Cultural influences and success factors at agile methodologies in software development – A systematic literature review
1. Introduction

In the past two decades, we have seen the increasing use of agile approaches in software development [VC19]. Various approaches, such as Scrum or Extreme Programming (XP), have become established in practice during this time and continue to spread rapidly internationally. The motivation for using agile methodologies is justified by the fact that with the help of these iterative models, a better reaction to changing requirements is made possible [CLC04]. Due to the iterative approach of these models and the associated frequent coordination within the development team, social components such as communication are of great importance.

Software development projects are increasingly realized with international cooperation [CRM11], [HM01], [KSW04]. This international cooperation inevitably arises with the increasing globalization of software development (GSD) [CÅO+09]. There are various challenges associated with GSD. Several studies (c.f. [EN01], [NWD08], [Pur12], [SRU12]) address the importance of communication, for example. Other studies point to the special nature of knowledge management [DS14]. Another important aspect associated with global software development is the influence of cultural differences (cf. [HCA+06], [HJP15], [HT07], [MHK05a], [OB06], [SGS16], [SM10], [SSA+11], [Su15]).

The cultural influence on the cooperation of people in software development has been proven by several studies (c.f. [ADD14], [CKW+15], [Cur02]). It is also known that agile methodologies are used in global software development (c.f. [BMB15], [Hos08], [JW12]). One can assume that different cultures affect agile methodologies. The question, therefore, arises as to how cultural influences affect procedural models. To classify cultural influences on agile methodologies, an understanding of the success factors of these approaches is also necessary. Likewise, it is not known whether a cultural influence in different contexts leads to an adaptation of the application of agile and hybrid approaches.

This systematic literature review aims to answer the following research questions:

- **RQ1**: Which success factors influence agile methodologies in software development?

- **RQ 2**: Which cultural factors are relevant when using agile methodologies in software development?

- **RQ 3**: Are there adjustments to agile methodologies based on the knowledge that cultural differences influence these models?
In this paper, we first describe the background of this SLR in Section 2. In Section 3, we explain the research protocol, the research method, and our approach. We present the results of this SLR in Chapter 4. Before we summarize the SLR in section 6, we delineate the limitations of this paper in Chapter 5.

2. Background

In this section, we describe the background of the SLR. We give a brief introduction to agile methodologies and cultural influences on software engineering.

2.1. Agile methodologies in software development

Software development process models (procedure models) describe how a result (what) based on a process model (when) from which role (who) within which activity (how) and with the help of which tools and methods (with what) should be realized (c.f. [BK13], [Roy70]). Procedure models are a central component of software engineering. They are used to reduce the complexity of software development projects and structuring activities. In addition to phase-oriented procedure models, agile approaches are used today. Components of software processes of the two types are also combined. This combination is known as a hybrid approach.

Agile methodologies originated in the United States in the mid-1990s and are often understood as a reaction to phase-oriented processes (c.f. [AGW08], [DDM10]). With the creation of the agile manifesto in February 2001, the authors created a uniform basis for agile approaches in software engineering [BBv+19]. In addition to four pairs of values, the agile manifesto also describes 12 principles and thus defines a value-based basis for agile approaches [Wil10]. At least since the publication of the agile manifesto, agile methodologies in software development have been in constant circulation, which continues to this day. Today, almost 20 years later, they are an established part of software development approaches in research and practice. In addition to Scrum, which is used most internationally, well-known and frequently used agile approaches are Extreme Programming (XP), Kanban, and Adaptive Software Development (ASD) (c.f. [KHP+17], [VC19]).

Agile approaches usually have specific characteristics in common. Abrahamson et al. [ASR+02] describe the following properties:

- Incremental: Small software releases in short cycles
- Cooperative: Customers and software developers work closely together in close coordination
• Uncomplicated: The procedure is easy to learn, to adapt and the rules are well documented
• Adaptable: It is possible to make changes to the product at the last possible moment.

The use of agile methodologies is usually associated with practices and roles defined in frameworks, e.g., for Scrum, the Scrum Guide [SS17]. These agile practices play a central role in adopting agile approaches. They can have different characteristics. For example, various ceremonies (meetings) are described for Scrum. In XP, however, the focus is on defining agile practices in technical activities such as testing or refactoring. In practice, countless variations of agile practices and methodologies are used. The high number of these differences is often justified by the adaptations in practice (c.f. [AWS+03], [FHC06], [Qur12], [WCP12]). Different agile methodologies are also used in combination [WCC12].

2.2. Cultural influences on software development

Cultural influences on software engineering (c.f. [Bor03], [CdC15]) and the implementation of software development projects (c.f. [CKW+15], [MW13]) are described in the literature. Likewise, the cultural influence on related disciplines such as requirements engineering [Sch14], project management (c.f. [Bau13], [LL16], [Lüc15]) and process optimization (c.f. [BMD02], [GS11]) is demonstrated in several studies.

The term ‘culture’ is not uniformly defined. The context is of crucial importance for the definition of the concept of culture (c.f. [Ham07], [KK52], [Oli95]). For this SLR, Oettinger’s definition of culture is used [Oet93, S. 41]: “... the term is used to describe the customs, beliefs, social structure, and activities of any group of people who share a common identification and who would label themselves as members of that group.”

In addition to the definition of the concept of culture, a description and comparability of cultures are necessary. Models are usually used for this (cf. . [AM10], [Bor03], [DM02], [FB17], [WK08]). These models can be divided into models that describe and compare cultures.

A variety of models are used in the literature in the context of software engineering:

<table>
<thead>
<tr>
<th>Type</th>
<th>Cultural model</th>
<th>Used in primary studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptive</td>
<td>Cultural Onion Model by Hofstede [Hof93, S. 22ff.]</td>
<td>[AJL11], [AVR+09], [JHL11], [SKS+04]</td>
</tr>
<tr>
<td></td>
<td>Levels of Culture by Karahanna et al. [KES06]</td>
<td>[AB08b], [LK06], [MM11]</td>
</tr>
</tbody>
</table>
In software engineering, the culture-describing models of the cultural levels, according to Karahanna et al. and the onion model by Hofstede are relevant. The model by Karahanna et al. is understood as a model for presenting different cultural levels and their relationships to one another. Hofstede's onion model focuses on the imprinting of people based on cultural properties and the visibility of these properties.

Culture-comparing models categorize critical figures in the form of dimensions. These dimensions serve to compare and interpret the cultures of concrete contexts, such as nation-states or companies. The model with the highest prevalence is the basic national values, according to Hofstede. This is also highly relevant outside of software engineering.

### 3. Research methodology

We select the systematic literature research (SLR) approach, according to Kitchenham and Charters [KC07], to ascertain the current state of research. We describe the individual steps, the associated goals, and the activities below. First, the purposes of this SLR are described and substantiated by the research questions. The search strategy is then set out in Chapter 3.1. The selection process used to select the relevant primary studies is later explained (see Chapter 3.2). We present the procedure for updating the results in Chapter 3.3 and subchapter 3.3.1. Finally, Chapter 3.4 and the quality check, as well as Chapter 3.5, describe the extraction of the data from the primary studies.

Research questions substantiate the objectives of an SLR. We describe these research questions below:
- **RQ1**: Which success factors influence agile methodologies in software development?
  With the help of this question, potential success factors that come into play when using agile methodologies are to be identified. The aim is to work out whether a cultural reference can be made to the success factors of agile software development.

- **RQ 2**: Which cultural factors are relevant when using agile methodologies in software development?
  Concerning this research question, the extent to which cultural aspects influence agile methodologies is to be identified. The essence of this research question is to find out whether research results are available that describe cultural factors that affect the use of agile methodologies in software development. Regarding the influence of cultural factors, the cultural levels described in Chapter 2.1.2 must be taken into account: national culture, regional culture, organizational culture, corporate culture, departmental culture, and subculture.

- **RQ 3**: Are there adjustments to agile methodologies based on the knowledge that cultural differences influence these models?
  The third research question requires that results were identified for the previous research question. This question aims to show whether agile procedural models can be adapted for use in software development due to cultural influences. If this is the case, it should also be taken into account how this adaptation was carried out, for example, which agile characteristic was adapted and how.

Kitchenham and Charters recommend in [KC07] preparing a research report for traceability. Usually, the research protocol represents the systematic description of the procedure of the SLR. This research protocol serves for the repeatability of an SLR. It also helps to reduce the bias of researchers. The following chapters, 3.1 to 3.5, present the content of the research protocol.

### 3.1. Search strategy

In the second phase, following the guidelines of Kitchenham and Charters [KC07] on the implementation of the SLR, the search strategy is initially defined. For this purpose, we first determined the search terms for literature research. The search terms are composed of several keywords. The definition of search terms is necessary to limit the number of results in the literature concerning the topics that are relevant for answering the research questions. The SLR selection process is then described and carried out. In the course of
this, we defined inclusion and exclusion criteria to facilitate the examination regarding the content relevance for this SLR.

Kitchenham and Charters recommend extracting the essential search terms from the research questions in [KC07]. The following subject areas result in the present work:

- Agile methodologies in software development
- Cultural aspects
- Success factors
- Influencing factors

In the following section, keywords are defined according to the subject area (see Table 2). As recommended by Kitchenham and Charter in [KC07], possible synonyms, related terms, or alternative spellings are used. The keywords are linked with Boolean operators to be able to build thematic search terms (see Table 1).

The quality of the keywords for the identification of relevant literature is crucial for the identification of relevant sources in the SLR [KC07]. Kitchenham and Charters describe in [KC07] that, for example, iterative test searches with different search terms, checking the result sets for known primary studies, and consulting experts are ways to ensure the quality of the search terms.

The search terms were refined separately for each subject area:

- For the topic of agile methodologies, iterative test runs are carried out first. In these searches, we continuously refined the keywords after checking the result set. For example, we found that taking widespread agile methodologies (see Chapter 2) into account makes it easier to consider publications that deal with concrete methodologies. These methodologies are taken into consideration as a keyword in a written form and, if available, in an abbreviated form.

- As with the previous topic, iterative test visits were carried out on the subject of cultural aspects. It turned out that it makes sense to consider the terms ‘factor’ and ‘influence’ in addition to the keyword, ‘culture’. This is mainly due to the narrowing of the result set about the research questions of this SLR. In the case of the keyword ‘culture’, there is also the peculiarity that the word ending is not clear in the context of this work. For example, it is conceivable that the term ‘culture’, as well as culture-relevant literature, can be identified. For this reason, the word extension is generally permitted for the keyword ‘culture’.
• The search terms derived from the research questions of the SLR are completed by the topic's success factor and influencing factor. For these topics, the English-language terms are used as search terms, since no effects concerning the result set could be observed in the iterative test visits with the use of synonyms.

<table>
<thead>
<tr>
<th>Subject area</th>
<th>Search terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile methodologies</td>
<td>&quot;agile software development&quot; OR &quot;agile development&quot; OR &quot;agile methodology&quot; OR &quot;scrum&quot; OR &quot;xp&quot; OR &quot;extreme programming&quot; OR &quot;kanban&quot; OR &quot;test driven development&quot; OR &quot;tdd&quot; OR &quot;feature driven development&quot; OR &quot;fdd&quot;</td>
</tr>
<tr>
<td>Cultural aspects</td>
<td>cultur* OR &quot;cultural impact&quot; OR &quot;cultural factors&quot;</td>
</tr>
<tr>
<td>Success factor</td>
<td>Success factor</td>
</tr>
<tr>
<td>Influencing factor</td>
<td>Influencing factor</td>
</tr>
</tbody>
</table>

Table 2: Allocation of the search terms to the subject areas

In the next step, we created a compound search term with which the search can be carried out. For this purpose, keyword groups are first formed based on the four subject areas. These are linked with Boolean operators:

<Agile methodologies> AND <Cultural aspects> AND <Success factor> AND <Influencing factor>

To be able to use the compound search term for research in the knowledge databases, it is necessary to replace the keyword-based keyword groups with the keywords assigned in Table 2. This results in the following search term:

("agile software development" OR "agile development" OR "agile methodology" OR "scrum" OR "xp" OR "extreme programming" OR "kanban" OR "test driven development" OR "tdd" OR "feature driven development" OR "fdd") AND (cultur* OR "cultural impact" OR "cultural factors") AND ("influencing factors" OR "success factors")

The search term is used manually in the present SLR for research in the two scientific digital libraries ACM Digital Library and ScienceDirect (see Table 3). Other libraries were not considered because the results in the various libraries show high redundancies. When searching in different digital libraries, it must be taken into account that the filter settings have to be configured differently depending on the library (see Table 3).

<table>
<thead>
<tr>
<th>Digital library</th>
<th>URL</th>
<th>Filter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Digital Library</td>
<td>dl.acm.org</td>
<td>• Use of the Advanced Search functionality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use of the library “The ACM Full-Text collection”</td>
</tr>
</tbody>
</table>
Due to the differences in the digital libraries (see Table 3), a separate search term was used for ACM and ScienceDirect:

### Table 3: Digital libraries and the settings used in March 2017

<table>
<thead>
<tr>
<th>Digital library</th>
<th>Search term</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Digital library</td>
<td><code>content.ftsec:(&quot;agile software development&quot; OR &quot;agile development&quot; OR &quot;agile methodology&quot; OR &quot;scrum&quot; OR &quot;xp&quot; OR &quot;extreme programming&quot; OR &quot;kanban&quot; OR &quot;test driven development&quot; OR &quot;tdd&quot; OR &quot;feature driven development&quot; OR &quot;fdd&quot;) AND (cultur* OR &quot;cultural impact&quot; OR &quot;cultural factors&quot;) AND (&quot;influencing factors&quot; OR &quot;success factors&quot;))</code></td>
</tr>
<tr>
<td>ScienceDirect</td>
<td><code>(&quot;agile software development&quot; OR &quot;agile development&quot; OR &quot;agile methodology&quot; OR &quot;scrum&quot; OR &quot;xp&quot; OR &quot;extreme programming&quot; OR &quot;kanban&quot; OR &quot;test driven development&quot; OR &quot;tdd&quot; OR &quot;feature driven development&quot; OR &quot;fdd&quot;) AND (cultur* OR &quot;cultural impact&quot; OR &quot;cultural factors&quot;) AND (&quot;influencing factors&quot; OR &quot;success factors&quot;))</code></td>
</tr>
</tbody>
</table>

Table 4: Used search terms in March 2017

We performed the search on March 22nd, 2017. Based on the search settings described in Table 3 and search terms presented in Table 3, 296 potentially relevant primary studies were identified. Of these, 171 were for ACM and 125 for ScienceDirect.

### 3.2. Study selection

The selection criteria are defined as inclusion and exclusion criteria and are used to identify the relevant studies to answer the research questions of this SLR. The selection criteria include both structural and content-related aspects for the paper selection.
Every paper must meet the inclusion criteria IC1, IC2, and IC3. For criteria IC4 to IC9, it is sufficient if the paper to be selected fulfills one of these criteria. The inclusion criteria are defined as follows:

- IC1: Papers published between 2007 and April 2020
- IC2: Papers published in the field of information technology
- IC3: Papers in English or German
- IC4: Papers dealing with success factors for agile methodologies in the context of software engineering
- IC5: Papers dealing with success factors for global software development
- IC6: Papers dealing with success factors for distributed software development
- IC7: Papers dealing with cultural (or other) influencing factors in the context of software development
- IC8: Papers dealing with cultural (or other) influencing factors on agile methodologies
- IC9: Research summaries that relate to the criteria mentioned above

The following structural exclusion criteria are also defined. EC1, EC2, and EC3 are the criteria specified at the start of the selection process. The criteria EC4 and EC5 were established later in the selection process, and here in particular when checking the border cases:

- EC1: Gray literature / unpublished work: technical reports, not published at specialist conferences or in specialist journals
- EC2: Contributions from specialist conferences or magazines with less than five pages
- EC3: Publications that have not undergone a peer-review process (such as specialist books or theses)
- EC4: Papers that describe or define the maturity levels of a software development process
- EC5: Papers dealing with software engineering and related topics (such as requirements engineering or project management) without a cultural connection

A four-stage selection process is carried out based on the defined inclusion and exclusion criteria. This selection process is an adaptation of the process described by Kitchinham and Charters in [KC07]. The four-stage process of this SLR is detailed below:

- **First step:** Checking the title and keywords of the respective source for relevance based on the criteria IC1 to IC9 and EC1 to EC5 described above.
- **Second step:** Checking the abstract of the corresponding source concerning the content-related relevance. If it is unclear whether a source appears relevant in terms of content or not, this will be verified as an uncertain case in the next step.

- **Third step:** The content relevance is validated by reading the corresponding introductory text and the summary of the potentially relevant sources. As in phase two, the papers that cannot be safely included or excluded due to the available information are included as potentially relevant primary studies in the next step.

- **Fourth step:** The entire content of the sources is read to check the relevance overall.

After the selection process had been carried out, 37 out of a total of 281 sources were classified as relevant for the SLR (see Figure 1). Based on the structural exclusion criteria, three publications (EC1: 1; EC2: 2) were not considered. Furthermore, 12 duplicates were identified in the result set of the digital library ACM. These were removed from the result set. The result set of the primary studies to be selected thus included 266 sources.
To be able to subject this result set to a content-related selection, the selection process described above has been carried out. This content-based selection, for example, excluded 45 papers in the first step, i.e., when checking based on the title and keywords. A result set of 221 primary studies was available for the second step of the selection process. At this selection level, reading the abstract, a further 75 primary studies are excluded. For the third stage of the selection process, 146 papers were selected by reading the introduction and the summary. We removed 48 papers from the set. The remaining 98 potentially relevant primary studies were subjected to the fourth selection step, in which we verified the content of the source. A further 61 primary studies were excluded.
The result of this selection, therefore, comprises 37 sources after the four process steps have been carried out.

Kitchenham and Charters describe in [KC07] that validation of the intermediate results of the multi-stage selection process makes sense. The exclusions from publications are particularly relevant for this validation. In this SLR, this validation was carried out by the supervisor of the underlying work to ensure the reliability of the literature selection. In this context, the content-related inclusion and exclusion criteria (IC4 to IC9 and EC4 and EC5) have been refined, particularly when discussing borderline cases.

### 3.3. Updating the result set

Several years have passed between the creation of this SLR at the beginning of the research project and the finalization of the present work. It can therefore be assumed that further potentially relevant sources for the result set of the SLR were published during this time. It is advisable to carry out the SLR process again to ensure a current summary of the available literature. We therefore carried out the following procedure:

1. **Validation of the search terms used in 2017 (see Table 3):**
   The searches should be carried out in the two digital libraries ACM and ScienceDirect with identical search terms and filter settings from the first search run on March 22, 2017. We compared the result set with that from 2017. This is motivated by ensuring that the algorithms of the search function in the digital libraries ACM and ScienceDirect have not changed and that the new result is therefore valid.

2. **Adjustment of the filter settings:**
   The adjustment of the time filter settings is necessary to be able to take into account the new, potentially relevant sources for this SLR. For this purpose, the time horizon is widened from 2017 to April 2020. Furthermore, former filters have been changed in the new surfaces. Some filters are no longer available, and new ones have been added. It is essential to examine these changes. If necessary, the filter settings have to be adjusted and documented accordingly.

3. **Checking the result set and incorporating new results into the SLR:**
   If the result set has changed, it is necessary to include the partial result set of the new potentially relevant results in this SLR.
The following description of the re-execution of the search and the execution of the selection process serves, in particular, to make this SLR traceable and repeatable from Chapter 4. The updated result set is used to increase readability.

1. **Validation of the search terms used in 2017 (see Table 3):**

   A new search with the filter settings used initially is necessary for the validation of the search terms. We performed the search for the first time on April 6, 2020, at the digital library ACM. We found that the design of the web portal, and thus, the user interface of ACM was renewed. The filter settings used in March 2017 (see Table 2) are affected by these changes in the user interface and must be adjusted accordingly. For example, the filter setting of the publication type, such as "Books," is possible on the results page and no longer in the search mask. The search term can now be used in a slightly modified form at ACM. It is no longer necessary to use the `content.ftsec:()` function for the full-text search, since a corresponding selection is now integrated into the user interface of the web portal. The result set on April 6, 2020, taking into account the relevant filter settings, corresponds to that of March 22, 2017.

   When validating the search term for ScienceDirect, the adaptation of the user interface of the web portal also became apparent. Furthermore, the previous search (referred to as Expert Search) was replaced by a new one (under the name Advanced Search). Due to the associated changes, the search term from March 2017 is no longer possible because:

   - The number of characters in the search term is limited to 250
   - The number of Boolean operators is limited to 8
   - Wildcards cannot be used

   This means that the result set cannot be validated using the above search term. We used the following three measures to solve this problem:

   1. **Identification of potential solutions with the support of ScienceDirect**

      Upon request, ScienceDirect Support confirmed that the old search function (Expert Search) is no longer possible. It is not possible to temporarily bypass the validators of the new Advanced Search or to use filter settings in the Expert Search.
2. **Use of the old search functionality (expert search) by calling the web service operation using the search term from March 2017**

It is not possible to access Expert Search through the web service method used previously. As a result, using this method leads to an error page. The old web service has been deactivated and can no longer be used.

3. **Iterative adaptation of the search term by carrying out pilot searches repeatedly**

The aim of this determination of a new search term is to obtain an approximately identical result set with the original one from March 2017. For this purpose, we performed several pilot searches to adapt the search term so that it can be used with the new search function.

First, we adjusted the search term concerning the character length and number of Boolean operators. We broke down the search term into several shortened search terms. An additional challenge arises from using the placeholder since this is no longer possible in the new search. In the original search term, a wildcard in the topic area Cultural Aspects is used for the keyword *cultur* (see Table 3). To check whether it is sufficient to replace the placeholder with similar terms from *culture*, we examined the result set from March 2017 for related and similar terms. It turned out that the use of the following terms seems sufficient: *culture* and *cultural*.

Adjusting the search term leads to four new search terms for ScienceDirect. The following table shows the adapted search terms:

<table>
<thead>
<tr>
<th>Digital library</th>
<th>Search term(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Digital library</td>
<td>(“agile software development” OR “agile development” OR “agile methodology” OR “scrum” OR “xp” OR “extreme programming” OR “kanban” OR “test driven development” OR “tdd” OR “feature driven development” OR “fdd”) AND (cultur* OR “cultural impact” OR “cultural factor”) AND (“influencing factors” OR “success factors”)</td>
</tr>
<tr>
<td>ScienceDirect</td>
<td><strong>Search term I:</strong> (“agile” AND (“development” OR “methodology”)&quot;) AND (culture OR cultural OR “cultural impact” OR “cultural factor”) AND (“influencing factors” OR “success factors”)</td>
</tr>
<tr>
<td></td>
<td><strong>Search term II:</strong> (“scrum” OR “xp” OR “extreme programming”) AND (culture OR cultural OR “cultural impact” OR “cultural factor”) AND (“influencing factors” OR “success factors”)</td>
</tr>
</tbody>
</table>

Since we used four search terms for ScienceDirect in 2017, it is necessary to perform the search for this digital library four times. The result sets are checked for duplicates and then combined into a result set. This result set can, in turn, be compared with that from March 2017.

The results of the pilot searches have shown that it is not possible to generate the result set from 2017 using the keywords used initially. The reviews showed that 64% of the 2017 sources are in the new result set. The validation of the missing sources has been demonstrated that the algorithm of the new Advanced Search from ScienceDirect appears to be more precise. For example, the sources are not included in the new result set, where individual keywords (such as "Agile Software Development") cannot be identified in the text. Likewise, the combination of the keyword groups is considered more precisely in the new Advanced Search. Due to these changes in the search function, it was necessary to adapt the search term, especially in the field of agile methodologies. The adjustment aimed to allow a broader set of results so that the result set from March 2017 could be fully reflected. The search term I (see Table 5) has therefore been adjusted as follows concerning the keywords: "agile software development" OR "agile development" OR "agile methodology": ("agile" AND ("development" OR "methodology")).

The adaptation of the search term I allows a higher degree of potential combinations of the keywords. With the help of this change, all primary studies from the result set from March 2017 could be included in the new result set.

2. Adjustment of the filter settings:

As described above, the options for filter selection and settings have changed in the meantime. When adjusting the filter setting, the time horizon must be taken into account first. Since there is already a result set for the period from January 2007 to
March 2017, the period from March 2017 to April 2020 must be taken into account for the renewed search. With ACM, this period can be taken into account at the monthly level by selecting the "Publication Date" filter. No monthly selection is possible with ScienceDirect. Accordingly, filtering is carried out here for the years 2017 to 2020 inclusive.

The following table provides an overview of the settings used:

<table>
<thead>
<tr>
<th>Digital library</th>
<th>URL</th>
<th>Filter settings</th>
</tr>
</thead>
</table>
| ACM Digital library   | dl.acm.org     | • Use of the Advanced Search functionality  
• Use of the library The ACM Full-Text collection  
• Use of the full-text search (via selecting “Anywhere” in filter “Search Within”)  
• Publication Date: Custom range from March 2017 to April 2020 |
| ScienceDirect         | sciencedirect.com | • Use of the Advanced Search functionality  
• Use of the full-text search (via “Find articles with these terms”)  
• Filter “Article types” activated for these types: “Review articles”, “Research articles”, “Editorials”  
• Time range filter enabled: 2017 – 2020 (month selection not possible) |

Table 6: Digital libraries and the settings used in April 2020

Another change in the filter settings of the new Advanced Search at ScienceDirect is the extended option to filter the type of publication (article type1). For the new search, the types, "Review articles," "Research articles," and "Editorials" are filtered here. The pilot searches showed that the sources to be classified as potentially relevant in the result sets were assigned to one of these three types. The previously used filter option "No Books" is no longer available in either of the digital libraries. The change in the filter settings described above for executing the full-text search must also be observed. Since the filter setting for one (or more) subject areas is no longer possible,

---

1 A list of the available “Article types”, as well as their description, is presented in the online presence of ScienceDirect in [Els20].
the subject affiliation of the sources in the result set must be validated manually by ScienceDirect.

3. Checking the result set and incorporating new results into the SLR:

At ScienceDirect, we performed four search iterations using the search terms I to IV (see Table 4), as described in point 1 above. These four searches resulted in a result set of 476 sources (see Figure 2).

![Diagram of search terms and result set](image)

We then checked this result set for duplicates. Fifty-eight duplicates were identified and removed from the result set. In the next step, we examined the result set for non-specialist sources. All sources that could not be assigned to computer science were regarded as foreign to the subject. Since a corresponding filter setting is no longer available with ScienceDirect, we did this filtering manually. The manual screening was carried out by checking the "Publication title", i.e., the title of the respective medium, and by validating the title. We identified 247 primary studies as non-specific for computer science and excluded them. We then compared the remaining set of 171
papers with the result set from 2017 for possible duplicates. This is necessary because the period with ScienceDirect cannot be limited to the monthly level, and therefore a redundant consideration of the months January, February, and March in 2017 arises. The comparison showed that four sources were already included in the 2017 result set, and we removed them from the new result set. This leaves 167 potentially relevant papers in the new result set of the digital library ScienceDirect.

For ACM, we identified the sources to be checked with the adapted filter settings for the new user interface and the use of search term 37.

The result set from April 2020 thus comprises 204 potentially relevant sources (see Figure 3). First, we checked these primary studies for any duplicates between the results from ScienceDirect and ACM. No duplicates could be found here. Afterwards, we verified the structural inclusion and exclusion criteria. We found 12 sources to be unpublished (EC1). Seven sources with less than five pages have also been identified (EC2). We also identified a duplicate. We removed these 20 sources to give a result set with 184 potentially relevant primary studies. For this result set, the four-stage selection process described above is carried out to determine the relevant sources for this SLR.

In the first step of validating this set, we checked the title and keywords and excluded 110 sources as not relevant. The high number of excluded sources can be justified by the fact that the keywords were not taken into account when validating non-specialist sources.
By examining the abstract in the second step of the selection for the remaining 74 sources, we excluded 28 further papers so that for the third step there were still 46 sources to be checked. In this third step, the validation of the introduction and summary of the primary studies, nine were excluded. For the fourth and final step of the content-related selection of the primary sources, the set comprised 37 primary studies to be checked. We classified 22 of these as relevant for this SLR.

We merged these 22 sources into the result set of 2017. These are treated and presented below as a uniform result set.
3.3.1. Merging the result sets

For a fuller description of the SLR, and for reasons of comprehensibility, e.g., for the presentation of the data extraction, it makes sense to combine both result sets of the selection processes from 2017 (see Chapter 3.2) and 2020 (see Chapter 3.3). The merge also increases readability, since it enables the handling of one result set. The merged result set contains 59 papers, 37 from the result set of 2017 and 22 from 2020.

3.4. Quality Assessment

According to Kitchenham and Charters in [KC07], the quality of an SLR can be checked in two ways. The first way involves selecting the primary studies based on inclusion and exclusion criteria. This aspect has been taken into account in the present SLR as part of the step-by-step selection, i.e., the implementation of the SLR (see section 3.2). The second way is to carry out a qualitative evaluation of the primary studies. According to Kitchenham and Charters in [KC07], this assessment of quality can support the data analysis and synthesis of the publications. This second type of quality assurance for the present SLR is described below.

Kitchenham and Charters recommend a checklist in [KC07] for checking and ensuring the quality of the individual studies. Various checklists for quality assurance for SLRs have been presented in the literature (e.g. [DD08b], [KC07], [KBB16], [SN07]). Based on Kitchenham and Charters [KC07], as well as Dybå and Dingsøyr [DD08b], we developed the following checklist for this SLR:

<table>
<thead>
<tr>
<th>ID</th>
<th>Quality criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA1</td>
<td>Is the finding of the paper based on research?</td>
<td>• What research design was used? (Quantitative, qualitative, mixed, other)</td>
</tr>
</tbody>
</table>
| QA2 | Is the research goal clearly described? | • Are research questions defined and documented in the paper?  
• Is the research context described? |
| QA3 | Does the paper meet scientific quality criteria? | • Is the validation of the data described?  
• Has the reliability of the data been explained?  
• Is there documentation of the data? (e.g., statistical analysis for quantitative research or any interview protocols for qualitative research) |
Are the findings validated?  

- Have the findings of the publication been validated? (e.g., validation of hypotheses concerning a theoretical model)

Table 7: Quality criteria

According to Kitchenham et al., in [KBB16], consistency is essential for carrying out the quality assessment. For the quality assessment of the primary studies, we used a yes / no evaluation in this SLR (cf. Kitchenham et al. in [KBB16] and Dybå and Dingsøyr [DD08b]). In this context in [KBB16], Kitchenham et al. emphasize the importance of the validation of the quality assessment by other scientists. This validation has been ensured by cross-checks of the second and third authors.

3.5. Data extraction

We used two different applications to document the extracted data. The primary studies have already been imported into Citavi\(^2\) for selection (see section 3.2). When importing, the base data of the respective publication is automatically transferred to Citavi. In addition to information about the author or authors, this base data also includes the title or year of publication. Other imported data differ depending on the publication type. For example, in the case of a journal article (paper), the respective magazine is adopted. In the case of a book (e.g., a monograph), on the other hand, this information is omitted. Furthermore, we stored the summary of the respective publication manually in Citavi. In addition to Citavi, Microsoft (MS) Excel was used. We created a data extraction form in MS Excel. We used this form primarily to record and document the manually extracted information from the primary studies. Table 8 shows the categorized content:

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Basic information (in Microsoft Excel)</td>
<td>ID (current number), author/s, title, year of publication, link to the publication, digital library, abstract</td>
</tr>
<tr>
<td>E2</td>
<td>Information of the publication (in Citavi)</td>
<td>DOI, type of publication, information depending on publication type – such as publisher, place of publication, magazine, volume, pages, ISBN, number of pages, etc.</td>
</tr>
<tr>
<td>E3</td>
<td>Agile methodologies (in Microsoft Excel)</td>
<td>Agile methodology (e.g. Scrum, XP, etc.)</td>
</tr>
</tbody>
</table>

\(^2\) Citavi is an application for literature management (reference manager) and knowledge organization.
<table>
<thead>
<tr>
<th>E4</th>
<th>Cultural aspects (in Microsoft Excel)</th>
<th>Cultural level (national culture, regional culture, organizational culture, corporate culture, departmental culture, subculture) and core messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5</td>
<td>Success factors (in Microsoft Excel)</td>
<td>Which success factors are examined in the respective publication? What do they affect, and how is this influence justified?</td>
</tr>
<tr>
<td>E6</td>
<td>Adaptation of agile methodologies (in Microsoft Excel)</td>
<td>Described adaptation of the agile methodologies and justification or motivation</td>
</tr>
<tr>
<td>E7</td>
<td>Research design (in Microsoft Excel)</td>
<td>Quantitative, qualitative, mixed, other</td>
</tr>
</tbody>
</table>

| Table 8: Overview of the content of the data extraction |

Based on the categories, the individual elements, if necessary, are explained in more detail below, and the associated procedure for data extraction is described:

- **Basic information (E1):**
  With two exceptions (digital library and the summary), the basic information was extracted automatically by the Citavi import. Both sets of information have been stored manually in Citavi. After the manual re-entry of this content, the basic information was exported to an MS Excel file and then transferred to the extraction form. The basic information thus represents the content-based basis for the MS Excel form. The basic information helps to structure the relevant primary studies and thus create an overview.

- **Information on the publication (E2):**
  The publication information is available in Citavi because the import automatically extracts it. We extruded selected publication information for the MS Excel form. This includes the publication type (for example, magazine article) and, if available, the respective medium (follow the previous example: the magazine). Like the basic information (E1), the publication information is used to structure the primary studies and is used to prepare the overview.

- **Agile methodologies (E3):**
  The information on the category of agile methodologies has been extracted manually from the primary studies. We documented the data in the MS Excel form. We noticed here that no specific content could be obtained in parts. In some primary studies, more than one agile methodology is considered and mentioned.
Besides, some primary studies do not describe concrete agile methodologies but consider a higher level at which the fundamental properties of this type of methodology are relevant. If specific agile methodologies or practices are named or described in publications, we extracted these accordingly. If agile methodologies are dealt with within the publication, but these are not specified, we extracted this information as "agile methodologies".

The extraction of the information about which agile methodologies are considered in the respective publication serves for the structural classification of the primary studies and thus for an overview. Furthermore, we considered the information about which agile methodologies is relevant for answering RQ1 (see Chapter 3), since the success factors can vary depending on the agile methodology. The extraction of information from agile methodologies is also important for answering RQ3 (see Chapter 3). For answering RQ3, the possible adaptation of agile methodologies is important. Accordingly, it is important to know which agile methodologies are examined explicitly in the respective publication.

- **Cultural aspects (E4):**
  As we did with the information on agile methodologies (E3), we extracted the cultural aspects manually from the paper. We documented the extracted data in the MS Excel form. First, we extracted the cultural levels mentioned in the publication. We documented the extraction of the culture levels with a yes/no. If one or more cultural level/s were mentioned in the paper, we extracted the respective cultural influence. We documented this extraction as a so-called “core message(s)”. We documented what this cultural influence relates to, e.g., whether this influence is due to regional differences. What has been influenced was also extracted. If this information is not or not sufficiently described in the publication, we noted this during the extraction.

Information on cultural aspects and influences is necessary to answer research questions 2 and 3 (see Chapter 3). Furthermore, the knowledge of specific cultural levels and their extent was relevant for the structural classification of the publications.

- **Success factors (E5):**
  As with the two previous categories (E3 and E4), we performed the extraction manually. The extracted information was also documented analogously in the MS
Excel form. The success factors, the subject matter, and the reasons were extracted separately. For publications that did not describe success factors, an "n/a" was documented.

The information on success factors when adopting or using agile methodologies is relevant for answering RQ1 (see Chapter 3).

- **Adaptation of agile methodologies (E6):**
  The extracted information on the adaptation of agile elements or methodologies was done manually, as with E3, E4, and E5. We documented the information in the MS Excel form. We did the data extraction in two steps. As soon as the description of the adaptation of agile methodologies was identified in a publication, we extracted it. The justification for the adaptation was also extracted.

  We used the information on the adaptation of agile elements of agile methodologies, and the reason for the adaptation, to answer the RQ3 of this SLR (see Chapter 3).

- **Research design (E7):**
  We extracted the data for the research design manually. We documented the information in the MS Excel form and made a distinction between four types (see Table 7).

  The information helps to structure the primary studies and is used to create an overview.
4. Results

Before the content of the results of the SLR is presented (see Chapter 4.4.2), the following subchapter gives an overview of the primary studies relevant to this SLR. This provides an insight into the distribution (e.g. by year of publication) or structure (for example with regard to the selected research methodology) of the sources. The subsequent presentation of the results of this SLR is based on the underlying research questions (see Chapter 4.2).

4.1.1. Overview of the studies

At the beginning of the overview, we discuss the distribution of the primary studies concerning the year of publication (see Figure 4). The primary studies in the field of agile software development are spread with constant growth over the years 2007 to 2019.

![Figure 4: Visualization of the number of sources per year of publication](image)

The years 2007 to 2009, 2012 and 2013, and 2014 to 2017 always show an even distribution of the primary studies. Nevertheless, an increase in publications in the research field is discernible in 2011, 2014, and 2019. The year 2020 only takes into account the months from January to April, which explains the low proportion. Even though fewer primary studies were published in 2018, the renewed increase in 2019 confirms the statement of constant growth made above.

The majority of the 59 primary studies were published in journals (51; 86.44%). The other primary studies (8; 13.56%) were published in conference proceedings. The following table shows the distribution of the primary studies per digital library:
<table>
<thead>
<tr>
<th>Digital library</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>14</td>
<td>23.73 %</td>
</tr>
<tr>
<td>ScienceDirect</td>
<td>45</td>
<td>76.21 %</td>
</tr>
</tbody>
</table>

Table 9: Distribution of primary studies per digital library

The relevant primary studies pursue different research approaches:

- Qualitative
- Quantitative
- Mixed (qualitative and quantitative)
- Research summaries (SLRs, mapping studies, etc.)
- Others (e.g., action research)

The primary studies of this SLR are divided into qualitative (19; 32.20%) and quantitative (17; 28.81%) research approaches. Research summaries in the form of SLRs, mapping studies, or similar are also often included in the results (14; 23.73%). A mixed research approach (5; 8.47%) is rarely used. The situation is similar to other research approaches, such as action research (4; 6.78%).

Various agile methodologies are considered in the primary studies of this SLR (see Figure 5).

![Figure 5: Visualization of the frequency of the agile methodologies under consideration](image)

Scrum was used most frequently (in 17 primary studies), followed by XP (in eight primary studies). Kanban has been considered in four primary studies. The frequency of these three agile methodologies, also in this order, coincides with the prevalence in practice (cf. [ASR+02], [DD08a], [KHP+17], [VC19]). Also, other agile methodologies were considered in the primary studies of this SLR, including Feature-Driven Development (in two
primary studies), Crystal, Scaled Agile Framework, Large-Scale Scrum and Agile Software Solution Framework (each in one primary study). The combination of Scrum and XP was considered in one primary study. Several agile approaches have also been used in some primary studies (see Table 10).

<table>
<thead>
<tr>
<th>Agile methodology</th>
<th>Primary study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum</td>
<td>[AFB+19], [BCT19], [CC08], [Dru14], [FIK+15], [GTF17], [HNM11], [LGc17], [LHC+19], [LKT17], [LSD+16], [PFG13], [PL14], [RKK16], [RRO+19], [SND+13], [SS12]</td>
</tr>
<tr>
<td>Extreme Programming (XP)</td>
<td>[AFB+19], [BCT19], [BU14], [CC08], [Dru14], [HNM11], [PFG13], [SND+13]</td>
</tr>
<tr>
<td>Kanban</td>
<td>[BJ19], [FIK+15], [LSD+16], [SS12]</td>
</tr>
<tr>
<td>Feature-Driven Development</td>
<td>[CC08], [SND+13]</td>
</tr>
<tr>
<td>Crystal Family</td>
<td>[CC08]</td>
</tr>
<tr>
<td>Scaled Agile Framework (SAFe)</td>
<td>[BCT19]</td>
</tr>
<tr>
<td>Large-Scale Scrum (LeSS)</td>
<td>[UKD+19]</td>
</tr>
<tr>
<td>Agile Software Solution Framework</td>
<td>[QH08]</td>
</tr>
</tbody>
</table>

Table 10: Overview of the agile methodologies dealt with in the primary studies

In addition to considering individual or multiple agile methodologies, 31 primary studies are included in the results of this SLR, which do not find a dedicated agile methodology. In these cases, for example, agile practices are taken into account or examined.

4.1.2. RQ1: Success factors of agile software development

The discussion in this chapter is the basis for answering RQ1: Which success factors influence agile methodologies in software development?

When extracting the data (see section 3.5), we noticed that differentiation into different categories of success factors makes sense. This categorization ensures the clarity of the presentation of success factors and increases readability. Likewise, the classification allows a later focus on several similar success factors, for example, for the use of agile methodologies in practice. There are various categorizations for success factors of agile methodologies in the literature (e.g. [CC08], [DPL16], [MKK09]).

The categories presented by Chow and Cao in [CC08] are used for this SLR: organizational, people, process, technical, and project. The choice of categories, according to Chow and Cao, is justified by the fact that the authors present their classification based on existing literature in [CC08]. Furthermore, the categories (and success factors), according to Chow and Cao, are verified by other authors ([SND+13]), or at least referenced (including [CLL14], [HNM11], [RKK16], [SS12]).
In addition to differentiating the success factors into different categories, another aspect is relevant. During the data extraction, we found that the considered context of the respective primary study is different. The context differs concerning the use of agile procedural models and agile transition:

- **Agile transition (K1):**
  In the context of K1, we assumed that a different software development approach (e.g., phase-oriented) was used before the agile transition. The transition to the use of agile methodologies or practices in companies is thus understood. The focus here is not the successful application of agile methodologies; it is much more relevant to consider the success factors during the transition.

- **Use of agile methodologies (K2):**
  In the context of K2, factors are taken into account that influence the success of using agile methodologies.

Individual success factors can only be assigned to one (e.g., sharing knowledge on K1) or both contexts (e.g., management support for K1 and K2). If a success factor can be assigned to both contexts, different effects of the respective success factor can follow. It is, therefore, necessary to consider and explain success factors with the same name differently depending on the context. Because of this, in addition to the categorization of the success factors described above, the following context also takes into account the respective context K1 and K2.

In the course of data extraction, we identified 39 different success factors from 38 primary studies. As described above, we can categorize these success factors in terms of their context. We assigned a large part of the success factors (25) to the use of agile methodologies or characteristics in software development (context K2). We assigned the remaining success factors (14) to the agile transition (context K1). We assigned five of these 39 success factors to both contexts (K1 and K2). These are the following success factors:

- Value-based work (technical factor)
- Management support (organizational factor)
- Organizational culture (organizational factor)
- Training and coaching (process factor)
- Collaboration with the customer (process factor)
We described the success factors based on the respective context in the following two subchapters. The section begins with the explanations of the success factors in the context of agile transition (K1).

### 4.1.2.1. Success factors of agile transition (K1)

The context K1 comprises 14 success factors. We extracted these from seven different primary studies. It should be noted that some primary studies name several success factors, and others only list one factor. Table 11 below shows the allocation of the primary studies to the categorized success factors:

<table>
<thead>
<tr>
<th>Category</th>
<th>Success factor</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical factors</td>
<td>Tech1: Value-based work</td>
<td>[AFB+19], [DPL16], [JZ15]</td>
</tr>
<tr>
<td></td>
<td>Tech2: Use the right agile practices</td>
<td>[CBA15]</td>
</tr>
<tr>
<td>Process factors</td>
<td>Proc1: Training and Coaching</td>
<td>[DPL16], [UKD+19]</td>
</tr>
<tr>
<td></td>
<td>Proc2: Customer collaboration</td>
<td>[AFB+19]</td>
</tr>
<tr>
<td>Project factors</td>
<td>Proj1: Project complexity</td>
<td></td>
</tr>
<tr>
<td>People factors</td>
<td>Peop1: Experience and skill in development team</td>
<td>[CBA15]</td>
</tr>
<tr>
<td></td>
<td>Peop2: Group problems (group maturity) when assembling agile teams</td>
<td>[GTF17]</td>
</tr>
<tr>
<td></td>
<td>Peop3: Self-organized team/team autonomy</td>
<td>[DPL16]</td>
</tr>
<tr>
<td></td>
<td>Peop4: Knowledge sharing</td>
<td>[CM18]</td>
</tr>
<tr>
<td>Organizational</td>
<td>Orga1: Gradual, iterative, continuous transition to agile</td>
<td>[DPL16], [JZ15]</td>
</tr>
<tr>
<td>factors</td>
<td>Orga2: Management support</td>
<td>[BCT19], [DPL16]</td>
</tr>
<tr>
<td></td>
<td>Orga3: Organizational culture</td>
<td>[CBA15], [DPL16]</td>
</tr>
<tr>
<td></td>
<td>Orga4: Communication and transparency regarding the transition to an agile approach</td>
<td>[DPL16], [UKD+19]</td>
</tr>
<tr>
<td></td>
<td>Orga5: Enabling environment</td>
<td>[BCT19]</td>
</tr>
</tbody>
</table>

Table 11: Categorized success factors for agile transition

We used an abbreviation for the textual description of each success factor to simplify identification and assignment to the associated category. We created this abbreviation from the abbreviated name of the assigned category (the first four characters) and a number that increases by one with each further success factor. We also used this identifier format for the success factors of the context K2 (see Chapter 4.1.4).

The number of mentions does not differ significantly from one another in context K1. The success factor **Tech1 valued-based work** is mentioned in three primary studies; the factors **Proc1 training and coaching**, **Orga1 gradual, iterative, continuous transition**, **Orga2 support for management**, **Orga3 organizational culture**, and **Orga4 communication and**
transparency regarding the agile transition are mentioned in two primary studies. The remaining eight success factors are each described in one primary study (see Figure 6).

![Figure 6: Success factors of the agile transition](image)

**Tech1 Value-based work:**

We identified the success factor of value-based work in three primary studies (see Table 10). Value-based work means the strict prioritization of requirements, for example, based on business value or the like, which should lead to the iterative provision of components with high business values [AFB+19], [DPL16]. Javdani and Ziaei describe in [JZ15] an empirically-developed framework for the agile transition. In their publication, they describe value-based work as an element for a successful agile transition: "Considering business values and focusing on achieving more business values during the transition, are the most important factors which lead to successful transition." In addition to focusing on work based on business values, Javdani and Ziaei also point out other (agile) values that a company should define when an agile transition begins. Altaf et al. describe in their research summary in [AFB+19] that non-functional requirements must not be neglected in an agile transition. Concerning the timely and regular delivery of product increments, Dikert et al., in their SLR in [DPL16], indicate that some release activities have extended lead times, and planning and time problems can arise from this.
Tech2 Use the right agile practices:

The primary studies on the context of agile transition do not only consider methodologies in their entirety. Individual agile practices are also frequently examined—for example, Claps et al. in [CBA15] consider the transition of agile practice as continuous deployment. To be able to continue the agile transition, e.g., to introduce another practice (in this case, continuous deployment), the use of agile practices can be a success factor. Claps et al. have identified continuous integration of the agile practice as a success factor for the practice of continuous deployment. The publication of the research results by Claps et al. can be used here as an example that using the right agile practices is a success factor in an agile transition.

Proc1 Training and coaching

Knowledge transfer is of central importance when introducing a new approach. According to two primary studies, this must also be taken into account in the agile transition (see [DPL16], [UKD+19]). In [DPL16], Dikert et al. present the training of the workforce concerning the new agile approach as a success factor. They also state that the accompanying coaching during the transition (by the workforce themselves or external experts) is a success factor. Uludag et al. describe in [UKD+19] the importance of a real understanding of the new methodology. To ensure this, the authors recommend carrying out joint workshops in addition to training. In the two publications ([DPL16], [UKD+19]), the authors point out that the training can also ensure the transparency of the agile transition and its goals.

Proc2 Customer collaboration

Another success factor is described by Altaf et al. in their research summary in [AFB+19]. In addition to the requirement that the customer is available, the authors also describe customer involvement in the project as a success factor.

Proj1 Project complexity

In the category of project-based success factors, only one success factor in the literature is named in the context of the agile transition. The authors Altaf et al. in [AFB+19] state that the complexity of the project is an aspect that has to be considered in an agile transition. This success factor is related to the success factor Orga1 (gradual, iterative, continuous transition to agile), since these are, for example, pilot projects for agile transition, and the complexity of the project is essential when selecting these pilot projects.
Peop1 Experience and skill in the development team

According to Claps et al. in [CBA15], the experience of a team and the skills of each team member influence the success of the agile transition. Due to the focus of the authors' study, the success factor relates in particular to agile characteristics with technical properties (here: continuous deployment).

Peop2 Group problems (group maturity) when assembling agile teams

In [GTF17], Gren et al. investigated how group maturity affects the introduction and application of agile methodologies in software development. They find that the problems within groups (here, software development teams) represent a success factor when assembling agile teams. The authors point out that this assessment was primarily carried out by roles responsible for agile processes (e.g., Scrum Master, Agile Coach).

Peop3 Self-organized team/team autonomy

The self-organizing character of agile software development teams is often defined in the rules of agile methodologies (see Chapter 2.1.1). Dikert et al. cite self-organization as a success factor in an agile transition in their SLR in [DPL16]. The authors justify this, among other things, with the fact that self-organization is to be seen as a prerequisite for the team to assume responsibility for the development process. They also describe that self-organization promotes the success of an agile transition by creating the commitment to change and the motivation to use agile characteristics continuously.

Peop4 Knowledge sharing

According to Cram and Marabelli in [CM18], sharing knowledge is an important aspect of successful software development. In their publication, the authors compare the differences regarding the sharing of knowledge in agile, hybrid, and phase-oriented approaches. They consider several agile practices, such as the Sprint Retrospective (from Scrum) or the pair programming, as well as the collective code ownership (both from XP) and the associated advantages for sharing knowledge. Cram and Marabelli particularly emphasize regular verbal communication (e.g., through stand-up meetings) and non-verbal communication (among other things, via storyboards) as essential aspects of knowledge sharing.
**Organizational Gradient, Iterative, Continuous Transition to Agile**

For the agile transition, the selection and adaptation or individualization of agile characteristics and methodologies is necessary. Dikert et al. point out in their SLR in [DPL16] that the choice of agile practices or procedural models should be in line with the corporate model. The authors also describe that the individualization and adaptation of agile practices by the development teams have resulted in the agile transition being successful.

In addition to the selection and possible adaptation of agile characteristics, according to Javdani and Ziaei in [JZ15], the procedure for agile transition is of high relevance. The authors take this into account in the publication of their framework in [JZ15] as follows:

- **Iterative transition**
  In [JZ15], the authors describe various options for iterative transition. They name the possibility of a multi-stage process in which agile practices can be gradually introduced. They also point out that they use proven concepts such as Deming’s PDCA cycle (cf. [MN09]) to add agile characteristics iteratively, i.e., repetitively based on a process. According to Javdani and Ziaei, the iterative approach leads to a gentle transition. This results in fewer adverse effects and, accordingly, more time to be able to adjust the results of the transition (e.g., in the corporate culture).

- **Gradual transition**
  Javdani and Ziaei point out in [JZ15] that the iterative transition should also be gradual to minimize risks and challenges. This step-by-step introduction can take place, for example, in the context of non-critical projects, in order to focus on the problems of the transition during the implementation of the project. In their study, they also state that the gradual transition to an agile approach is particularly useful to facilitate the transition for critics. They justify this with the fact that many companies are not yet ready for the agile transition since the changes have a pervasive effect on many areas in the company. A quick and extensive transition can be overwhelming and lead to failure.

- **Continuous transition**
  In their study in [JZ15], the authors note that the combination, especially with the iterative transition, is essential. As stated above, the main focus of the iterative transition is on repeating the same (or similar) process steps. According to Javdani and Ziaei in [JZ15], continuous transition means that the process used is not interrupted. Here, however, the transition of individual agile characteristics and
not of a procedure model as a whole is to be understood. With each new agile practice to be introduced, the respective process starts over and should not be interrupted.

Javdani and Ziaei in [JZ15] and Dikert et al. in [DPL16] indicate that the agile transition does not necessarily have a result to be measured. The needs of individual teams vary much more, which requires a structured approach to the transition. Javdani and Ziaei write in [JZ15]: “... In this case, transitioning to Agile can be a non-stop process and can be re-started when a new practice needs to be adopted.”

**Orga2 Management support**

Berkani et al. describe the support of management as a success factor in their qualitative study in [BCT19] on an agile transition at a central bank. The authors point out the importance of supporting management, particularly in the agency-wide (company-wide) agile transition. Dikert et al. designate this success factor as "absolute necessity" in their SLR in [DPL16] and base this on the relevance of the factor in various studies. They justify the importance of the support of the management, among other things, with the impact on different framework conditions (such as the release plans for software applications in the company) and other success factors (such as Proz1 training and coaching). The authors also point out the far-reaching consequences of management decisions, which can help eliminate impediments, for example.

**Orga3 Organizational culture**

The cultural level (see Chapter 2.1.2) of the organizational culture (corporate culture) has to be considered in many ways during the agile transition. Dikert et al. state in their SLR in [DPL16] the importance of the ability to change the organizational culture. They justify this due to the frequent need to change the organizational culture, which other authors also describe (cf. [MKK09]). The authors assume that the agile transition has a direct impact on various aspects of the company (including bureaucracy and the inconvenience of change as such). Furthermore, they mentioned in [DPL16] that with an ingrown organizational culture during the agile transition, the vision of this change must be continuously pointed out. Berkani et al. also point to organizational culture as a success factor in [BCT19]. This is how they describe in their publication that a campaign for cultural awareness was prepared in the central bank under consideration for the start of the agency-wide (company-wide) agile transition.
**Orga4 Communication and transparency regarding the transition to an agile approach**

Dikert et al. state in their SLR in [DPL16] the importance of communication regarding the new approach. They describe that intensive communication with as many employees as possible is necessary for a successful agile transition; otherwise, the new, agile approach cannot establish itself. By way of communication, the authors suggest individual conversations with management, as well as digital forms of communication such as newsletters or online discussions to best explain the motivation or goals of an agile transition.

Another aspect that has to be considered in this success factor is the transparency of the agile transition. According to Uludag et al. in [UKD+19], this transparency often comes from clear communication. The authors describe transparency as a success factor and justify this in [UKD+19] as follows: “... higher transparency incentivizes FTs (Feature Teams) to deliver higher quality software.” Dikert et al. in [DPL16] describe the transparency of the change as a critical success factor. The reason for this is that by sharing experiences and communicating the status of the agile transition, the company and its workforce have developed in the same direction.

**Orga5 Enabling Environment**

In their qualitative study on the agile transition at a central bank in [BCT19], Berkani et al. discuss the positive effects of a favourable environment. They describe a pleasant environment in which the introduction of agile practices can spread across different teams. Likewise, a favourable environment is characterized by the fact that positive experiences can be gained through the experimental use of agile practices in projects.
4.1.2.2. **Success factors of using agile methodologies in software development (K2)**

We assigned 25 success factors to the context of using agile methodologies (K2) and extracted them from 27 primary studies. Here, as with K1, it should be noted that some success factors are named in different primary studies, and others have only been extracted from one primary study (see Table 12).

<table>
<thead>
<tr>
<th>Category</th>
<th>Success factor</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical factors</td>
<td>Tech1: Agile requirements engineering</td>
<td>[CC08], [Dru14]</td>
</tr>
<tr>
<td></td>
<td>Tech2: Delivery strategy of product increments</td>
<td>[CC08], [SSK20]</td>
</tr>
<tr>
<td></td>
<td>Tech3: Selecting and using the right agile practices</td>
<td>[BU14], [SD17], [SS12]</td>
</tr>
<tr>
<td></td>
<td>Tech4: Scope of documentation</td>
<td>[CC08], [SSK20]</td>
</tr>
<tr>
<td></td>
<td>Tech5: Agile practice: test first</td>
<td>[Dru14]</td>
</tr>
<tr>
<td></td>
<td>Tech6: Technical training</td>
<td>[CC08]</td>
</tr>
<tr>
<td></td>
<td>Tech7: Value-based work</td>
<td>[ABG17], [CC08], [Dru14], [SSK20], [TSE+15]</td>
</tr>
<tr>
<td>Process factors</td>
<td>Proc1: Agile culture, values and principles</td>
<td>[BSE+16], [BZT+14], [CC08], [GKS18], [HNM11], [MKK09], [OK18], [SSK20], [STE17], [TMO+20]</td>
</tr>
<tr>
<td></td>
<td>Proc2: Customer involvement and collaboration</td>
<td>[CC08], [SND+13]</td>
</tr>
<tr>
<td></td>
<td>Proc3: Process definition</td>
<td>[SSP+15]</td>
</tr>
<tr>
<td></td>
<td>Proc4: Visibility of the process</td>
<td>[MKK09], [SSP+15]</td>
</tr>
<tr>
<td></td>
<td>Proc5: Training and coaching</td>
<td>[CC08], [SSK20]</td>
</tr>
<tr>
<td>Project factors</td>
<td>Proj1: Character of the project</td>
<td>[SND+13]</td>
</tr>
<tr>
<td></td>
<td>Proj2: (Dynamic) Project plan</td>
<td>[SND+13]</td>
</tr>
<tr>
<td>People factors</td>
<td>Peop1: Experience and skill in the development team</td>
<td>[BZT+14], [CC08], [FIK+15], [GKS18], [MKK09], [SSK20], [TMO+20]</td>
</tr>
<tr>
<td></td>
<td>Peop 2: Communication</td>
<td>[BMP+19], [CC08], [LHC+19], [MKK09], [RRO+19], [SSK20]</td>
</tr>
<tr>
<td></td>
<td>Peop 3: Societal culture</td>
<td>[MKK09]</td>
</tr>
<tr>
<td></td>
<td>Peop 4: Teamwork quality</td>
<td>[GKS18], [LSD+16]</td>
</tr>
<tr>
<td></td>
<td>Peop 5: Team satisfaction</td>
<td>[CC08], [Dru14]</td>
</tr>
<tr>
<td></td>
<td>Orga1: Workplace environment</td>
<td>[BZT+14], [RKK16]</td>
</tr>
</tbody>
</table>
Table 12: Success factors for using agile methodologies

In the context of K2, the number of mentions of success factors in primary studies is distributed differently to K1 (see Figure 7). Few success factors show high mentions in primary studies: Proc2 Involvement and cooperation with the customer are mentioned in ten, Mens1 Experience and skill in a team in eight, Mens2 Communication and Orga3 Organizational culture in six and Tech7 Value-based work in five primary studies.

Figure 7: Success factors of using agile methodologies

**Tech1 Agile Requirements Engineering**

Drury-Grogan states in her study in [Dru14] the importance of agile requirements engineering. The author draws attention to the advantages of the fast response options to
changing requirements. Also, she describes various types of requirements documentation when using agile methodologies and emphasizes the variability. In their survey on the factors responsible for agile software development in [CC08], Chow and Cao also mention following an “agile-oriented requirement management process” as a success factor. According to the authors, this success factor affects the quality dimension.

**Tech2 Delivery strategy of product increments**

In their survey in [CC08], Chow and Cao describe the delivery strategy as the most critical success factor for agile software development. They relate the delivery strategy to the principles of the agile manifesto and indicate that the success factor and the first and third principles\(^3\) coincide in content. According to the authors, the factor affects the dimensions of “delivering on time,” costs (“delivering with estimated cost and effort”), and scope (“meeting all requirements and objectives”). It is the only success factor identified that affects three out of four dimensions. The authors, therefore, place the success factor in the first position of the critical success factors.

Sinha et al. also point out in [SSK20] the success factor of the continuous delivery of software. In their SLR, they describe the motivation for incremental and continuous delivery of software as follows: “The software is delivered as it is developed and the client can use and experience the new software. So using the software before full installation, helps the customer familiarize themselves with the software.” According to Sinha et al., the success factor is also to be classified as critical; after all, it has the second-highest relevance in their SLR.

**Tech3 Selecting und using the right agile practices**

According to Senapathi and Srinivasan in [SS12], the success factor for the selection and use of agile practices relates, among other things, to the continuous optimization of the development team’s approach. In their study [SS12], the authors describe that the in-depth use of agile practices is essential. This detailed use is described using the example of agile practices from Kanban, which, according to the authors, lead to positive and effective outcomes. Another aspect that the authors describe is the individualization of agile practices in different teams (also within a company). This individualization can arise for various reasons. The experience of the teams with an agile approach is mentioned here as an example. In their qualitative study, Senapathi and Drury-Grogan in [SD17] confirm the importance of this factor.

\(^3\) The first and third principles of the agile manifesto describe the early and continuous delivery of valuable (principle 1) and functioning (principle 3) software (cf. [BBv+19]).
In their study on the reuse of source codes in [BU14], Barzilay and Urquhart also point out the importance of selecting and using agile practices. The authors found that a barrier exists when it comes to changing the source code developed by other people. According to the authors, this barrier is reduced when the agile practice of collective code ownership from XP is used. The selection and use of agile practices must, therefore, be classified as relevant for the targeted further development of a software development team (e.g., concerning team skills).

Tech4 Scope of documentation

Sinha et al. in [SSK20] refer to the choice of the correct scope of documentation when using agile procedural models in global software development. When the authors explain the success factor, they refer to Uikey et al. in [USR11] and refer to the aspect of the need for documentation: “Agile development encourages the practitioners to write project documents which are necessary to achieve the project goals. ... Only necessary documentation is done in agile.” Chow and Cao also refer to the correct amount of documentation in [CC08]. They describe the correct scope of the documentation as an attribute of the success factor agile software engineering techniques. According to the authors, the success factor influences the success dimensions of quality and scope.

Tech5 Agile practice: Test First

In her study in [Dru14], Drury-Grogan examines the relationships between agile software development teams and success factors in project management. Regarding the success factors, the author refers to the golden triangle of project management (cf. [Wes03]) with the criteria quality, time, and budget. In her study, Drury-Grogan found that the quality criterion and many decisions made by agile software development teams, in particular, create a relationship. This is how she describes the importance of test first approaches and emphasizes the relationship to the quality criterion of the golden triangle.

Tech6 Technical training

In [CC08], Chow and Cao name the influence of technical training for the team (or individual team members) as an attribute of the team capability factor. The team capability success factor influences the time and cost dimensions of success. According to the authors, team capability and delivery strategy (see Tech3) are the most significant success factors for agile software development.

Tech7 Value-based work

The authors Alahyari et al. examine the aspects of value-based work in agile software development organizations in their study in [ABG17]. They describe the focus on values
as a critical success factor in agile software development. The authors were able to identify 134 value aspects of the study. They categorized these value aspects and had them prioritized by the study participants. According to the authors, the categories "Delivery process with respect to time," "Perceived quality," and "Cost (product, project)" have the three highest priorities in order. The authors also describe which activities are used to create values. It is striking here that various agile methodologies and characteristics (including test-driven development, development loops, or iterative ways of working or pair programming and scrum of scrums) are named. Some of these activities can be assigned to other success factors of agile software development.

An example of this is the connection to the delivery strategy (see success factor Tech3). In her study in [Dru14], Drury-Grogan also describes the effects of value-based work by agile software development teams as a factor for quality. She describes this in [Dru14] as follows: "Quality also refers to doing work that adds value for the client and avoiding work that does not add value."

Torrecilla-Salinas et al. present in [TSE+15] the positive effects of value-based work on the estimation, planning, and management of agile web projects. The authors explain that in addition to business value, technical aspects must also be taken into account and the prioritization of the product backlog is essential for the success of agile software development. Chow and Cao refer in [CC08] to the value-based work concerning the delivery strategy (see success factor Tech3) in which the essential features are realized and delivered first. A connection to value-based work can be seen here since the priority of the requirements to be developed is usually based on the assigned value. The requirements for the numerous iterations are selected based on the priority. Sinha et al. also address the aspect of priority of requirements in [SSK20] and present this as a success factor.

**Proc1 Agile culture, values, and principles**

In their study in [BJ19] on the use of lean models in software development, Bakke and Johansen describe the impact of the culture of continuous improvement on the success of the project. The authors describe this culture primarily at the level of the software development team but also deal with the influences on the corporate culture. They also explain that the culture of continuous improvement mostly has a positive effect on the adaptation (tailoring) of the methodology or processes used. The authors also point out that the introduction of this culture is associated with training and workshops for team members and takes time. For this, it makes sense to carry out this cultural change process step-by-step to make it sustainable.
Misra et al. deal with adhering to agile principles and values in their survey in [MKK09]. The authors describe the importance of referring to agile principles for project success. For example, they relate to the acceptance of changes during the iteration or the planning and control by the team, not by the management.

**Proc2 Customer involvement and collaboration**

The success factor Proc1 is the most frequently mentioned in the primary studies of this SLR. Sinha et al. state in their SLR on success factors of agile methodologies in global software development in [SSK20] that "regular customer feedback" is the most important critical success factor. The authors justify the importance of the frequency of the mentions in the literature. Misra et al. describe this success factor as important. In their survey in [MKK09], they describe three aspects that they summarize as a success factor “customer-oriented topics”: customer satisfaction, customer cooperation, and customer loyalty. According to the authors, the three aspects are directly related to each other. For example, good customer cooperation has an impact on customer satisfaction. They represent customer loyalty (related to a project) as the most important of the three aspects (followed by customer cooperation and customer satisfaction; in this order). Tam et al. also show in their survey in [TMO+20] that customer cooperation is a success factor for agile software development. In this context, they also point out that the social culture (see Mens4) and personal characteristics have an impact on customer involvement and cooperation.

Bermejo et al. explain in their survey on agile software development in Brazil in [BZT+14] the positive effects of agile procedural models on the improved interaction with the customer. The authors describe the impact of the success factor “communication with the customer” on project success. They substantiate this factor by the frequency of the interactions with the customer, useful documentation for the team, and the motivation of the customer. They refer to Lindvall et al. in [LBB+02] and confirm the statements about the effect of this factor on the success of agile software development. In their survey, the authors continue to describe that other success factors of agile software development (such as Mens1) have a positive effect on the customer’s cooperation. Gren et al. have examined in their survey in [GKS18] the relationship between individual skills and agile practices. The authors have found that customer access (e.g., the availability of the customer for feedback or clarification of open questions about requirements) affects the success of agile software development. Hoda et al. investigate in their study in [HNM11] different ways of involving customers in agile and self-organized software development teams. The authors describe the importance of customer involvement in agile software development and can demonstrate the effect of insufficient customer involvement on the
project result. Schön et al. describe in their SLR in [STE17] on agile requirements engineering various aspects regarding the integration and cooperation of the customer. For example, they explain the type of customer involvement (indirect or direct) as an important factor for the success of the project. The authors also refer to the quality of communication between the customer and the development team.

Bjarnason et al. investigated in [BSE+16] the effects of distances on software engineering, both spatial distances and other types resulting from different specialist competences. The authors describe how different distances can affect software engineering practices and how this can affect project success. They name various aspects regarding integration and cooperation with the customer, which are also assigned to agile software development. In addition to communicative challenges, e.g., through a different understanding of requirements due to various technical skills on the part of the product owner and the development team, the authors also describe the importance of involving the customer in software development projects with spatial distances.

Chow and Cao describe the success factor (in terms of their success dimension scope) as critical. However, they also point out in [CC08] that the factor only affects this dimension and that no influence on the other three dimensions (quality, time, and costs) could be demonstrated. Ochodek and Kopczyńska confirm the results of Chow and Cao in [OK18] in their survey on agile requirements engineering practices.

**Proc 3 Process definition**

Chow and Cao identified the project management process as a critical success factor for agile software development in their survey in [CC08]. They have examined whether agile project management processes are adhered to and how they affect the success of software development projects. Chow and Cao describe the success factor in [CC08] with the following attributes: "Following agile-oriented requirement management process," "Following agile-oriented configuration management process," "Good progress tracking mechanism," and "Honouring a regular working schedule." They also name the attribute of direct communication (see Mens2). According to the authors, the success factor affects the quality (of the product) dimension. Stankovic et al. have verified the results of Chow and Cao in IT companies of the former Yugoslavia in their survey in [SND+13]. The authors found that they could not demonstrate the effect of the success factor on quality, as determined by Chow and Cao in [CC08]. However, they were able to explain an impact on the success dimension of costs.
Proc4 Visibility of the process

Selleri et al. describe in their SLR in [SSP+15] the visibility of the process as an important factor for understanding the process. According to the authors, this is not only possible through communication, e.g., in the development team, but requires the visibility (and transparency) of the process. The success factor is directly related to the success factor Proc1 (training and coaching), because according to the authors, the visibility of the process makes it easier, for example, to familiarize new team members with the process and thus has an impact on the training. According to the authors, the visibility and the resulting improved understanding of the procedure also make it possible to identify ways of improving the process or to optimize customer involvement.

Proc5 Training and coaching

Misra et al. in [MKK09] relate the success factor training and coaching to continuous learning and training and coaching within the development team. The authors also point out that less technical training is often required when using certain agile practices (such as pair programming in XP). Agile procedural models provide some agile practices that ensure an implicit knowledge distribution in the team. Selleri et al. describe different aspects of the training in their SLR on the combination of CMMI and agile software development in [SSP+15]. The authors point out that new team members have to be trained for agile characteristics when in doubt. Also, with the combination of the concepts CMMI and agile software development, increased training effort can be expected.

Proc6 Self-organizing team

The self-organization of agile software development teams is defined in the regulations of many agile methodologies (see section 2.1.1). Two primary studies have identified self-organization as a success factor for agile software development. For example, in their study in [CC08], Chow and Cao cite self-organizing teamwork as an attribute of the success factor team environment. According to the authors, this factor affects quality. Sinha et al. describe in [SSK20] the self-organization of teams as a success factor for agile software development. They mention the self-organizing character, for example, who takes responsibility for their work. They also point out the possibility of working from home.

Proj1 Character of the project

As explained above in the discussion of the Proc4 success factor, Stankovic et al. in [SND+13] verified the results of Chow and Cao from [CC08]. This study demonstrates proven success factors that Chow and Cao could not sustain with their survey. One of these success factors is Proj1 - the non-life-critical project character. In [SND+13], the
authors describe the project's character as life-critical, for example, when it comes to air traffic control programs. Non-life-critical projects can still be business-critical. According to Stankovic et al., the project character has an impact on the success dimensions of time and costs. According to the authors, a significant influence of the project character can be demonstrated whether the project is completed in the planned time or not.

**Proj2 (Dynamic) Project plan**

Stankovic et al. demonstrate in their survey in [SND+13] the influence of a dynamic project plan on the success of agile software development. According to the authors, the factor has a positive effect on the success dimension “cost.” As with the success factor Proj1 project character, Chow and Cao were unable to demonstrate this factor in [CC08].

**Peop1 Experience and skill in the development team**

Bermejo et al. also describe in [BZT+14] the importance of the correct composition of the team members and, in particular, refer to communication and cooperation. Sinha et al. also highlight the importance of existing skills in agile teams in their SLR in [SSK20] in the context of global software development. They justify this with the frequent mention of the success factor in the literature and describe the effects on the project success and the use of agile procedural models in the case of non-existent but necessary social and technical skills.

Chow and Cao describe **team capability** as one of the three “top critical success factors” in [CC08]. The high level of competence and experience of the team members is assigned to the success factor. In their study, the authors point out the importance of the right skills. In their study, Chow and Cao demonstrated that the success factor **team capability** affects the success dimensions of time and costs. Tam et al. can also prove in the survey in [TMO+20] the team skills and, in this context, the aspects of the experience and skills of the team members as a success factor for agile software development. Senapathi and Srinivasan also describe in their study in [SS12] the importance of experience and the technical skills of team members for the effective use of agile practices. At this point, however, it should be noted that in their qualitative study, Senapathi and Drury-Grogan in [SD17] could not prove the importance of the technical skills of the team members. Misra et al., who also examined the success factor in their study in [MKK09], were unable to demonstrate any significant connection between the competencies of the team members and the successful application of agile procedure models.

Gren et al. describe in their study in [GKS18] the effects of non-technical skills on the success of the project. For example, they refer to communication, coordination, and team structure. However, the authors point out that the effect of non-technical skills is of little
relevance to the use of agile practices. Fagerholm et al. in [FIK+15] describe social skills as an important factor for the performance of a team. This is not only to be considered when carrying out software development projects but is also relevant, according to the authors, when selecting new team members.

**Peop2 Communication**

Due to definitions in the agile manifesto such as the prioritization of interactions (see section 2.1.1), the mention of communication as a success factor for agile software development is not surprising. This can be seen in the application of various agile practices in which communicative skills are highly relevant. In their multiple case study in [RRO+19], Ram et al. state that regular feedback in software development teams contributes to success in the operationalization of metrics. The authors describe the introduction of regular feedback meetings that are in line with the respective iterations. They explain a connection between these feedback meetings and the constant learning growth of the teams (and the company). In the context of global software development in [SSK20], Sinha et al. present the effectiveness of communication as a success factor. The effectiveness of communication channels is particularly relevant for different development locations. The authors Borrego et al. also point out in [BMP+19] the importance of communication in agile global software development.

Concerning the connection between communication and the success of agile software development, it should be mentioned that the authors Misra et al. in [MKK09] also attach importance to this success factor when describing the success factor *social culture* (see success factor Peop4). The authors also point out that, for example, the use of effective communication methods or regular feedback between team members can have an impact on success. Chow and Cao name another aspect of communication in [CC08]: Direct "*face to face*" communication in daily meetings. The authors list this aspect as an attribute of the "*project management process*" success factor demonstrated in their study.

Liu et al. describe in their survey in [LHC+19] the effects of using sprint planning and feedback in computer game development. The authors establish a relationship between the feedback from specific roles (such as testers) and the successful application of sprint planning. According to the authors, quick feedback has a positive effect on the projects and is therefore listed as a success factor in this SLR.
The authors Misra et al. describe in their study in [MKK09] that the influence of a social culture can be seen as a success factor when using agile software development. They see a connection between precise definitions of values of the agile manifesto (see section 2.1.1), such as the prioritization of individuals and interactions and the socio-cultural influence on a company (for example, through the communication skills of the employees). Misra et al. justify this in [MKK09] as follows: "... personal characteristics are important success factors, as is the societal culture, because the personal characteristics of the people are greatly influenced by their societal culture and vice versa".

Lindsjørn et al. investigate in their survey in [LSD+16] how the teamwork quality of agile software development teams affects project success. According to the authors, various models for the relationship between teamwork quality and team performance are described in the literature (e.g. [HWG04], [SSB05]). They also note that for development teams that use agile methodologies, various effects on aspects of teamwork are described in the literature (including [CC08], [PHS+08]). They discovered that the quality of teamwork is essential in (agile) software development. According to the authors, the quality of teamwork influences the performance of a development team and product quality. The authors note that the connection between the performance of a development team and the quality of teamwork is given higher weight by team members than, for example, by managers. The authors Gren et al. in [GKS18] also establish a connection between "teamwork skills" and the optimized planning of iterations.

Drury-Grogan describes team satisfaction in [Dru14] as a success factor. She describes the proven effects of various agile characteristics, such as agile practices (including the retrospective, collective code ownership) on team satisfaction. Chow and Cao indirectly refer to team member satisfaction in [CC08]. After all, they describe team motivation as an attribute of the critical success factor team capability. The basis for motivation in teams corresponds to the level of satisfaction among team members (cf. [TM07]).

In [RKK16], Rola et al. present a model that describes the configuration of a workplace environment that corresponds to the needs of the agile approach Scrum. The model is based on the cell structure of the honeycomb form and is composed of cells of different types. The cell types are defined as follows: "Conference cell," "Social/kitchen cell," "Chill out cell," "Development team cell," and "Product owner cell." Based on these cell types,
the authors combine the ideal workplace environment of Scrum Teams and validate the model in a case study. According to the authors, the optimized workplace environment with the help of the model has an impact on the successful use of Scrum and, thus, also on the success of the project.

Bermejo et al. describe in [BZT+14] the workplace environment as an influencing factor for the interaction skills between members of software development teams. The workplace environment is particularly relevant for agile software development, which focuses, among other things, on team interactions. In addition to the possibilities for communicating with the team (for example, no open-plan office with other agile teams), this also affects the organizational ability of the team (e.g., through the use of Kanban boards).

**Orga2 Effective learning environment**

Sinha et al. have identified the effective learning environment as a success factor for agile methodologies in global software development in their SLR in [SSK20]. The authors justify this by the fact that agile methodologies are continually developing and that only the teams are responsible for processing the tasks. From this, they conclude that an effective learning environment is important for the team to be able to meet these requirements. An effective learning environment enables the team to learn new things and to develop continuously.

**Orga3 Organizational culture**

The authors Misra et al. named in their survey in [MKK09] the organizational culture as a critical success factor for agile software development. The authors refer to various findings by Lindvall and Basili [LBB+02] and Abrahamsson et al. [ASR+02]. They describe the organizational culture as a success factor for the use of agile procedures in software development and present evidence in their publications. The results of the survey confirm these findings: "The organization should have the right culture, which involves supporting rapid communication, dynamicity in requirements changes, trusting people, and obtaining fast feedback from customers." In their quantitative study on agile software development in Brazilian companies in [BZT+14], Bermejo et al. also refer to the importance of organizational culture. Referring to Lindvall and Basili in [LBB+02], the authors also describe the openness of corporate culture for negotiations between employees as an important factor for the use of agile methodologies. Furthermore, they explain the effects of organizational culture on flexibility in the event of changes (e.g., the requirements) or the personal responsibility of the development teams. The results of their quantitative study in [BZT+14] confirm the findings of Lindvall and Basili from [LBB+02] and Misra et al. from [MKK09].
Alzoubi et al. define in [AGA16] the organizational culture concerning Hofstede et al. from [HHM10] as "the values, attitudes, and behaviours that represent an organization's working environment, vision, and objectives." The authors Alzubi et al. describe the organizational culture as a success factor because it enables fast communication. They also point out the supportive effect of an agile approach in software development on the trust between the stakeholders. In their SLR in [RP20], Raharjo and Purwandari describe organizational culture as a challenge when using agile methodologies in software development. They emphasize the support of management, the management of communications, and stakeholders as opportunities to meet these challenges.

Bakke and Johansen describe in [BJ19] the importance of organizational culture, for example, in the sense of the ability to change and learn for the success of lean approaches in software development. They refer to the context of the introduction of a culture of continuous improvement (see success factor Proc3 Agile culture, values, and principles).

Khan et al. point out in [KSK+19] that organizational culture has an impact on the coordination of team members in the context of global software development.

**Orga4 Team size**

Sinha et al. list team size as a success factor in [SSK20]. They refer to the empirical study by Shameem et al. in [SKK+18]. Shameem et al. point out that team size, through greater transparency and coordination, has a direct impact on trust in the team.

**Orga5 Management support**

Senapathi and Srinivasan point out in [SS12] that this success factor is particularly crucial when it comes to establishing the use of agile methodologies throughout the company and thus creating an agile organization. With this success factor, it can therefore be assumed that a few teams are already working with an agile approach and that management support is a success factor for the further spread in the company. The discriminatory power in the context of agile transition (K1) represents the existing use of the agile approach in development teams. Senapathi and Drury-Grogan confirm the positive impact of management support in their qualitative study in [SD17].

In contrast to Senapathi and Srinivasan, as well as Senepathi and Drury-Grogan, Chow and Cao note in [CC08] that, although they also examined the success factor in their research, they could not prove it.
4.1.3. RQ2: Cultural influences on agile software development

The second research question deals with cultural influences on agile software development: **Which cultural factors are relevant when using agile methodologies in software development?**

As described in Chapter 3, the cultural levels described in Chapter 2.1.2 must be taken into account when answering this SLR question. We used the cultural levels to categorize the cultural influence in order to structure the cultural impact.

The following table shows the assignment of the primary studies and cultural influence:

<table>
<thead>
<tr>
<th>Cultural level</th>
<th>Description</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>National culture</td>
<td>Cultural influences on agile practices</td>
<td>[AVH17]</td>
</tr>
<tr>
<td></td>
<td>Collaboration in virtual teams</td>
<td>[GB17]</td>
</tr>
<tr>
<td></td>
<td>Influence on the areas of software engineering (e.g., requirements engineering)</td>
<td>[GBS11]</td>
</tr>
<tr>
<td></td>
<td>Influence on distributed software development</td>
<td>[HT07]</td>
</tr>
<tr>
<td></td>
<td>Use of ERP system</td>
<td>[HG11]</td>
</tr>
<tr>
<td></td>
<td>End user involvement in enterprise systems</td>
<td>[Hwa12]</td>
</tr>
<tr>
<td></td>
<td>Scrum in the context of global software development</td>
<td>[LKT17]</td>
</tr>
<tr>
<td></td>
<td>Decision making</td>
<td>[MD07]</td>
</tr>
<tr>
<td></td>
<td>Influence on communication in distributed teams</td>
<td>[SSA+11]</td>
</tr>
<tr>
<td></td>
<td>Influence on teamwork/project success in global software development</td>
<td>[Su15]</td>
</tr>
<tr>
<td></td>
<td>Influence on the motivation of software engineering teams</td>
<td>[VBC+14]</td>
</tr>
<tr>
<td>Organizational culture</td>
<td>Usability in agile software development</td>
<td>[CSR+19]</td>
</tr>
<tr>
<td></td>
<td>Influence on the agile transition</td>
<td>[JZ16]</td>
</tr>
<tr>
<td></td>
<td>The success of lean/agile methodologies in large-scale environments</td>
<td>[PFG13]</td>
</tr>
<tr>
<td></td>
<td>Influences on the process optimization when using agile and hybrid methodologies</td>
<td>[PT18]</td>
</tr>
<tr>
<td></td>
<td>Taking cultural influences into account when introducing, applying and improving agile methodologies</td>
<td>[QH08]</td>
</tr>
<tr>
<td></td>
<td>Influences on global software development</td>
<td>[Gal09]</td>
</tr>
<tr>
<td></td>
<td>Test Governance Framework (Outsourcing IS development)</td>
<td>[DB15]</td>
</tr>
<tr>
<td>Subculture</td>
<td>Integration of UI experts in agile software development teams</td>
<td>[LGC17]</td>
</tr>
<tr>
<td></td>
<td>Agile culture is characterized by the customer’s perception of the leadership style of the team, the reaction to stress and customer involvement</td>
<td>[JWA14]</td>
</tr>
</tbody>
</table>

Table 13: Assignment of the primary studies to cultural influences per cultural level

Twenty primary studies are used to answer the second SLR question. More than half of these 20 primary studies deal with influences of the cultural level national culture (11). Another seven studies deal with the impacts of the cultural level of organizational culture.
We discuss the influences of the cultural level of the subculture in two studies. The impact of the cultural levels of the regional culture, the corporate culture, and the departmental culture are not addressed in the primary studies.

The answer to the second SLR question is structured based on the cultural levels:

**Influences of the cultural level of national culture**

Different models are described in the literature for the comparability of different cultures (see section 2.1.2). Some primary studies describe different or individual models but do not use them because they are, for example, research summaries. In the primary studies that use a model, the Hofstede model of cultural dimensions is most often used. Only one primary study uses a different model. The following table shows the assignment of the primary studies to the models:

<table>
<thead>
<tr>
<th>Cultural level</th>
<th>Cultural model used</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>National culture</td>
<td>Hofstede's cultural dimensions</td>
<td>[AVH17], [GBS11], [HG11], [Hwa12], [MD07]</td>
</tr>
<tr>
<td>National culture</td>
<td>No model used</td>
<td>[GB17], [LKT17], [SSA+11], [Su15], [VBC+14]</td>
</tr>
<tr>
<td>National culture</td>
<td>Adapted “situating culture” approach according to Weisinger and Salipante</td>
<td>[HT07]</td>
</tr>
</tbody>
</table>

Table 14: Cultural models used at the national culture level

The qualitative study by Ayed et al. in [AVH17] has the highest relevance for the present work. The authors examine the cultural influence on agile practices in Europe (Belgium) and Asia (Malaysia and Singapore). They use the comparative cultural model of the cultural dimensions, according to Hofstede, for their investigation. With the help of the study, the authors were able to demonstrate national cultural influences for certain agile practices for five cultural dimensions: power distance (PDI), individualism (IDV), uncertainty avoidance (UAI), long-term orientation (LTO) and indulgence (IVR). The authors found no effects on the masculinity dimension (MAS). Based on the findings of the qualitative study, 16 hypotheses were developed that explain the impact of national cultures on agile practices. For example, the authors assume that a high power distance (PDI) has a negative impact on process improvement, customer engagement, or the transparency of teamwork. They also describe the assumption that a high level of long-term orientation (LTO) has a positive effect on the commitment to apply new practices. These hypotheses are defined as the basis for future research projects.
Various studies dealt with the influence of national culture on aspects of global software development (cf. [HT07], [LKT17], [SSA+11], [Su15]). Lous et al. identified in their SLR in [LKT17] factors that restrict the use of Scrum in globally distributed projects. The authors describe the effects of differences in national culture as a critical factor. These affect communication in particular. With agile software development, this effect can have a significant impact on project success. In their SLR, the authors also deal with the question of how to deal with these challenges. Concerning the national cultural aspects, they describe, among other things, the possibilities of establishing rituals in communication, increasing the frequency of communication, or the use of various communication models. Serce et al., in their study in [SSA+11], also refer to the effects of differences between different national cultures on communication. In the study, different teams made up of students from different countries were considered concerning communicative behaviour. The authors identified differences in communication content. For example, the Turkish members of a student development team focused their communication on planning content, while other team members (from Panama) documented a more significant amount of text. In their study in [HT07], Huang and Traut deal with the effects of different national cultures on distributed software development. In addition to the potential language barrier, they also describe the influences of cultural factors in digital communication in distributed teams. The authors point out the importance of an equal understanding, for example, of the technologies used to be able to make the necessary decisions as a team when realizing projects. They also describe the potential effects of cultural differences. They have found that this can lead to severe misunderstandings and, as a result, frustration. Huang and Traut therefore emphasize the importance of intercultural training to avoid such situations. Großer and Baumöl examine the cultural aspects of virtual collaboration in their study in [GB17]. Due to the teams working in different locations, the authors often face challenges due to cultural differences. The behaviour in virtual communication (e.g., expressing criticism) can vary depending on the culture. They call these challenges psychological distance. To be able to meet these challenges, it is essential to make them known to the team and to support the team in overcoming them. The authors mention, among other things, the sensitization of team members to cultural differences.

Ghinea et al. describe in their survey in [GBS11] the effects of national cultural differences in the example of the use of procedural models in software development in India and Norway. They also discuss the importance of usability in software development projects. About usability, the authors examine the consideration of requirements and quality assurance. Based on the survey, they found that the use of procedural models is different in the two countries. In India, for example, agile methodologies are used much more
frequently. The authors did not find any relation to cultural effects. Likewise, there is a higher acceptance of formal procedural models in India than in Norway. The authors attribute this aspect to the higher power distance and uncertainty avoidance in India.

In their survey in [HG11], Hwang and Grant deal with the behavioral aspects of users when introducing ERP systems. The authors investigate how behaviour can be justified based on cultural influences. For the comparability of the culture, the authors refer to the cultural dimensions according to Hofstede. According to the authors, the dimensions of power distance and avoidance of uncertainty affect the acceptance of ERP systems. Because of this, the early involvement of the users of these applications makes sense as part of the desired development. Hwang and Grant describe that agile methodologies can support this since regular user involvement is possible, and users can test product increments early or even apply them. In another study, Hwang in [Hwa12] also made a comparison between Japanese and American culture. Here Hwang examined to what extent the intrinsic motivation and innovative personal power affect the introduction of ERP systems depending on the culture. With the help of the study, he demonstrated that effects due to cultural differences in national culture have to be taken into account when introducing complex IT systems. In the specific example, the intrinsic motivation of users in the USA has to be taken into account, but in Japan, this aspect is not relevant. On the other hand, personal innovation in Japan has to be taken into account. According to Hwang, these effects can be countered, for example, with training courses to promote intrinsic motivation in the USA and the innovative strength of the workforce in Japan, thus increasing the likelihood of a successful ERP implementation.

Martinsons and Davison examine in their study in [MD07] how national culture affects decision-making in management. The study takes into account the national cultures of the United States, Japan and China. The authors deal with the question of how software applications can support decision-making and whether the use of such applications varies in the respective countries (and cultures). They find that decision-making varies by culture. For example, decision-making in Japan is shaped by existing business relationships. In the USA, decision-making is motivated, for example, by the fact that one's own advantage and high importance are taken. In their view, it is therefore necessary that software applications used in decision making take these differences into account.

The motivation of software development teams based on the survey in [VBC+14] by Verner et al. is also influenced by national culture. The authors describe this relationship based on the success of the project. While the teams in the USA, Australia, and Chile were motivated by the success of the project, this could not be demonstrated for Vietnamese teams. There is also a significant distinction in the question regarding external
motivation factors. The authors were only able to motivate the Vietnamese team through external factors such as realistic expectations from the customer. The authors also note that the motivation of the teams is not culture-dependent in all aspects. For example, good teamwork, good communication by the project manager, or the right work environment are equally motivating factors for all teams.

In his qualitative study in [Su15], Su describes various ways of dealing with the cultural influences of different national cultures. He presents a process for cultural sensory perception which is intended to help the workforce develop a dynamic view of culture and its various facets. Besides, the employees should be trained to build up knowledge in the subject area of culture. Furthermore, Su describes the possibility of further advancement of the development of the workforce through incentive systems and personnel management measures (e.g., job rotation). In his view, it is important to make the cultural differences known in companies and to learn from project experiences to be better able to handle the influence of cultural differences.

**Influences of the cultural level organizational culture**

No culture comparative model is used in the primary studies to describe the influences of different organizational cultures (see Table 14). A study uses the culture-describing model from Schein ([DB15]) for a better understanding.

<table>
<thead>
<tr>
<th>Cultural level</th>
<th>Cultural model used</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational culture</td>
<td>No model used</td>
<td>[CSR+19], [Gal09], [JZ16], [PFG13], [PT18], [QH08]</td>
</tr>
<tr>
<td></td>
<td>Schein’s iceberg model</td>
<td>[DB15]</td>
</tr>
</tbody>
</table>

Table 15: Cultural models used at the organizational culture level

Javdani and Ziaei describe organizational culture in [JZ16] as the central challenge for the agile transition. They justify this with the effects of this culture level on various aspects when introducing agile procedural models. Their research shows that, for example, the selection and introduction of agile practices can be problematic if many team members put their own advantages before those of the team. The authors also describe fundamental cultural differences between organizational cultures that focus on control and skills and agile procedural models such as Scrum that focus on cooperation. Pernstål et al. state in their research summary in [PFG13] that organizational culture is an important influencing factor in the use of agile procedural models in software development in large-
scale environments. The organizational culture should support the introduction and application of agile procedural models and thus ensure, for example, a similar understanding of agile values, in the sense of continuous improvement. The authors also emphasize the importance of communication and collaboration. In their SLR in [PT18], Papadakis and Tsironis also emphasize the importance of an organizational culture that allows changes and thus enables continuous optimization. In her study in [Gal09], Galinac also describes organizational culture as an important influencing factor in the area of improving procedural models in software development (SPI). Doležel and Buchalcevová present their Test Governance Framework in [DB15]. In the multiple case study, the authors deal with the influences of organizational culture on changes in the organization of software development. They describe various effects that organizational cultures can have on software development projects and the understanding of software quality.

In their study in [QH08], Qumer and Henderson-Sellers present a framework for evaluating, introducing, and improving agile methodologies. The authors point out that the type of organizational culture is highly relevant. They differentiate between collaborative, cooperative, or non-collaborative. They also describe the implementation of cultural change towards an agile culture and way of thinking as the most important challenge for the successful application of agile methodologies in software development.

Curcio et al. investigate the possibilities and challenges of integrating usability into agile software development in their study in [CSR+19]. In the research summary, the authors point out that organizational culture can be a challenge when using agile methodologies. According to the authors, organizational culture can affect the distribution of resources or the type and scope of documentation.

### Influences of the cultural level subculture

Two primary studies deal with the influences of the subculture on agile software development deals. No culture comparative model is used in either study.

<table>
<thead>
<tr>
<th>Cultural level</th>
<th>Cultural model used</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subculture</td>
<td>No model used</td>
<td>[JWA14], [LGC17]</td>
</tr>
</tbody>
</table>

Table 16: Cultural models used at the subculture level

The authors Larusdottir et al. treat in their study in [LGC17] the integration and optimization of User-Centered Software Design (UCSD) in agile software development. In their study, they describe the connections between the two approaches and also deal with the cultural peculiarities of UCSD and agile software development. Their results show that these cultural differences are significant: "So our research shows that UX professionals..."
rarely fit in the team culture." They argue that a strong team culture in Scrum in which USCD experts are integrated into the team seems problematic. The authors recommend intensifying communication with users to make the importance of the USCD aspects transparent and thus simplify the integration of the UCSD experts into the Scrum team.

In their case study in [JWA14] on the evaluation of surveys that were developed to evaluate agility in practice, Jalali et al. point to an agile subculture. The authors describe this agile subculture as the customer's perception of the team's leadership style, the reaction to stress, and customer involvement or lack of involvement. From the authors' point of view, the agile subculture is a characteristic with which the degree of agility can be measured.

### 4.1.4. RQ3: Adaptation of agile methodologies in software development

The basis for the third SLR question is the answer to the previous SLR question on cultural influences on agile methodologies (see Chapter 4.1.3). As a reminder, RQ3 is rephrased: **Are there adjustments to agile methodologies based on the knowledge that cultural differences influence these models?**

There are only two primary studies to answer this question:

<table>
<thead>
<tr>
<th>Description of the primary study</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLR on agile practices in global software development</td>
<td>[VdP+18]</td>
</tr>
<tr>
<td>Introduction of an agile approach (Lean)</td>
<td>[BJ19]</td>
</tr>
</tbody>
</table>

Table 17: Paper describes the adaptations of agile methodologies

The SLR from Vallon et al. in [VdP+18] is based on the study results of Jalali and Wohlin in [JW10], [JW12] and was designed to continue the SLR and generate current results. The SLR addresses agile practices in the context of global software development. The authors have identified some studies that address the adaptation of agile methodologies or characteristics. They indicate the cultural influences on agile methodologies, but do not describe any concrete adaptations of agile approaches based on cultural aspects. However, it is also pointed out that these adaptations are not described in sufficient detail to draw general conclusions from them.

Both Vallon et al. in [VdP+18], as well as Bakke and Johansen in [BJ19], describe the need for agile methodologies to be adapted to the respective context such as the company. The authors justify this by the fact that the respective peculiarities of companies
(such as culture) or the organizational context (such as distributed teams in global software development) are not taken into account in the rules of agile approaches.

A statement on adaptations based on cultural influences is not made in the primary studies. Likewise, no specific adaptations are described. The third SLR question, therefore, only offers a clue that the cultural peculiarities when using agile methodologies must be taken into account when adapting.

5. Limitation of the study

The biggest challenge in systematic literature searches is to ensure that the results of the relevant primary studies are complete. For the present SLR, the searches in the digital libraries ACM and ScienceDirect were carried out to take into account different publishers. For both libraries, we performed the searches with the same filter settings to identify as many relevant primary studies as possible (see section 3.1). An additional difficulty arises from the search functionality of digital libraries. The search for related primary studies at ACM and ScienceDirect is linked to certain requirements. Restrictions regarding the composition and length of search terms must be taken into account (cf. [BKB+07], [KFD17]). The guidelines of the digital libraries were taken into account when developing the search terms (see chapters 3.1 and 3.3). For the present SLR, it must also be taken into account that a renewed literature search was necessary due to the long period (see section 3.3). The fact that the search function at ScienceDirect had changed fundamentally, and therefore the search term had to be adjusted, is also a limitation. We refined the used search terms and filter settings using pilot searches (see section 3.1). We compared the results of these pilot searches with selected primary studies, which we believe must be included in the result set. With the help of this procedure, we were able to ensure our defined degree of completeness and guarantee quality in this context.

Another challenge when creating an SLR is to ensure the quality of the extraction of the primary studies during the selection process. For the present SLR, we defined inclusion and exclusion criteria that were used for each process step in the selection (see section 3.2). The selection process was carried out by the first author. The other authors have checked the result sets of the respective steps in the selection process. This is intended to avoid the potential bias of the first author and the associated distortions in the selection of the result sets. However, the systematic approach to this literature search cannot guarantee that other researchers would have included or excluded the same studies.
6. Conclusion

We carried out this SLR to verify which cultural influences on agile software development are described in the existing literature. The results of this SLR include:

- **Success factors of agile software development (RQ1):**
  The majority of the primary studies of this SLR (38; 64.4%) deal with success factors of agile methodologies and characteristics. These success factors are described in two parts based on the respective context of the primary study. The context is either the use or introduction of agile methodologies. For the context K1, the transition of agile procedural models, we identified and described 14 success factors from seven primary studies. The context K2, the use of agile procedural models, has been assigned and explained 25 different success factors in 27 primary studies.

The 39 success factors identified are categorized based on Chow and Cao from [CC08] as follows: technical factors, procedural factors, project-based factors, human factors, and organizational factors.

The success factors with the most frequent mentions in the primary studies are:

- **Context C1 – Agile Transition:**
  - Value-based work (technical factor): mentioned 3 times in the primary studies
  - Training and coaching (process factor): mentioned 2 times in the primary studies
  - Organizational culture (organizational factor): mentioned 2 times in the primary studies
  - Management support (organizational factor): mentioned 2 times in the primary studies
  - Communication and transparency regarding agile transition (organizational factor): mentioned 2 times in the primary studies
  - Gradual, iterative and continuous agile transition (organizational factor): mentioned 2 times in the primary studies

- **Context C2 – Using agile methodologies:**
  - Value-based work (technical factor): mentioned 5 times in the primary studies
  - Customer involvement and collaboration (processual factors): mentioned 10 times in the primary studies
- Experience and skill in the development team (people factor): mentioned 8 times in the primary studies
- Communication (people factor): mentioned 6 times in the primary studies
- Organizational culture (organizational factor): mentioned 6 times in the primary studies

Concerning the number of mentions of individual success factors in the primary studies, some unexpected results can be reported. It is therefore surprising that with K1, i.e., the agile transition, support from management with only two mentions is rarely identified as a success factor. After all, this aspect is often mentioned as a success factor in the literature and accepted as a widespread common assumption. Likewise, the focus on value-based work and the organizational culture has not yet received the necessary attention. The factors count as relevant in both contexts due to the number of their mentions and require more significant consideration. In particular, the description of the organizational culture as a success factor in the use of agile methodologies and their transition can be classified as highly relevant for this work. As the SLR shows, the cultural influence in the literature has so far not been sufficiently investigated and described. The influences and effects of a cultural nature on agile software development, therefore, require further research.

At this point, certain success factors should also be mentioned which have been classified as critical for decades. These include, in particular, the involvement and cooperation of the customer (which had the highest number of mentions of all success factors). The importance of this factor was already emphasized by Royce when presenting the waterfall model in [Roy70]. The importance of communication and team skills are not particularly surprising when using agile methodologies and characteristics. The importance of the social and professional skills of individual team members and the composition of software development teams have been mentioned in the literature for years (cf. [ACC12a], [CA18], [HP06], [MRF15], [Saw04]).

- **Cultural influences on agile software development approaches (RQ2):**
  For the answer to the second SLR question and the description of cultural influences on agile methodologies, 20 primary studies were taken into account. This
corresponds to approximately a third of the result set. Most of these primary studies deal with the influences of national culture on agile software development (11; 55%). Only two cross-cultural models are used in these primary studies. The most frequently used model is the cultural dimensions model according to Hofstede. This is used in five primary studies. Another model is used by a primary study with the situating culture approach, according to Weisinger. No model is used in the remaining five studies. The same applies to the other cultural levels examined, namely the organizational culture and the subculture. No culture comparative models are used for either of these cultural levels.

The cultural influence described in the primary studies often refers to the use of agile procedural models in global (and distributed) software development. This is not surprising since, in this environment, there are often cultural differences among the members of the software development team (for example, due to the different locations of the teams), and corresponding influences are noticeable. Other aspects where cultural influences are described are the team composition and work, but also the use of certain agile practices, such as pair programming. The influence of organizational and subcultures is rarely specified. For example, there is no discussion of how hierarchical cultures compare to cooperative cultures when using agile methodologies.

• Adaptation of agile methodologies (RQ3):

Based on the cultural influences described in the primary studies, this third SLR question aimed to identify any adaptations of agile characteristics and methodologies. These adaptations ultimately represent possibilities of how to react to specific influences from different cultural levels to successfully carry out agile software development based on the factors described in SLR question 1. Only two primary studies were classified as relevant for answering the third SLR question. It should be noted that one primary study was also assigned to the first SLR question.

The primary studies deal with the adaptations of agile procedural models and characteristics and, among other things, reduce the need for the organizational culture. The presentation of the adjustments and the reasoning based on certain cultural influences are not specific in the primary studies but rather general.
Research output in the context of agile software development has been increasing steadily over the past 13 years. The context considered in the studies is mostly either the application or introduction of agile methodologies and practices.

Scrum is the procedural model that is most often examined in the primary studies of this SLR. XP and Kanban should also be mentioned as relevant procedural models since they, too, are often the subject of studies in the primary studies.
References


<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Journal</th>
<th>Year</th>
<th>Volume</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BKB+07]</td>
<td>Brereton, P. et al.</td>
<td>Lessons from applying the systematic literature review process within the software engineering domain.</td>
<td>Journal of Systems and Software</td>
<td>2007</td>
<td>80</td>
<td>571–583</td>
</tr>
</tbody>
</table>


