

Towards a Taxonomy of Agile Methods: The Tree of Agile Elements

Michael Neumann

Faculty IV, Dpt. of Business Informatics

Hochschule Hannover - University of Applied Sciences and Arts

Hannover, Germany

michael.neumann@hs-hannover.de

Abstract—Agile methods require constant optimization of one’s approach and leading to the adaptation of agile practices. These practices are also adapted when introducing them to companies and their software development teams due to organizational constraints. As a consequence of the widespread use of agile methods, we notice a high variety of their elements: Practices, roles, and artifacts. This multitude of agile practices, artifacts, and roles results in an unsystematic mixture. It leads to several questions: When is a practice a practice, and when is it a method or technique? This paper presents the tree of agile elements, a taxonomy of agile methods, based on the literature and guidelines of widely used agile methods. We describe a taxonomy of agile methods using terms and concepts of software engineering, in particular software process models. We aim to enable agile elements to be delimited, which should help companies, agile teams, and the research community gain a basic understanding of the interrelationships and dependencies of individual components of agile methods.

Keywords-Agile method, agile software development, taxonomy

I. INTRODUCTION

Software processes describe how a result (what) is to be realized based on a process model (when) by which role (whom) within the framework of which activity (how) with the help of which tools and methods (with what) [1]. The use of process models enables a systematic and controllable flow of software development projects. Therefore, software process models are an important aspect in the field of software engineering. Furthermore, the choice of the process model influences various aspects of software development projects, such as product quality or project success [2], [3]. Today we know three types of process models: phase-oriented models, agile methods, and hybrid models [4].

Agile methods such as Scrum or Extreme Programming (XP) have been established in software development in the past two decades. Today, the widespread use of agile methods can be observed in many companies around the globe [2]. Agile methods have been characterized in the literature in several ways. Abrahamsson et al. characterize them as incremental, collaborative, and adaptable [5]. Beck describes them as efficient and flexible [6]. The characteristics of agile methods reflect the motivation for their use in practice. Many companies have recognized that software development projects can be subject to high dynamics, depending on the

product and market [7]. Thus, it is appropriate to choose a software development process, like agile methods, that provides opportunities to handle challenges according to these dynamic situations.

Although agile methods such as Scrum or XP combine agile practices, roles, and artifacts in order to provide a set of rules for successful use in practice, it is known that with the steadily increasing usage in the past, agile methods have been adapted [8]. This adaption usually includes the agile practices and roles described in the guidelines of agile methods [9]. The variety of agile practices makes it difficult to classify and select them for their use in practice. Furthermore, agile practices differ in context and level of detail. For example, they are described in the context of requirements engineering (e.g., User Stories [10]), estimating project effort (e.g., [11]) or product management (e.g., Definition of Done). Also, the variety of agile practices, artifacts, and roles in practice is reflected in the literature, where a different understanding of agile practices prevails, which is shown by different definitions (e.g., [12], [13]). However, for both practice and the research community, a uniform understanding of the various facets of agile practices, e.g., their level of detail, appears helpful. This paper addresses these challenges and deals with the following research questions: *How can we describe the elements of agile methods in a systematic manner?*

The paper at hand is structured as follows: In Section II, we present the background in form of the variety of agile methods, before we describe the related work in Section III. Next, we explain the selected research approach in Section IV. We present the results in Section V. First, we introduce the concept of Agile Elements in Section V-A. Based on the theoretical aspects and facets of agile methods, we describe the selected list of agile practices V-B. The main contribution of this paper, the tree of agile elements, is presented in Section V-C. Before the paper closes with a conclusion and an outlook of future work in Section VII, we describe the Threats to Validity in Section VI.

II. BACKGROUND

A variety of agile practices and methods exist in literature and practice (e.g., [12], [13]). To show this diversity, we first explain how and where agile practices are primarily defined

and described. Furthermore, we describe why the variety is constantly expanding, especially in practice.

Agile methods are usually described in guidelines. For example, the Scrum Guide defines the relevant agile practices, roles, artifacts, and values for the use of Scrum [14]. Although these guidelines describe the use of the respective agile method and define its components, it is regularly found that these frameworks deviate in practice. The deviation is due to one of the essential characteristics of agile methods: Adaptability. The aspect of self-optimization is defined in the guidelines of Scrum or XP (e.g. [14]. Dedicated practices such as retrospectives are described for this purpose. Thus, the adaptations of agile methods include the use of agile practices or roles, which leads to a multitude of agile practices over time [8].

TABLE I
OVERVIEW OF AGILE PRACTICES IN SCRUM AND XP

Scrum	XP
Product Goal	Metaphor
Product Backlog	40 Hour Week
Sprint Goal	On-Site Customer
Sprint Backlog	Coding Standards
Product Increment	Collective Ownership
Definition of Done	Continuous Integration
Sprint	Pair Programming
Daily Scrum	Coding Standards
Sprint Planning	Refactoring
Sprint Review	Testing
Sprint Retrospective	Simple Design
Refinement	The Planning Game

Several agile practices are defined and described in the guidelines of agile methods, and usually, they are following the values and principles of the agile manifesto [13]. As shown in Table 1, in Scrum and XP are twelve different practices described, each. In certain aspects, agile practices may be understood as very similar in different agile methods. An example of this is The Planning Game in XP [6] and Sprint Planning in Scrum [14]. Other agile practices differ in how they are described. The daily stand-up meeting is an example, the process of which is defined differently in Scrum [14] and Kanban [15]. There are also differences concerning the categorization of agile practices. In Scrum, agile practices are categorized into artifacts and ceremonies; in XP, no such categorization is used.

As a result of the widespread use of agile methods, new agile practices have been constantly developed, or existing ones have been adapted over the past two decades [8]. For example, the Agile Alliance currently maintains a glossary of 75 practices [16]. The glossary includes practices from methods such as Scrum and specific topic areas like product management. The multitude of agile practices results not only from the differences in meaning and goals. They are also used in combination and often adapted in practice [17].

III. RELATED WORK

The creation and use of taxonomies to describe the organization of things (e.g., their relation to one another) in a systematic manner is a well-known approach in software engineering (SE) [18]. However, in the field of agile software development, taxonomies, especially concerning this paper's topic, the structure of agile methods, practices, roles, and artifacts, have not yet been extensively investigated. However, several studies deal with taxonomies in other contexts (e.g., requirements engineering or effort estimation) of agile methods in software development. The motivation to create taxonomies includes, for example, proposing to other researchers a description of agile software development relevant aspects that is as specific as possible, such as the classification of different scale levels. Likewise, various taxonomies in agile software development are characterized by their practical relevance in that they explain the relationships between essential activities, such as requirements engineering or effort estimation.

For instance, Dingsøyr et al. present a taxonomy of scales in context of large scale agile software development [19]. Their taxonomy consists of three levels of scale (small, large and very large). The authors describe the levels depending on the number of agile software development teams and the associated coordination approaches of the teams for a project.

Saher et al. present a taxonomy for requirement changes in the context of agile software development [20]. The authors present a categorization of elements of requirement changes in order to be able to propose a model for the management of requirements changes in agile software development projects.

A taxonomy, which aims to describe dependencies in agile software development projects is presented by Strode et al. [21]. The taxonomy consists of three main categories: Knowledge dependency, task dependency, and resource dependency. Each category contains specific forms, for example, the historical knowledge in the category knowledge dependencies.

Another topic cover Usman et al. with their taxonomy of effort estimation in agile software development [22]. The taxonomy is based on several dimensions, such as the actual effort estimate (e.g., estimation unit), but also the context (e.g., agile method in use), effort predictors (e.g., size or teams experience), and estimation techniques (e.g., type of estimation like group estimation). The taxonomy was validated with case studies.

Sutling et al. present a taxonomy concerning the skills of agile project managers [23]. They performed a systematic literature review to identify four categories for their taxonomy: Skills, knowledge, personal attitude and behaviour. The categories consists of several aspects. For example, the authors identified three base skills for the category

skill: Communication skill, team building skill and problem solving skill.

Besides the SE field, taxonomies are also a topic of relevance in other research areas like information systems (IS) (e.g., [24], [25]).

IV. RESEARCH APPROACH

Several research methods are used for creating taxonomies in the field of SE [18]. We decided to use the revised taxonomy design method according to Usman et al. [18]. We argue the choice because the authors provide a specific, validated procedure for creating taxonomies. The taxonomy design method consists of four phases and 13 activities (see Fig. 1).

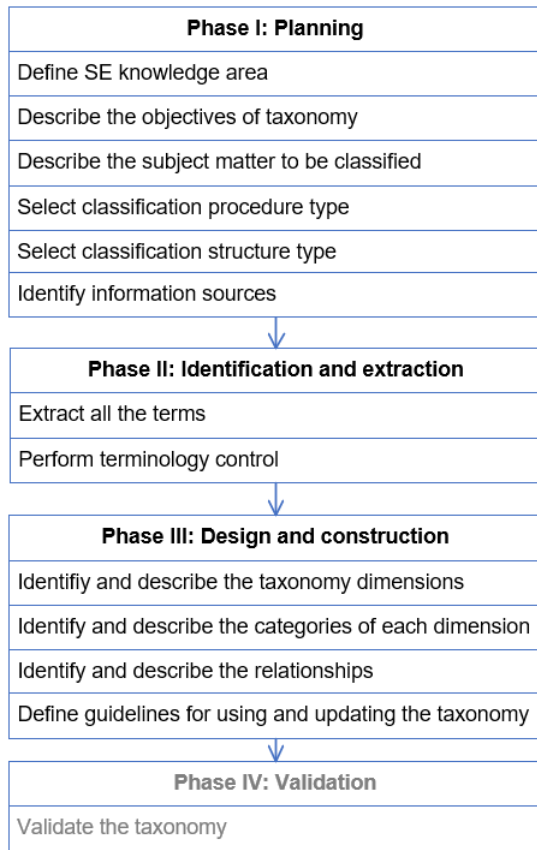


Figure 1. Taxonomy Design Method according to [18]

We conducted the phases one, two and three. Phase four, the systematic validation is out of scope of this paper. We will come back to our plans for validation the taxonomy presented in Section VII. The following subsections are structured based on the phases of the taxonomy design method approach:

A. Phase I: Planning

The first phase includes preparatory activities, which we specify in the following and explain our approach:

- *Define SE knowledge Area:* We used the Software Engineering Body of Knowledge (SWEBOK) [1] as the main reference for selecting the SE knowledge area. The SWEBOK provides 15 knowledge areas, for example: software requirements, software design and software quality. Agile methods are part of the knowledge area of "Software Engineering Models and Methods" according to the SWEBOK.
- *Describe the objectives of taxonomy:* There are different overviews of agile methods, practices, roles, and artifacts both in literature (e.g., [13]) and in practice (e.g., [16]). Some authors examine and list the agile practices used in practice (e.g., in SLRs [26]). The content in the respective overviews, glossaries, and studies differ from one another. Therefore, it is a challenge for both, researchers and practitioners, to distinguish between the respective components of agile methods. Our taxonomy represents a systematic structure of the components of agile methods and is intended to contribute to a uniform understanding of agile practices or roles.
- *Describe the subject matter to be classified:* This taxonomy takes into account the following components of agile methods: Agile practices, artifacts and roles. Furthermore, we consider other aspects such as techniques or tools for carrying out agile practices.
- *Select classification procedure type:* A choice is usually made between a quantitative and a qualitative approach when selecting the classification procedure types [18]. We use the qualitative method for this taxonomy, as we currently do not have any numerical scales for classifying agile methods and their components.
- *Select classification structure type:* According to Kwasnik, several structure types can be used for describing taxonomies: hierarchy, tree, paradigm and faceted analysis [27]. We selected a tree structure for our taxonomy. Although we have identified a hierarchy in the systemic structure, the inheritance does not necessarily have to be of a generic nature (e.g., is-a). A tree structure covers this requirement according to Kwasnik [27].
- *Identify information sources:* A sound information base is required to create the taxonomy. We use the overview of agile practices and roles by [28]. We argue the choice, explain the creation of the list and its content in Section V-B. In addition, we have taken into account practice-relevant sources such as the guidelines of agile methods (e.g., [14]) or the agile practices glossary of the Agile Alliance [16].

B. Phase II: Identification and extraction

In this second phase we describe how we extracted the relevant data from the information sources described above:

- *Extract all the terms:* We extracted the relevant information, in particular dates, concepts and dependencies, from the sources described above.

- *Perform terminology control:* While analyzing and extracting the terms in the previous activity, we noticed differences in terminology. Different authors understand different aspects under the same terms (e.g., agile practice). We present a brief overview of these differences in Section V-A and define the relevant terms for our taxonomy there in order to ensure uniform terminology.

C. Phase III: Design and construction

The aim of the third phase is to organize the extracted data and information and to systematically build up the taxonomy.

- *Identify and describe the taxonomy dimensions:* According to Broughton, we differentiate between top-down and bottom-up approaches [29]. We have taken a mixed approach here. First of all, we distinguished between three basic top-level dimensions: Activities, roles, and artifacts. We present the definitions in Section V-A. We developed the dimension of activities during data extraction because we noticed that we need an additional level of abstraction for agile practices.
- *Identify and describe the categories of each dimension:* In the next step, we developed categories for each dimension according to Broughton [29] and chose a mixed approach. We have not made any further details for roles. In the case of activities, we created the category of agile practices. The sub-category techniques is assigned to this category. The third category is tools. When we developed the categories, we used the extracted data from Phase II (see Section IV-B) and well-known concepts from the SE field [1], [30].
- *Identify and describe the relationships:* When it comes to relationships, we distinguished between the dimensions, within a dimension, and between categories and the related dimensions/categories. The relationships are delimited from one another. We have not defined any relationship between the dimensions. Activities are abstracts of agile practices, which results in dependencies between the first category level and the dimension. A collection of agile practices of the same type can also exist in the first category. Techniques are always described for a specific agile practice, just as the category tools are linked to the techniques' categories.
- *Define guidelines for using and updating the taxonomy:* The aim of a guideline for the use and adaptation of the taxonomy is to ensure usability. The dimensions, as well as categories and their sub-categories, can be expanded or changed. Practitioners can use the taxonomy, for example, to get an overview of specific agile practices or to have a checklist of which techniques they are already using for specific agile practices.

We understand and describe the taxonomy as a living artifact in the sense of agile software development and

justify this, particularly with the high dynamics in agile software development and the associated changes to be assumed in agile practices or roles. We relate these dynamics and changes primarily to the content of the taxonomy, not to the systematic aspects of the taxonomy as such.

V. RESULTS

In this section, we present the results of our taxonomy design approach, which also serves as the answer of our research question. First, we deal with the terminology of agile software development and define the terms for our taxonomy. In the following we present the selected list of agile practices, artifacts and roles and finally introduce our taxonomy: The tree of agile elements.

A. The Elements of Agile Methods

As mentioned in Section II, usually agile practices, artifacts, roles and values are described in guidelines of agile methods. In this section we will give a brief overview of the broad understanding of these terms in the literature and practice. Based on that we will define and introduce the elements of agile methods using the well-known aspects from software engineering theory according to [1], [30].

The term agile practice is not uniformly defined and used. As shown in Section II, several agile practices with various differences are defined in the guidelines of agile methods. Also, the understanding in the literature differs. According to Diebold and Zehler in [31], agile practices are “... *established instructions, e.g., tasks, activities, technical aspects, or guidelines, with a specific focus or aspect in the development of software which is performed according to single or less agile core values and Agile Principles*”. Abrahamsson et al. [5] define: “*Practices are concrete activities and work products that a method defines to be used in the process.*” Sidky et al. [32] define agile practices as “... *concrete activities and practical techniques used to develop and manage software projects in a manner consistent with the agile principles*”. Williams [13] describes them as follows: “*Conversely, practices are the applications of principles to a particular situation. Practices change as one moves from one environment and situation to another.*”.

Several aspects in the context of software processes, like artifacts, activities, methods, or roles [1], [30] should be considered to identify relevant elements.¹

Thus, we define *Agile Elements* as follows:

Definition 1: Agile elements are agile activities, artifacts or roles that are described in literature or guidelines and documents of agile methods. Activities are abstractions of agile practices. Roles define responsibilities for executing certain activities along with the skills, competences or experiences

¹For this work and the definition of Agile Elements, we adopt the commonly used definitions in software engineering for the terms software processes, artifacts, activities, methods and roles [1], [30].

necessary for their execution [1]. Artifacts are abstractions of input, output or intermediate results [1], [33].

From our point of view, there are various levels of detail in agile practices. Therefore, it seems helpful to be able to describe agile practices in an abstract manner. Thus, we define agile activities in order to be able to handle similar agile practices. In addition to these activities, we also understand roles and artifacts as agile elements.

To ensure a clear separation between agile activities and practices, based on the definition 1, we define the term Agile practice as follows:

Definition 2: Agile practices are defined as carrying out activities using tools and methods. The implementation can differ depending on the area of application.

Activities are carried out through agile practices using a (systematic) technique and, maybe, tools. A technique has to be understood as a method, according to the definition in SE [1]. We chose deliberately the term technology to distinguish it from the term agile method in this paper. Also, we use the term tool related to the known understanding in SE defined and described in [1].

B. List of Agile Practices

Various overviews and lists of agile practices exist in literature and practice. These lists differ due to different understandings and definitions regarding agile practices (see Section II). Thus, we decided to create an extended list of agile practices based on various overviews from the literature (e.g., [12], [13], [34]). We created the list of agile practices based on a systematic approach both in terms of content synthesis and source based [28].

As we created the list of agile practices last year, we conducted two steps to ensure that the content of the list is still up to date and useful for our taxonomy.

Discussion with agile experts and researchers: We verified the list and discussed it with eight experts from the agile community during workshops. The workshops were conducted during conferences and other research projects. Also, we discussed the list in several one on one talks with other researchers and agile experts.

Verify the list of agile practices based on actual findings in the literature: However, our research approach presented in this paper does not include a systematic literature review method, we tried to ensure to find the relevant literature using the guidelines according to Kitchenham and Charters [35]. We used Google Scholar for a systematic search of overviews of agile practices using the following search string: ("overview" or "list") and ("agile practice" or "agile method" or "agile methodology"). The search filter settings included studies from 2010 to October 2020. We screened a result set of 232 studies and excluded every study, which do not provide an overview or list of agile practices. We also excluded the studies, which are not dealing with software

development or software engineering and are not written in English.

Based on the findings from these discussions and the literature, we decided to use the list. Our list of agile practices includes 32 agile practices and 10 roles. We present the complete overview of the list, its origins, and the systematic creation in [28].

C. Tree of Agile Elements

Based on the selected list of practices and roles (see V-B), we created the taxonomy of agile elements. In the first step, we created the dimensions artifacts, roles, and activities. Next, we checked, which categories we could identify and create for each dimension of the taxonomy. We did not create any other categories for the artifacts and roles dimensions. As the agile activities represent abstract practices (see Definition 1), we defined agile practices as the first category of the activity dimension. The sub-category techniques is assigned to the category agile practice. We have also introduced the sub-sub-category tools, which is assigned to specific techniques. We present the systematic structure of the taxonomy in Fig. 2.

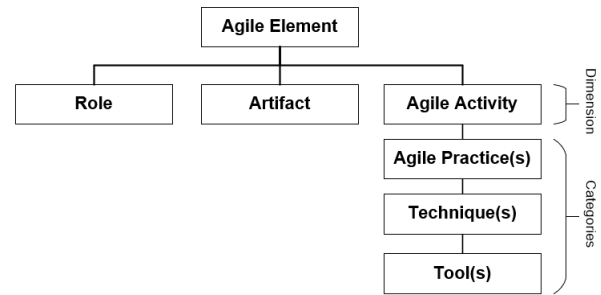


Figure 2. Taxonomy of Agile Elements

After creating the structure of the taxonomy, we transferred the entries of the overview from our list of agile practices (see Section V-B) to the taxonomy of agile elements. Therefore, we checked if there is an agile practice listed in our overview, which can be not transferred to an activity. We decided to transfer 30 agile practices into agile activities, which means that we exclude two practices from the overview of Section V-B: 30% of Cockburn's level 2 and Cockburn's level 3 people and Scrum of Scrums. We have added the team composition using Cockburn's levels as an agile practice to the activity Empowered and self-organized team. Scrum of Scrums is an agile practice used, in particular in scaled agile methods (e.g., Nexus or SAFe). We do not take this peculiarity into account at this point. From our perspective, it could be useful to include scaled agile activities in the Tree of Agile Elements in future. However, this was not within the scope of this paper.

TABLE II
AGILE ELEMENTS PER DIMENSION

Count	Agile practice	Dimension
1	Continuous Integration and builds	Activity
2	Daily Standup Meetings	Activity
3	Pair programming	Activity
4	Retrospective / Learning Loop	Activity
5	Test Driven Development	Activity
6	Review Meeting	Activity
7	Using and maintaining a backlog	Activity
8	Planning Meeting	Activity
9	Iteration based process	Activity
10	Customer integration and collaboration	Activity
11	Coding Standards	Activity
12	Refactoring	Activity
13	Tracking progress	Activity
14	Communication	Activity
15	Metaphor / Vision	Activity
16	Office structure	Activity
17	Empowered and self-organized team	Activity
18	Small and frequent releases	Activity
19	Energized Work	Activity
20	Collective code ownership	Activity
21	Documentation	Activity
22	Software configuration management	Activity
23	Knowledge sharing	Activity
24	Planning Game	Activity
25	User Stories	Activity
26	Definition of Done	Activity
27	Simple design	Activity
28	Inspections	Activity
29	Low process ceremony	Activity
30	No big design up front	Activity
31	Scrum of Scrums	Activity
32	Scrum Master	Role
33	Product Owner	Role
34	Developer	Role
35	Facilitator	Role
36	Programmer	Role
37	Customer	Role
38	Tester	Role
39	Tracker	Role
40	Coach	Role
41	Consultant	Role
42	Big Boss	Role
43	Product Increment	Artifact
44	Product Backlog	Artifact
45	Sprint Backlog	Artifact

Thus, our tree of agile elements consists of 30 agile activities, three artifacts and 11 roles. An excerpt of the tree of agile elements is shown in Fig. 3. As defined in Section V-A, we represent agile activities, artifacts, and roles as agile elements (see Definition 1). Individual agile activities can be executed in the form of one or more agile practice/s. In the example of the planning meeting activity, these are the sprint planning meeting in Scrum, the planning game in XP, or the attribute of time boxing. The level of detail can also differ from one agile practice to another. While the Sprint Planning Meeting is defined and described as an event in Scrum, collaborative planning is implicitly considered. The agile project estimation is not defined in detail in Scrum.

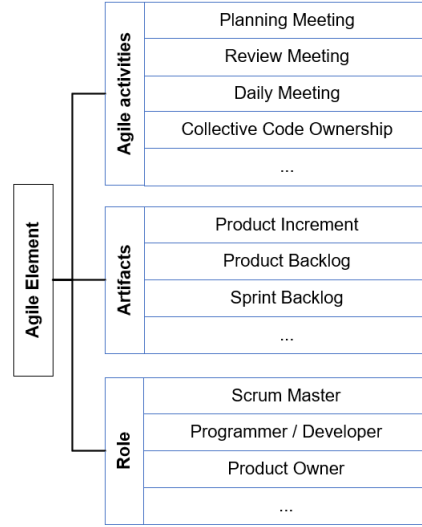


Figure 3. Example of the Tree of Agile Elements

The agile elements of the dimensions roles, artifacts and activities are presented in Table 2.

As mentioned above, we created categories for the dimension of agile activities. The first category level consists of the related agile practices to the given activity. The underlying sub-categories include techniques and tools. To illustrate this, we use the agile practice Agile project estimation (see Fig. 4). Various agile techniques related to estimating can be assigned to the Agile project estimation. As with the level of agile practices, the level of detail between the individual techniques can also differ. Planning poker, for example, is a precise technique, while point estimates could to be understood more generalised. Using the example of Planning Poker, we also show two tools with which the technology can be carried out. For instance, this can be the physical deck of cards and a virtual variant.

VI. THREATS TO VALIDITY

Although we followed a validated systematic approach to creating the taxonomy of agile elements, there are some validity threats that we need to consider and address. The structure of threats to validity is based on the guidelines of Runeson and Hoest [36].

Construct validity: With the taxonomy design method, we have chosen a systematic approach to create the taxonomy. The results are clearly described, and we have defined the ambiguous aspects, such as the terminology. The taxonomy is essentially based on our list of agile practices [28]. We verified the content of the list based on two measures, which we describe in Section V-B. A challenge in creating the taxonomy was taking into account the dynamics in the further development and pronounced adaptation of agile methods. As described in Section IV, our taxonomy of agile elements has to be interpreted as a living artifact,

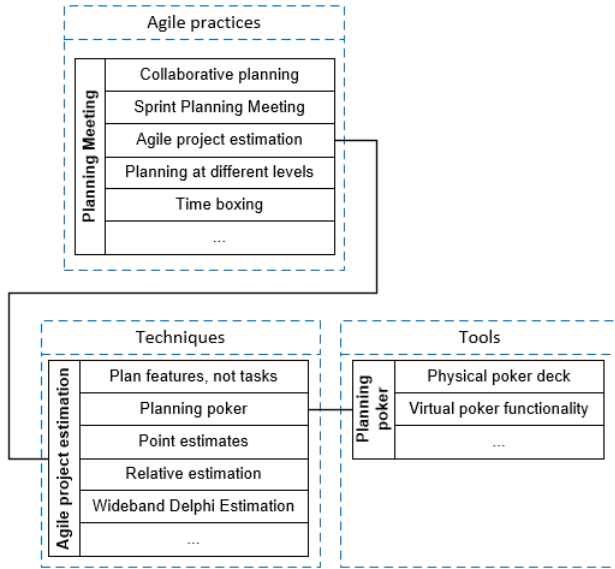


Figure 4. Example of the Tree of Agile Elements

which means that it will be adapted in the future based on further data. With this, we want to take the dynamics into account, assuming that this mainly affects the content of the taxonomy.

Internal validity: A comprehensive analysis was the basis for the creation of the taxonomy. Although these analyses were carried out systematically and based on the literature, some internal validity threats apply. It is a major challenge to ensure that the relevant literature has been considered. With the help of the systematic procedures described in this work, e.g., the identification of existing overviews of agile practices, we tried to identify as much relevant literature as possible. In order to avoid researcher bias we discussed the taxonomy with other researchers and agile experts.

External validity: The external validity is obviously limited due to the missing validation. We are planning a case study with a qualitative research approach to gain in-depth understanding of the taxonomy and its content.

VII. CONCLUSION AND FUTURE WORK

Agile methods became established approaches over the past two decades. With the steadily increasing use, the variety of agile practices has grown. This multitude is particularly evident in agile practices but also in roles and artifacts. The reasons for this can again be found in agile methods themselves. In the guidelines of common agile methods such as Scrum or Kanban, the teams are encouraged to optimize their own approach. Likewise, many companies and teams adapt certain practices or artifacts due to organizational or regulatory constraints. This paper addresses these challenges and aims to provide a systematic description of the elements of agile methods. In order to

be able to create this systematic description in the form of a taxonomy, we have chosen a systematic approach: The Taxonomy Design Method. This four phase approach includes various activities that we carried out, except for the validation phase. As a result, we have presented the variety of agile methods and their elements and defined and described the terminology for the taxonomy created. The basis for the formal and content creation of the taxonomy is our extended List of Agile Practices [28]. This overview was systematically created based on various lists of agile practices from the literature. The list also contains agile practices and roles of the best-known and most frequently used agile methods Scrum and XP.

Our taxonomy of agile elements represents agile activities, artifacts, and roles in the first dimension level. Agile activities are assigned to specific agile practices, which in turn are assigned to techniques and tools. The taxonomy is intended to contribute to a basic understanding of the elements of agile methods. For practitioners, it provides an overview to help them find their way around the variety of agile methods and, for example, to receive new incentives to use new practices or artifacts. For researcher, the taxonomy can be used as a basis, e.g., for conducting case studies and thus, contribute to a uniform understanding of the elements of agile methods.

We have not yet methodically validated the taxonomy. Only an informal exchange with other researcher and experts from the agile community has taken place. For the future, we are planning a multiple case study with a qualitative design, which includes group and individual interviews and workshops.

REFERENCES

- [1] *Guide to the Software Engineering Body of Knowledge SWE-BOK. Version 3.0.* IEEE Computer Society, 2014.
- [2] VersionOne and Collabnet, “14th annual state of agile survey report,” 2020. [Online]. Available: <https://www.stateofagile.com/>
- [3] S. L. Ramirez-Mora and H. Oktaba, “Productivity in agile software development: A systematic mapping study,” in *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017, pp. 44–53.
- [4] M. Kuhrmann, E. Hanser, C. Prause, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektene, F. McCaffery, and O. Linssen, “Hybrid software and system development in practice: waterfall, scrum, and beyond,” in *Proceedings of the 2017 International Conference on Software and System Process*, R. Bendraou, D. Raffo, H. LiGuo, and F. M. Maggi, Eds. New York, NY: ACM, 2017, pp. 30–39.
- [5] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods: Review and analysis,” vol. 478, pp. 7–94, 2002.

- [6] K. Beck, *Extreme programming explained: Embrace change*, 5th ed. Boston: Addison-Wesley, 2000.
- [7] T. Dingsøy, T. Dybå, and N. B. Moe, "Agile software development: An introduction and overview," in *Agile Software Development: Current Research and Future Directions*. Springer Berlin Heidelberg, 2010, pp. 1–13.
- [8] K. C. X. Wang and M. Pikkarainen, "Assimilation of agile practices in use," *Information Systems Journal*, vol. 22, no. 6, pp. 435–455, 2012.
- [9] M. Kuhrmann, E. Hanser, C. R. Prause, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektre, F. McCaffery, and O. Linssen, "Hybrid software and system development in practice: waterfall, scrum, and beyond," in *Proceedings of the 2017 International Conference on Software and System Process*, R. Bendraou, D. Raffo, H. LiGuo, and F. M. Maggi, Eds. New York, NY: ACM, 2017, pp. 30–39.
- [10] F. Castillo-Barrera, M. Amador-Garcia, H. G. Perez-Gonzalez, F. E. Martinez-Perez, and F. J. Torres-Reyes, "Adapting bloom's taxonomy for an agile classification of the complexity of the user stories in scrum," in *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*. Los Alamitos, CA, USA: IEEE Computer Society, 2018, pp. 139–145.
- [11] M. Duran, R. Juarez-Ramirez, S. Jimenez, and C. Tona, "Taxonomy for complexity estimation in agile methodologies: A systematic literature review," in *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, pp. 87–96.
- [12] P. Diebold and M. Dahlem, "Agile practices in practice: A mapping study," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14. New York, NY, USA: ACM, 2014.
- [13] L. Williams, "Agile software development methodologies and practices," in *Advances in computers*. Academic Press, 2010, vol. 80, pp. 1–44.
- [14] K. Schwaber and J. Sutherland, "The scrum guide," 2021. [Online]. Available: <https://www.scrumguides.org/scrum-guide.html>
- [15] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [16] A. Alliance, "Agile glossary," 2021. [Online]. Available: <https://www.agilealliance.org/agile101/agile-glossary/>
- [17] N. Kurapati, V. S. C. Manyam, and K. Petersen, "Agile software development practice adoption survey," in *Intl. Conf. on Agile Software Development*, 2012, pp. 16–30.
- [18] M. Usman, R. Britto, J. Börstler, and E. Mendes, "Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method," *Information and Software Technology*, vol. 85, pp. 43–59, 2017.
- [19] J. I. T. Dingsøy, T.E. Fægri, "What is large in large-scale? a taxonomy of scale for agile software development," *Product-Focused Software Process Improvement. PROFES 2014. LNCS*, vol. 8892, pp. 273–276, 2014.
- [20] F. B. N. Saher and O. Ghazali, "Requirement change taxonomy and categorization in agile software development," in *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, 2017, pp. 1–6.
- [21] D. Strode and S. Huff, "A taxonomy of dependencies in agile software development," in *Australasian Conference on Information Systems 2012 Proceedings*, 2012. [Online]. Available: <https://aisel.aisnet.org/acis2012/26>
- [22] M. Usman, R. Britto, J. Börstler, and E. Mendes, "An effort estimation taxonomy for agile software development," *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 4, pp. 641–674, 2017.
- [23] K. Sutling, Z. Mansor, S. Widyarto, S. Lecthmunan, and N. Arshad, "Understanding of project manager competency in agile software development project: The taxonomy," *Information Science and Applications (LNEE)*, vol. 339, pp. 859–868, 2015.
- [24] K. Conboy, "Agility from first principles: reconstructing the concept of agility in information systems development," *Information Systems Research*, vol. 20, pp. 643–659, 2009.
- [25] J. Jöhnk, M. Röglinger, M. Thimmel, and N. Urbach, "How to implement agile it setups: A taxonomy of design options," in *Proceedings of the 25th European Conference on Information Systems (ECIS)*, 2017.
- [26] S. Jalali and C. Wohlin, "Global software engineering and agile practices: a systematic review," *Journal of Software: Evolution and Process*, vol. 24, no. 6, pp. 643–659, 2012.
- [27] B. Kwasnik, "The role of classification in knowledge representation and discovery," *Libr. Trends*, vol. 48, no. 1, pp. 22–47, 1999.
- [28] M. Neumann, "Working paper: cultural impact on agile methodologies: A causal impact model," *Arbeitspapier aus der Fakultät für Wirtschaft und Informatik der Hochschule Hannover*, 2020, <https://sync.academiccloud.de/index.php/s/9GfJrdCv2g3eHuX>.
- [29] V. Broughton, *Essential Classification*. Facet Publishing, 2004.
- [30] I. Sommerville, *Software Engineering 9th Edition*. Pearson, 2010.
- [31] P. Diebold and T. Zehler, "The right degree of agility in rich processes," in *Managing Software Process Evolution*. Cham and s.l.: Springer International Publishing, 2016, vol. 2, pp. 15–37.
- [32] J. A. A. Sidky and S. Böhner, "A disciplined approach to adopting agile practices: the agile adoption framework," *Innovations in Systems and Software Engineering*, vol. 3, no. 3, pp. 203–216, 2007.

- [33] D. M. Fernández, B. Penzenstadler, and M. K. and. M. Broy, "A meta model for artefact-orientation: Fundamentals and lessons learned in requirements engineering," in *International Conference on Model Driven Engineering Languages and Systems. LNCS.*, 2010, pp. 183–197.
- [34] S. Jalali and C. Wohlin, "Agile practices in global software engineering—a systematic map," in *Intl. Conf. on Global Software Engineering*, 2010, pp. 45–54.
- [35] B. Kitchenham and Charters S., "Guidelines for performing systematic literature reviews in software engineering," 2007.
- [36] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.