web (ICPW2008), Uppsala, Sweden, 28-30 September, 2008.

[121] M. Käpyaho and M. Kauppinen, "Agile requirements engineering with prototyping: A case study," presented at the 23rd International Requirements Engineering Conference (RE 2015), Ottawa, ON, Canada, 2015.

[122] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl, "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson," presented at the 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, MD, USA, 2013.

[123] S. A. Licorish *et al.*, "Adoption and Suitability of Software Development Methods and Practices," presented at the 23rd Asia-Pacific Software Engineering Conference (APSEC 2016), Hamilton, New Zealand, 2017.

[124] J. A. Livermore, "What Elements of XP are being Adopted by Industry Practitioners?," presented at the The IEEE SoutheastCon 2006, Memphis, TN, USA, 31 March-2 April, 2006, 2006.

[125] J. A. Livermore, "Factors that Significantly Impact the Implementation of an Agile Software Development Methodology," *Journal of Software,* vol. 3, no. 4, pp. 31-36, 2008.

[126] K. M. Lui and K. C. C. Chan, "Pair programming productivity: Novice–novice vs. expert–expert," *International Journal of Human-Computer Studies,* vol. 64, no. 9, pp. 915-925, 2006.

[127] L. Mahadevan, W. J. Kettinger, and T. O. Meservy, "Running on Hybrid: Control Changes when Introducing an Agile Methodology in a Traditional "Waterfall" System Development Environment," *Communications of the Association for Information Systems,* vol. 36, p. Article 5, 2015.

[128] K. Mannaro, M. Melis, and M. Marchesi, "Empirical Analysis on the Satisfaction of IT Employees Comparing XP Practices with Other Software Development Methodologies," in *Extreme Programming and Agile Processes in Software Engineering: 5th International Conference, XP 2004,* vol. LNCS 3092, J. Eckstein and H. Baumeister Eds. Berlin: Springer, 2004, pp. 166-174.

[129] A. Marks, "Information Systems Development Methodolgies in Developing Higher Education," *Creative Education,* vol. 3, no. 1, pp. 114-119, 2012.

[130] M. N. Masrek, N. Hussin, and N. Tarmuchi, "An exploratory study on systems development methodologies for web‐based applications," *Information Management & Computer Security,* vol. 16, no. 2, pp. 137-149, 2008.

[131] F. R. Maulana and T. Raharjo, "Identification of Challenges, Critical Success Factors, and Best Practices of Scrum Implementation: An Indonesia Telecommunication Company Case Study," *Journal of Physics: Conference Series,* vol. 1811, no. 1, p. Article 012120, 2021.

[132] P. Mohagheghi and C. Lassenius, "Organizational implications of agile adoption: a case study from the public sector," in *ESEC/FSE 2021: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering,* Athens, Greece, D. Spinellis, G. Gousios, M. Chechik, and M. Di Penta, Eds., 2021: Association for Computing Machinery, pp. 1444-1454.

[133] A. C. Nelson and J. T. C. Teng, "Do systems development methodologies and CASE tools decrease stress among systems analysts?," *Behav. Inf. Technol.,* vol. 19, no. 4, pp. 307-313, 2000.

[134] R. E. Quelal, M. Villavicencio, and L. E. Mendoza, "A survey of agile software development methodologies in Ecuador," in *13th Iberian Conference on Information Systems and Technologies (CISTI 2018),* Caceres, Spain, 13-16 June, 2018 2018: IEEE, pp. 1-6.

[135] M. M. Rahim, A. Seyal, and M. N. A. Rahman, "Use of software systems development methods An empirical study in Brunei Darussalam," *Information and Software Technology,* vol. 39, no. 14-15, pp. 949-963, 1998.

[136] Y. Rezgui, A. Marks, H. AL-Hajri, and I. Wilson, "ISDM adoption using Delphi and AHP,," presented at the 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2012), Wuhan, China, 23-25 May, 2012, 2012.

[137] Y. Rezgui, A. Marks, H. AL-Hajri, and I. Wilson, "Developing an ISDM Adoption Decision Model Using Delphi and AHP," *Arabian Journal for Science and Engineering,* vol. 39, no. 4, pp. 2799 - 2815, 2014.

[138] A. Ribeiro and L. Domingues, "Acceptance of an agile methodology in the public sector," *Procedia Computer Science,* vol. 138, pp. 621-629, 2018.

[139] K. Rindell, J. Ruohonen, J. Holvitie, S. Hyrynsalmi, and V. Leppänen, "Security in agile software development: A practitioner survey," *Information and Software Technology,* vol. 131, no. March 2021, p. Article 106488, 2021.

[140] T. L. Roberts Jr, M. L. Gibson, K. T. Fields, and R. K. Rainer Jr, "Factors that impact implementing a system development methodology," *IEEE Transactions on Software Engineering,* vol. 24, no. 8, pp. 640-649, 1998.

[141] T. L. Roberts, M. L. Gibson, and K. T. Fields, "System Development Methodology Implementation: Perceived Aspects of Importance," *Information Resources Management Journal,* vol. 12, no. 3, pp. 27-38, 1999.

[142] T. L. Roberts, W. Leigh, R. L. Purvis, and M. J. Parzinger, "Utilizing knowledge links in the implementation of system development methodologies," *Information and Software Technology,* vol. 43, no. 11, pp. 635-640, 2001.

[143] B. Rowlands, "Power and Authority Over Systems Developers by the Business Client," presented at the 28th International Conference on Information Systems (ICIS 2007), Montreal, Quebec, Canada, December 9-12, 2007, 2007.

[144] C. Schindler, "Agile Software Development Methods and Practices in Austrian IT-Industry: Results of an Empirical Study," presented at the International Conference on Computational Intelligence for Modelling Control & Automation, 2008, Vienna, Austria, 2008.

[145] M. Serour and D. Winder, "Radar: A model for IT organizational change. Four Australian case studies on the adoption and diffusion of method engineering," presented at the European and Mediterranean Conference on Information Systems 2007 (EMCIS2007) June 24-26, 2007, 2007.

[146] C. Snook and R. Harrison, "Practitioners' views on the use of formal methods: an industrial survey by structured interview," *Information and Software Technology,* vol. 43, no. 4, pp. 275-283, 2001.

[147] Z. Šochová, "Software development methodology for fast changing environment," *Frontiers in Artificial Intelligence and Applications,* vol. 199, no. 1, pp. 379 - 388, 2009.

[148] T. r. Tekbulut, A. İnal, and B. l. Doğanay, "LAPIS – LOGO Agile Process Improvement System," in *Software Process Improvement and Capability Determination - 15th International Conference, SPICE 2015, Gothenburg, Sweden, June 16-17, 2015. Proceedings*, T. Rout, R. V. O'Connor, and A. Dorling Eds. Cham, Switzerland: Springer International Publishing, 2015, pp. 131–144.

[149] D. Tudor and G. A. Walter, "Using an agile approach in a large, traditional organization," presented at the AGILE 2006 (AGILE'06), Minneapolis, MN, USA, 23-28 July, 2006, 2006.

[150] D. Vavpotic and M. Bajec, "An approach for concurrent evaluation of technical and social aspects of software development methodologies," *Information and Software Technology,* vol. 51, no. 2, pp. 528-545, 2009.

[151] D. Vavpotič, M. Bajec, and M. Krisper, "Scenarios for Improvement of Software Development Methodologies," in *Advances in Information Systems Development - Bridging the Gap between Academia and Industry*, A. G. Nilsson *et al.* Eds. Boston, MA: Springer, 2006, pp. 277-288.

[152] S. L. R. Vrhovec, "Agile development of a hospital information system," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 - Proceedings*, Opatija, Croatia, 2016: IEEE, pp. 416-421.

[153] S. K. Weiss and P. Brune, "Crossing the boundaries – Agile methods in large-scale, plan-driven organizations: A case study from the financial services industry," in *Advanced Information Systems Engineering, 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings*, vol. LNCS 10253. Cham: Springer, 2017, pp. 380-393.

[154] J. Yli-Huumo, O. Taipale, and K. Smolander, "Software Development Methods and Quality Assurance: Special Focus on South Korea," in *Systems, Software and Services Process Improvement - 21st European Conference, EuroSPI 2014, Luxembourg, June 25-27, 2014. Proceedings*, B. Barafort, R. V. O'Connor, A. Poth, and R. Messnarz Eds. Berlin, Heidelberg: Springer, 2014, pp. 159-169.

[155] A. Yoshii and K. Higa, "Analysis of the peculiarity of the japanese software development style in offshore software development," *IEEJ Transactions on Electrical and Electronic Engineering,* vol. 6, no. 1, pp. 46-50, 2011.

Appendix A. Search queries and results

Table A1 shows the search queries that were used for searching in Scopus and the results from our searches. For searching in the database, a combination of search query 1 and search query 2 was used. For example, as the first row in the table shows, the result of searching "Systems Development Method" and "Use" in Scopus generated 192 papers.

Table A1. Combinations of search criteria that generated search results

Search query 1	Search query 2	Number of papers
Systems Development Method	Use	192
Systems Development Method	Utilizing	7
Systems Development Method	Usage	11
Systems Development Method	Adoption	25
Systems Development Method	Implementation	79
Systems Development Method	Method Rationale	6
Systems Development Method	Rationality Resonance	0
Systems Development Method	Customize	0
Systems Development Method	Diffusion	7
Systems Engineering Method	Use	116
Systems Engineering Method	Utilizing	6
Systems Engineering Method	Usage	5
Systems Engineering Method	Adoption	7
Systems Engineering Method	Implementation	48
Systems Engineering Method	Method Rationale	1
Systems Engineering Method	Rationality Resonance	0
Systems Engineering Method	Customize	1
Systems Engineering Method	Diffusion	1
Systems Development Methodology	Use	195
Systems Development Methodology	Utilizing	3
Systems Development Methodology	Usage	28
Systems Development Methodology	Adoption	29
Systems Development Methodology	Implementation	92
Systems Development Methodology	Method Rationale	0
Systems Development Methodology	Rationality Resonance	0
Systems Development Methodology	Customize	0
Systems Development Methodology	Diffusion	8
Systems Engineering Methodology	Use	120
Systems Engineering Methodology	Utilizing	9
Systems Engineering Methodology	Usage	7
Systems Engineering Methodology	Adoption	9
Systems Engineering Methodology	Implementation	76
Systems Engineering Methodology	Method Rationale	0
Systems Engineering Methodology	Rationality Resonance	0
Systems Engineering Methodology	Customize	0
Systems Engineering Methodology	Diffusion	1
Software Development Method	Use	342
Software Development Method	Utilizing	2
Software Development Method	Usage	33
Software Development Method	Adoption	93
Software Development Method	Implementation	152
Software Development Method	Method Rationale	1
Software Development Method	Rationality Resonance	0
Software Development Method	Customize	4
Software Development Method	Diffusion	9
Software Engineering Method	Use	225
Software Engineering Method	Utilizing	2
Software Engineering Method	Usage	20
Software Engineering Method	Adoption	28
Software Engineering Method	Implementation	92
Software Engineering Method	Method Rationale	0
Software Engineering Method	Rationality Resonance	0
Software Engineering Method	Customize	2
Software Engineering Method	Diffusion	2
Software Development Methodology	Use	433

Search query 1	Search query 2	Number of papers
Software Development Methodology	Utilizing	16
Software Development Methodology	Usage	47
Software Development Methodology	Adoption	90
Software Development Methodology	Implementation	220
Software Development Methodology	Method Rationale	0
Software Development Methodology	Rationality Resonance	0
Software Development Methodology	Customize	3
Software Development Methodology	Diffusion	12
Software Engineering Methodology	Use	174
Software Engineering Methodology	Utilizing	5
Software Engineering Methodology	Usage	13
Software Engineering Methodology	Adoption	13
Software Engineering Methodology	Implementation	102
Software Engineering Methodology	Method Rationale	0
Software Engineering Methodology	Rationality Resonance	0
Software Engineering Methodology	Customize	0
Software Engineering Methodology	Diffusion	3
Total		*3,227*

Appendix B. Papers included in the detailed analysis

Table B1 presents a detailed analysis of the papers that we could access. The leftmost column contains the authors. The next two columns show the classification according to the two investigated phases; the result is presented in Figure 2. Columns four to six present the classification of the investigated type of reasoning. Finally, the rightmost column shows the result from our thematic analysis, i.e., the identified research themes. The identified research themes and papers are presented in Table 4.

Table B1. Detailed analysis

	Phase		Type of reasoning			
Authors	**Adoption**	**Use**	**Rational**	**Irrational**	**Non-rational**	**Research theme (included in Table 4)**
Abdul et al. [100]		X				
Ahmadzai and Bakhsh [101]	X					
Abrahamsson [63]		X				
Alt et al. [75]	X		X			Choice of SDM
Altuwaijri and Ferrario [102]		X				
Andersson and Nilsson [89]	X		X	X		Understanding operationalization
Avison et al. [66]		X				
Ayed et al. [103]	X					
Babb et al. [51]	X					
Bannerman et al. [104]	X	X				
Barrow et al. [105]	X					
Barrow et al. [106]	X					
Berger and Beynon-Davies [83]	X				X	Method deviation
Bygstad et al. [49]	X					
Chen et al. [87]		X	X			Method deviation
Chevers and Whyte [107]		X				
Clark et al. [96]		X	X			Achieved contribution
Coleman and Verbruggen [108]	X					
Dada and Sanusi [109]		X				
De Sousa et al. [110]	X					
Dennehy and Conboy [80]		X	X			Competing rationalities

Authors	Phase: Adoption	Phase: Use	Type of reasoning: Rational	Type of reasoning: Irrational	Type of reasoning: Non-rational	Research theme (included in Table 4)
Fitzgerald [67]		X				
Fitzgerald [68]		X				
Fruhling et al. [59]	X					
Fruhling and De Vreede [58]	X					
Fruhling and Zhang [111]		X				
Gandomani et al. [55]	X					
Gandomani et al. [112]	X					
Gandomani et al. [113]	X					
Ganis et al. [114]	X					
Gannon [115]	X					
Ghani and Bello [116]	X	X				
Hajjdiab and Taleb [117]	X					
Hajjdiab et al. [52]	X					
Hardy et al. [61]		X				
Hazzan and Dubinsky [90]	X		X			Understanding operationalization
Heikkilä et al. [11]		X	X	X		Method deviation
Häggmark and Ågerfalk [10]		X	X			Method deviation, Competing rationalities
Iivari and Huisman [69]		X				
Iivari and Maansaari [62]		X				
Jayakody and Wijayanayake [118]		X				
Jamil and Al-Hajry [50]	X					
Jones and Kydd [119]	X					
Joseph et al. [60]		X				
Karlsson [120]		X				
Käpyaho and Kauppinen [121]		X				
Laanti et al. [54]	X					
Lagerberg et al. [122]	X	X				
Licorish et al. [123]	X	X				
Livermore [124]	X					
Livermore [125]	X					
Lui and Chan [126]		X				
Mahadevan et al. [127]	X					
Mannaro et al. [128]	X					
Marks [129]	X	X				
Masrek et al. [130]		X				
Maulana and Raharjo [131]	X					
McAvoy and Butler [77]	X			X		Competing rationalities
McAvoy and Butler [78]	X			X		Competing rationalities
McAvoy et al. [56]	X					
Middleton [84]	X	X	X			Method deviation
Middleton [85]		X	X			Method deviation
Middleton and McCollum [86]		X	X			Method deviation
Mishra and Mishra [72]	X		X			Choice of SDM
Mnkandla and Dwolatzky [92]		X		X		Understanding operationalization
Mohagheghi and Lassenius [132]	X					
Mohallel and Bass [94]	X			X		Achieved contribution
Mohan et al. [81]		X	X			Understanding operationalization
Mäki-Runsas [91]		X	X	X	X	Understanding operationalization

	Phase		Type of reasoning			
Authors	Adoption	Use	Rational	Irrational	Non-rational	Research theme (included in Table 4)
Nelson and Teng [133]		X				
Patanakul and Rufo-McCarron [82]	X			X	X	Achieved contribution1, Method deviation2
Poston et al. [53]	X					
Procter et al. [64]		X				
Quelal et al. [134]		X				
Rahim et al. [135]	X	X				
Rezgui et al. [136]	X					
Rezgui et al. [137]	X					
Ribeiro and Domingues [138]	X					
Rindell et al. [139]		X				
Roberts Jr et al. [93]	X			X		Achieved contribution
Roberts Jr et al. [140]	X					
Roberts et al. [141]	X					
Roberts et al. [142]	X					
Rowlands [143]		X				
Saarinen [88]		X		X		Method deviation
Salo and Abrahamsson [65]		X				
Sauer and Lau [76]	X		X			Competing rationalities
Schindler [144]	X	X				
Serour and Winder [145]	X					
Snook and Harrison [146]	X	X				
Šochová [147]		X				
Soundararajan et al. [95]		X	X			Achieved contribution
Strode et al. [79]		X	X			Competing rationalities
Taylor et al. [73]	X		X			Choice of SDM
Tekbulut et al. [148]		X				
Toleman et al. [57]	X					
Tripp and Amstrong [74]	X			X		Choice of SDM
Tudor and Walter [149]	X					
Vavpotic and Bajec [150]	X					
Vavpotič et al. [151]	X					
Vavpotič and Hovelja [45]	X					
Vejandla and Sherrell [71]	X		X			Choice of SDM
Vijayasarathy and Turk [70]		X				
Vrhovec [152]		X				
Weiss and Brune [153]	X					
Yli-Huumo et al. [154]		X				
Yoshii and Higa [155]		X				

Note: [1]Irrational, [2]Non-rational

Biographical notes

Tanja Elina Havstorm
Tanja Elina Havstorm is a university lecturer and Ph.D. Candidate in Informatics, at Örebro University, at the Centre for Empirical Research on Information Systems (CERIS). Her current research and teaching interests include management in information systems, software development methods (SDM), and requirements engineering. Her current research focuses on the effective use of management techniques in software development organizations by researching the challenges in succeeding with agile SDM use, such as cargo cult behavior. Havstorm has worked as a lecturer in information systems at Örebro University, in Sweden, since 2014 and has been teaching students on both the Bachelor's and Master's levels. Havstorms' dissertation 'Cargo cult in Agile Software Development' is planned for Autumn 2023. More information is available at https://orcid.org/0000-0001-8604-8862

Fredrik Karlsson
Fredrik Karlsson is professor in Informatics at Örebro University, Sweden. He received his PhD in Information Systems Development from Linköping University. His research on information security, tailoring of systems development methods, system development methods as reusable assets, CAME-tools, requirements engineering, method rationale and electronic government has appeared in a variety of information systems journals and conferences. Some examples are European Journal of Information Systems, Journal of the Association for Information Systems, Information and Software Technology, Strategic Journal of Information Systems, and Scandinavian Journal of Information Systems. More information available at https://orcid.org/0000-0002-3265-7627