FURTHER DEVELOPMENT OF AN AGILE TECHNIQUE TOOLBOX FOR MECHATRONIC PRODUCT DEVELOPMENT

K. Goevert and U. Lindemann

Abstract

Agile development gets more and more into the focus of mechatronic product development. They will solve existing chalenges e.g. digitalization with the dynamic and flexible approach of agile development. This paper focus on agile techniques, how they can be presented and linked to each other to develop an agile technique toolbox. For this nine different agile processes are analyzed and different workshops with industrial companies are implemented to meet the industrie needs. With the toolbox different possibilities are prepared to combine different techniques depending on the user needs.

Keywords: agile, toolbox, product development, organisation of product development, design methods

1. Introduction

Agile development recently focuses more and more on mechatronic product development. Companies hope to solve current challenges like digitalisation, globalization, individualization, and the need for system integrated solutions with agile development techniques (Reichwald and Piller, 2009). Agile development describes a fast, flexible, and adaptable organizational process in dynamic and uncertain situations (Hofert, 2016). However, the most companies do not focus on the details of agile development before they start applying it. They begin with agile mechatronic product development because competitors are using agile methods. However, they do not understand how to use it in the best way. That is why they typically start with scrum, the most famous agile technique (Komus and Kuberg, 2015). However, the most famous technique is not the best in every situation (Albers et al., 2015). Furthermore, Scrum only describes a management process up to requirements definition and does not describe how the requirements are realized (Goll and Hommel, 2015, p. 83).

There is a high demand for a toolbox of agile techniques. The goal of the study is a toolbox for agile techniques which adapts and further develops the paper Gövert et al. (2017). The goal of the study motivated by survey results and the knowledge that most companies only use scrum because it is the most famous one. The further developed toolbox shall be deployable in multiple situations, combine more methods, and have a strong focus on industrial demands.

2. Research methodology

This research project follows the design research methodology of Blessing and Chakrabarti (2009). The methodology is structured in the four phases research clarification, descriptive study 1, prescriptive study, and descriptive study 2.

The first phase deals with research clarification. This phase includes a first literature study and the definition of the research goal (see Section 1). Descriptive study 1 is the second phase of the research methodology. A detailed literature study is performed in this phase (Blessing and Chakrabarti, 2009). In this research project, it is a detailed literature study on agile development in general as well as agile techniques and method toolboxes (see Section 3). After the literature review, it can ascertain that an agile technique toolbox required. In the prescriptive study, solution approach requirements are identified and a solution approach is developed (Blessing and Chakrabarti, 2009). Requirements on the agile toolbox are based on industry workshops with midsized German companies in the metal and electronic industry. In the workshops, agile toolbox user stories and use cases were derived. There are two main results of the research project. The first result is the analysis of the links between agile techniques. The second result is the agile technique toolbox, the implementation based on the agile technique analysis. In this toolbox, agile techniques are linked to each other and the important information of each technique is presented in a standardized template. The last and fourth phase of the method is the descriptive study, where the approach and results of the research are validated (Blessing and Chakrabarti, 2009). In this research project, an interim result was validated in two expert workshops with two different companies.

3. Literature background

The literature research focuses on agile development, different agile techniques, and the state of the art of method toolboxes which are the focus of the research project.

3.1. Agile development

Agile development describes a fast, flexible, and adaptable organizational process in dynamic and uncertain situations (Hofert, 2016). This definition is valid for software development as well as for mechatronic product development. For software development an agile manifesto is also defined. The manifesto includes four agile values and twelve agile principles (Beck et al., 2001). The first value is that individuals and interactions are more important than processes and tools (Beck et al., 2001). The second value Beck et al. (2001) defines that a working software is more important than detailed documentation. The third value is that collaboration with the customer is more important than contract negotiations. The fourth value is that reacting to changes is more important than following a plan (Beck et al., 2001). The principles define the values in detail (see Beck et al., 2001). These values and principles are defined for software development but they are also relevant for mechatronic product development and can are transferrable (Gövert et al., 2017).

Mechatronic product development, has to be differentiated from agile project management of development projects and agile product development. In general, project management defines planning, monitoring, coordinating, and controlling measurements, which are necessary to redesign systems, processes and system solutions (Kuster, 2011). In contrast, product development defines subject specific designs, corresponding elaborations, and an integration of the subject specific designs to a complete solution (Gausemeier and Plass, 2014). These definitions apply to agile development as well and have to be taken into account in a multidisciplinary product development.

3.2. Agile techniques

Agile techniques include agile processes, methods, artifacts, roles, and tools: combining the definition of Klein (2016) with the definition of agileMPPs in Gövert et al. (2017). A process is a series of methods which are combined and used to achieve a specific goal or subgoal (Lindemann, 2009). A method is a systematic combination of tasks to achieve a specific goal and subgoal (Lindemann, 2009). Artifacts are documents and prototypes which describe product specifications and sub results in a project (Klein, 2016, p. 37). They are an outcome applying methods. Roles describe the responsibilities of the different persons in the process (Cooper and Sommer, 2016, p. 520). Lastly, tools support the activities of methods and processes (Klein, 2016, p. 37). An overview of the connection between the different agile techniques is shown in Figure 1. All agile techniques can be part of agile product development and agile project management.
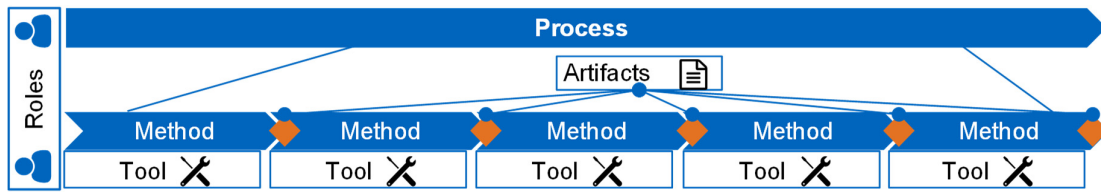
Figure 1. Overview of the connection between the different agile techniques

The main focus of the research is on the following nine agile processes and the integrated agile techniques: Scrum, Extreme Programming, Design Thinking, Lean Startup, TAF agile Framework, Agile Hybrid Model, Agile Stage Gate Model, Design Driven Development, and Disciplined Agile Delivery. These techniques are either the highly important or have been adapted to mechatronic product development.

Scrum offers an iterative project framework and supports a visible and transparent project approach. The approach does not plan each step and process in detail. The project members have a high responsibility and are structured in different roles. (Schwaber, 2004)

Extreme programming (XP) is an iterative and flexible approach as well and has a strong focus on software development. It combines existing software methods and processes with a focus on customer needs. (Beck, 2005, pp. 3–4)

Design Thinking is an agile innovation method which focuses on complex problems. The process is iterative and has a strong customer focus. The five fundamental steps are problem definition, need-finding, idea generation, prototyping, and testing. (Uebernickel et al., 2015, pp. 16–24)

Lean Startup combines approaches of agile software development and lean management (Mueller and Thoring, 2012, p. 155). It is an iterative process of the following six steps: build, code, measure, data, learn, and ideas (Ries, 2011). The process has a focus on high-tech innovations for Startups (Mueller and Thoring, 2012).

TAF Agile Framework is an agile approach which is developed for mechatronic product development. The framework starts with an idea or problem followed by requirements engineering. In each area of desirability, feasibility, and viability a Plan-Do-Check-Act-Cycle is integrated and a final solution or product is developed. (Böhmer et al., 2017, pp. 6–9)

The agile hybrid model is a mix of scrum and the waterfall model. Different methods of scrum are integrated in the development process. The main difference to scrum is that, in this process a product owner team exists. This team consists of persons from product management, development, and project management. (Schröder, 2017, p. 224)

The Agile Stage Gate process combines the stage gate process with agile methods of Scrum. The methods are used in the development phases of the stages. The benefits are a better customer focus and reduced cycle times. (Cooper and Sommer, 2016, p. 521)

Design Driven Development highlights the focus on design, especially at the requirements definition. It combines artifacts of scrum and extreme programming. (Enard et al., 2013)

Disciplined Agile Delivery is a structured process with agile elements. The process has different roles and milestones compared to traditional development processes. Also iterative elements are implemented in the development phase. (Rau, 2016)

The paragraphs above provide a very short introduction to the different processes and techniques. Detailed information of the processes and agile techniques is available in the specified literature.

3.3. Method toolboxes, method collections, and templates

A method toolbox is a knowledge collection of different methods, which are systematically structured in a template. Additionally, the methods of a toolbox can be applied in different situations of the development process and for each situation different methods are applicable. The toolbox assuts selecting the most approvable method for each different situation (Ehrlenspiel and Meerkamm, 2017, p. 360). In contrast, a method collection only describes different methods for specific applications. There is no systematic template to describe or select the methods.

Based on the results of Ponn (2007, p. 94) the scope of the study was extended to include five paper-based method toolboxes, eight paper-based method collections, three digital toolboxes, and six digital method collections. Table 1 and 2 provide a brief overview of existing toolboxes and method collections listing the name, number of methods, focus of the toolboxes, structure and type of description is, and availability of templates.

In the tables, there are two agile toolboxes (Klein, 2016; Gövert et al., 2017) and one agile method collection (agilealliance.org). The toolbox of Klein (2016) only focuses on agile techniques of the scrum process. Each scrum technique is categorized into different agility classes and work steps of the process (Klein, 2016). The other agile method toolbox focuses on 22 methods, which are categorized in the different v-model phases (Gövert et al., 2017). Most of the methods are scrum and extreme programming methods. The agile method collection of Table 2 focus on agile software. For each of them a short description is available but no selection support for specific situations exist (agilealliance.org).

The template structures are analysed, based on the identified toolboxes and method collections. Ten of the toolboxes and method collections have a standardised template. The goal of the templates is to present a short, standardized, and fast overview of the different methods (Albers et al., 2015, p. 5).

Different template criteria are listed in Table 3. Criteria which are part of the most templates are: references (in 7 of 10 methods), goal (in 6 of 10 methods), short description (in 6 of 10 methods), Input and output (in 6 of 10 methods), and an approach/ process for each method (in 5 of 10 methods).

Table 1. Paper based toolboxes and method collections

Paper based toolboxes and method collections			
Name,No of methods Toolbox (T), method collection (M)	Focus of the toolbox/ method collection	Structure and type of description	Template, Features
VDI-2221 (1993); [77 methods]; (M)	Focus on development and construction	5 categories; inconsistent	No
Eversheim (2003); [38 methods]; (T)	Focus on innovation management	No; standardized, Connected	Yes;
Strasser (2004); [106 methods]; (T)	Focus on product development in general	6 categories; standardized, connected	Yes; additional criteria
Braun (2005); [52 methods]; (M)	Focus on strategic product and process planning	No; standardized, connected	No
Pahl et al. (2007);[35 methods]; (M)	Focus on methods of draft and design phase	No; inconsistent	No
Ponn (2007); [36 methods]; (T)	Focus on situational design process planning & target-oriented application of design	16 categories; standardized, connected	Yes; matrix
Lindemann (2009) [87 methods]; (T)	Focus on product development in general	No; standardized	Yes
Naefe (2012); [22 methods]; (M)	Focus on construction methodology	9 categories; inconsistent	No
Haberfellner (2015); [107 methods]; (M)	Focus on systems engineering – system design and architecture	10 categories; inconsistent	No; categorized
Graner (2015); [17 methods]; (M)	Focus on product development in general and product development projects	6 categories; standardized	No
Klein (2016); [28 methods]; (T)	Focus on agile engineering/ scrum	4 categories; standardized	Yes; agility classes
Ehrlenspiel (2017); [90 methods]; (M)	Focus on integrated product development	3 categories; 20 sub-categories	No
IPH-Methods [73 methods]; (M)	Focus on holistic production systems	No; standardized	Yes

Table 2. Digital toolboxes and method collections

Digital based toolboxes and method collections			
Name, No of methods Toolbox (T), method collection (M)	Focus of the toolbox/ method collection	Structure and type of description	Template, Features
Portal MAP-Tool; KIT [136 methods]; (M)	Focus on the entire process chain from the market to the product	Standardized	Yes
Portal CiDaD TUM [83 methods]; (M)	Focus on product development and problem solution	Standardized	Yes
Portal innovation-wissen.de; [63 methods]; (M)	Focus on strategy and innovation knowledge	Standardized	Yes
Portal meport.net [58 methods]; (M)	Focus on general development methods	Standardized	Yes
Portal WiPro; RWTH [115 methods]; (T)	Focus on innovation methods	Standardized	Yes; Simple selection mechanism
Portal methodus; TU Braunschweig [63 methods] ; (M)	Focus on product development in general	Standardized	Yes
InnoFox; KIT [>100 methods]; (T)	Focus on product development in general	Standardized	Yes; Selection mechanism
agilealliance.org; (M)	Focus on agile methods	Standardized	No
Goevert et al. [22 methods]; (T)	Focus on agile methods connected to the v-model	Standardized	Yes

Table 3. Overview of the template structures

x = criteria is included in the template		Dis-/ Advantages	Area of application	Team/ Company size	Goal	Short description	Strength and weakness	Characteristics	Input and Output	Approach/ Process	Expenditure of resources	Complexity	Tool	Agile technique	Visualization	Synonym	Notes	References	Similar methods	Training effort	Examples
Toolboxes and method collections	Eversheim (2003)	x			x				x								x	x			
	Strasser (2004)			x	x	x		x												x	
	Braun (2005)	x			x			x		x			x					x			
	Ponn (2007)				x	x			x						x						
	Lindemann (2009)		x		x				x	x			x				x	x			
	Klein (2016)		x			x			x					x		x					
	IPH-Methods	x			x		x			x								x			x
	Portal WiPro; RWTH	x	x	x		x				x	x	x				x		x	x		
	InnoFox; KIT	x	x			x			x	x	x		x					x	x		x
	Portal methodus; TU Braunschweig					x	x		x						x	x	x	x	x		x
	Number of mentions	5	4	2	6	6	2	2	6	5	2	1	3	1	2	3	3	7	3	1	3

3.4. Research gap

Section 3.3 presents different method toolboxes and method collections but there is no method toolbox for agile methods in general. There is one paper-based method toolbox of scrum methods from Klein (2016). However, other agile methods were not taken into account in this toolbox. The toolbox from Gövert et al. (2017) has a strong focus on the v-model and is not applicable in each development process. A broad overview of agile methods is presented in the digital method collection of agilealliance.org but it is not a toolbox and they focus on software development. So users of the method collections have no support to find the right technique for each situation. From this research gap the research question of this project derived: How can agile techniques be presented to support the selection process of agile techniques, which are used in the agile mechatronic product development process?

4. Requirements on the agile toolbox

A literature research was conducted to identify requirements on an agile toolbox (Section 4.1). Additional, in a workshop with different industrial partners, industrial needs and requirements for an agile toolbox were identified and in two additional workshops use cases considered (Section 4.2).

4.1. Requirements based on literature review and the research goals

A literature review was conducted to identify requirements on an agile technique toolbox. The following eleven requirements were identified:
- R1: Clear allocation of the agile techniques to application areas (Albers et al., 2015, p. 2)
- R2: Concise short description and goal of each technique (Braun, 2005, p. 177)
- R3: Presentation of input and output information of the agile techniques usage (Braun, 2005, p. 177)
- R4: Tools and templates should be linked to the techniques (Braun, 2005, p. 177)
- R5: Application approaches should be linked to the techniques (Braun, 2005, p. 177; Ehrlenspiel and Meerkamm, 2017, p. 360)
- R6: Clear and quick identification of method benefits (Braun, 2005, p. 177)
- R7: Criteria for the technique selection (Ehrlenspiel and Meerkamm, 2017, p. 361)
- R8: Links to detailed literature (Ehrlenspiel and Meerkamm, 2017, p. 361)
- R9: Extensibility to expand and upgrade the toolbox (Ehrlenspiel and Meerkamm, 2017, p. 361)
- R10: Extend toolbox not limited to scrum techniques only (to extend the toolbox of (Klein, 2016))
- R11: Include agile techniques, which are relevant for mechatronic product development (derived from the research goal)

4.2. Requirements based on an industrial workshop

In an industrial workshop with seven German midsized companies, user stories were developed and collected for the agile technique toolbox. User Story is an agile method to identify requirements from a customer perspective at a high level (Cohn, 2010; Choma et al., 2016). From each of the seven companies', one company representative participated in the workshop. The representatives were the heads of the development department of midsize companies from the metal and electronics industry in Germany. The following main user stories for the toolbox were developed in the workshop:
- Story 1: As a manager I need a toolbox which is formulated in a neutral way and easily adaptable to specific situations.
- Story 2: As a developer, I need a toolbox which explains agile development and techniques step by step so that I am not overloaded with information.
- Story 3: As a developer, I need a fast introduction with method advantages in the foreground.
- Story 4: As a manager, I need a toolbox which presents based on few selection questions suggests a suitable number of usable agile techniques for the specific situation to have an easy and short decision process.
- Story 5: As a manager, I need an agile technique toolbox which is easy to understand so that every department and developer is inspired.

HUMAN BEHAVIOUR AND DESIGN

In two additional workshops with two of the seven companies, six relevant use cases were identified. Table 4 shows the identified use cases. The use cases describe who, how, when, and what the objective of using the agile technique toolbox is. Participants of the workshop were the heads of development departments, project managers and developers from each company. The use cases were prioritised based on input from the workshop participants. Each of them had three factors to rate the different use cases. From the twelve identified use cases, the six relevant are shown in Table 4. From these use cases, additional user stories/ requirements on the toolbox were derived. For example, the user story to corresponding use case three is: As a project manager, I need to combine a workshop with the toolbox at the beginning of the project to select agile methods which will be used in the project. From the workshop, six additional user stories were derived from the use cases. In addition to the other five from the first workshop.

Table 4. Overview of the identified toolbox use cases

	Who?	How?	When?	Objective of the use
1	Project group	Integrated overview in the toolbox	At the beginning of the project	Introduction to agile development, clarify differences between agile development and current development
2	Project manager, Product manager, (Team)	Implemented decision criteria in the toolbox	At the beginning of the project	Decision if agile development is to follow, or no
3	Project manager, Team	Workshop in combination with the toolbox	At the beginning of the project	Selection of the agile methods, which are used in the project
4	Product manager, customer, developer	Workshop in combination with the toolbox	Iterative selection	Selection of the techniques in the different phases/ tasks
5	Core team	Workshop in combination with the toolbox	At the beginning of the project	Depending on the project type the toolbox propose some agile basis methods for this application area
6	Project manager	Implemented decision criteria in the toolbox	At the beginning of the project	Selection of the agile methods, which are used in the project

5. Analysis results and the further developed agile toolbox

This section presents the results of applying the agile techniques from the nine agile processes (Section 5.1) and the resulting updated agile toolbox.

5.1. Analysis results of the state of the art

As described in Section 2, the links between processes and methods, methods and tools, methods and roles, methods and artifacts, tools and artifacts, as well as roles and artifacts were analysed. Links between processes and other techniques were not analysed because there are only indirect links via methods. Links between tools and artifacts were analysed to identify the output of the tools. To identify the responsibilities of the artifacts, the links between the roles and artifacts were analysed. An overview of the analysis results is shown in Figure 2 with exemplary extracts of the different matrices.

The extracts of Figure 2 illustrates the links of different scrum techniques. The analysis is based on information of Abrahamsson (2002, pp. 27–36) and Klein (2016, pp. 60–66). Scrum has eight different methods and most of the methods are workshop concepts or periodical meetings. There are three main roles included in scrum and four different tools. The user story for example is a method where user stories are developed. For this method, a template/ tool exists and the produced artifact of the method is formulated user stories that the development team can work with. In scrum, 11 additional artifacts can be produced. Only some of them tools are available. Other artifacts are produced with the method application. The main artifact of scrum are product increments, which are developed in different sprints. The Scrum Master is connected to all methods because he or she moderates all meetings, which are

matched in every method. Nevertheless, the scrum master is not responsible for all artifacts of the meetings. Therefore, only the responsible roles are linked to the artifacts.

For the other eight processes, the analysis of the links between the techniques exists as well. The logic of the agile technique toolbox, which is described in the next section, is based on these results.
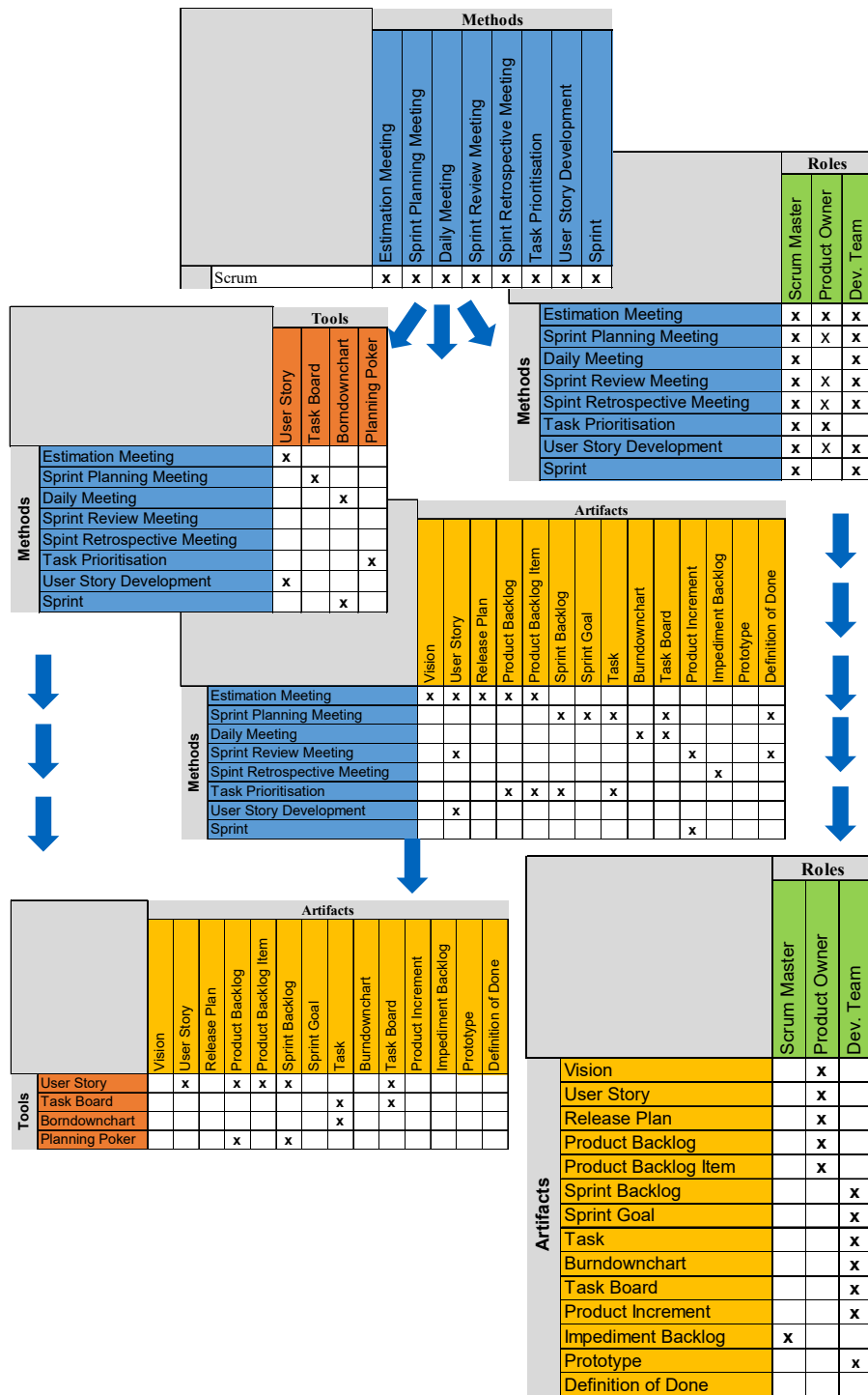
Methods

	Estimation Meeting	Sprint Planning Meeting	Daily Meeting	Sprint Review Meeting	Spint Retrospective Meeting	Task Prioritisation	User Story Development	Sprint
Scrum	x	x	x	x	x	x	x	x

Methods × Roles

	Scrum Master	Product Owner	Dev. Team
Estimation Meeting	x	x	x
Sprint Planning Meeting	x	X	x
Daily Meeting	x		x
Sprint Review Meeting	x	X	x
Spint Retrospective Meeting	x	X	x
Task Prioritisation	x	x	
User Story Development	x	X	x
Sprint	x		x

Methods × Tools

	User Story	Task Board	Borndownchart	Planning Poker
Estimation Meeting	x			
Sprint Planning Meeting		x		
Daily Meeting			x	
Sprint Review Meeting				
Spint Retrospective Meeting				
Task Prioritisation				x
User Story Development	x			
Sprint			x	

Methods × Artifacts

	Vision	User Story	Release Plan	Product Backlog	Product Backlog Item	Sprint Backlog	Sprint Goal	Task	Burndownchart	Task Board	Product Increment	Impediment Backlog	Prototype	Definition of Done
Estimation Meeting	x	x	x	x	x									
Sprint Planning Meeting						x	x	x		x				x
Daily Meeting									x	x				
Sprint Review Meeting		x									x			x
Spint Retrospective Meeting												x		
Task Prioritisation				x	x	x		x						
User Story Development		x												
Sprint											x			

Tools × Artifacts

	Vision	User Story	Release Plan	Product Backlog	Product Backlog Item	Sprint Backlog	Sprint Goal	Task	Burndownchart	Task Board	Product Increment	Impediment Backlog	Prototype	Definition of Done
User Story	x			x	x	x				x				
Task Board								x	x					
Borndownchart								x						
Planning Poker				x		x								

Artifacts × Roles

	Scrum Master	Product Owner	Dev. Team
Vision		x	
User Story		x	
Release Plan		x	
Product Backlog		x	
Product Backlog Item		x	
Sprint Backlog			x
Sprint Goal			x
Task			x
Burndownchart			x
Task Board			x
Product Increment			x
Impediment Backlog	x		
Prototype			x
Definition of Done			

Figure 2. Extract of the matrices (taking the example Scrum)

5.2. The further developed agile toolbox

A digital agile technique toolbox was developed based on the requirements and user stories (from Section 4) and the analysis results of the agile techniques (see Section 5.1). The toolbox features three levels. At the first level, the users can decide between four different possibilities to select the agile techniques for the project. The first option is to choose the techniques by answering few selected questions. Based on these questions, possible techniques are suggested depending on the situation. Additionally, these questions identify if an agile development is suitable or if it is better to develop with standardised processes. The second option is to choose one of the nine existing agile processes and all the techniques connected to them (no adaptions). The third option is to use the search option if someone likes to consider a specific agile technique. The main focus of this paper lies on the fourth option, to navigate to different techniques based on application area.

If the user choose this option, the application areas are categorized as agile project management or agile development. The project management areas are defined by Kuster (2011) and categorized into several types: coordination, application, monitoring, control, customer integration, and planning (definition see Section 3.1). The application areas of agile development are based on an analysis of the phases of the waterfall model (Goll, 2011, p. 84), stage gate process, v-model (Grande, 2011, p. 110), incremental prototyping (Goll, 2011, p. 113) and the spiral model (Schatten et al., 2010, p. 58). These are models and processes which are used in the mechatronic product development and represent application areas and tasks. The following eleven application areas are identified: system analysis, idea generation, requirements definition, system design, test/ validation, ideation, implementation, risk analysis, acceptance, integration, and market launch. If the user select one of the application areas, suitable agile methods for this area are suggested. If the user choose one of the methods, a template including short information open. The template is structured in 13 information fields. These fields are: situation/ goal, roles, advantages, disadvantages, required input, generated output/ artifacts, approach/ short description with visualization, tools, linked methods, further literature, examples, space for personal notes, and type of agile technique. These template criteria are identified from the template analysis of the different toolboxes (see Table 3). Additional criteria implemented as well, because the industry identified them as relevant. If the information on the template is provided in blue and the user chose it they forwarded to further information or to a tool/ template. Thus, the agile techniques are linked to each other. If the users want to go one step back or to return to the application field overview, buttons are provided for it. One example technique is presented in Figure 3. The user chooses the option to find the right agile technique by using application areas. The user needs a technique for work coordination, in area of the agile project management. For work coordination five different agile techniques are suggested. In this example the user chooses the task board as an agile technique. The task board is an agile technique for coordination and planning. The technique needs a team and scrum master or project coordinator. The advantages are an interactive task planning and coordination, the technique is easy to implement and team members get a fast overview of the status. Disadvantages are that working in different places is difficult and in a huge project the overview can get lost. The required input are user stories and the generated output are a task overview and current status. In the next field the approach of the technique is described and visualised. The first step of the agile technique task board is to select the user stories for the sprint. These stories are collected in the first row. Second step is to derive the tasks from each story and to collect them in the row "ToDo". If someone works on a task the user has to switch the task to the row "in progress" and fulfilled tasks is the row "done". Each team member is responsible for their own tasks. Tool of the technique is a task board template which is linked to the toolbox and the user can print it. The toolbox indicates that user stories are a connected technique. Further fields show a literature link and an example of a task board from a student project. The technique task board is an agile tool and method. In the last field the user can write own notes to the techniques. All blue fields which are shown in Figure 3 has further stored information. This templates exist for all other techniques as well.
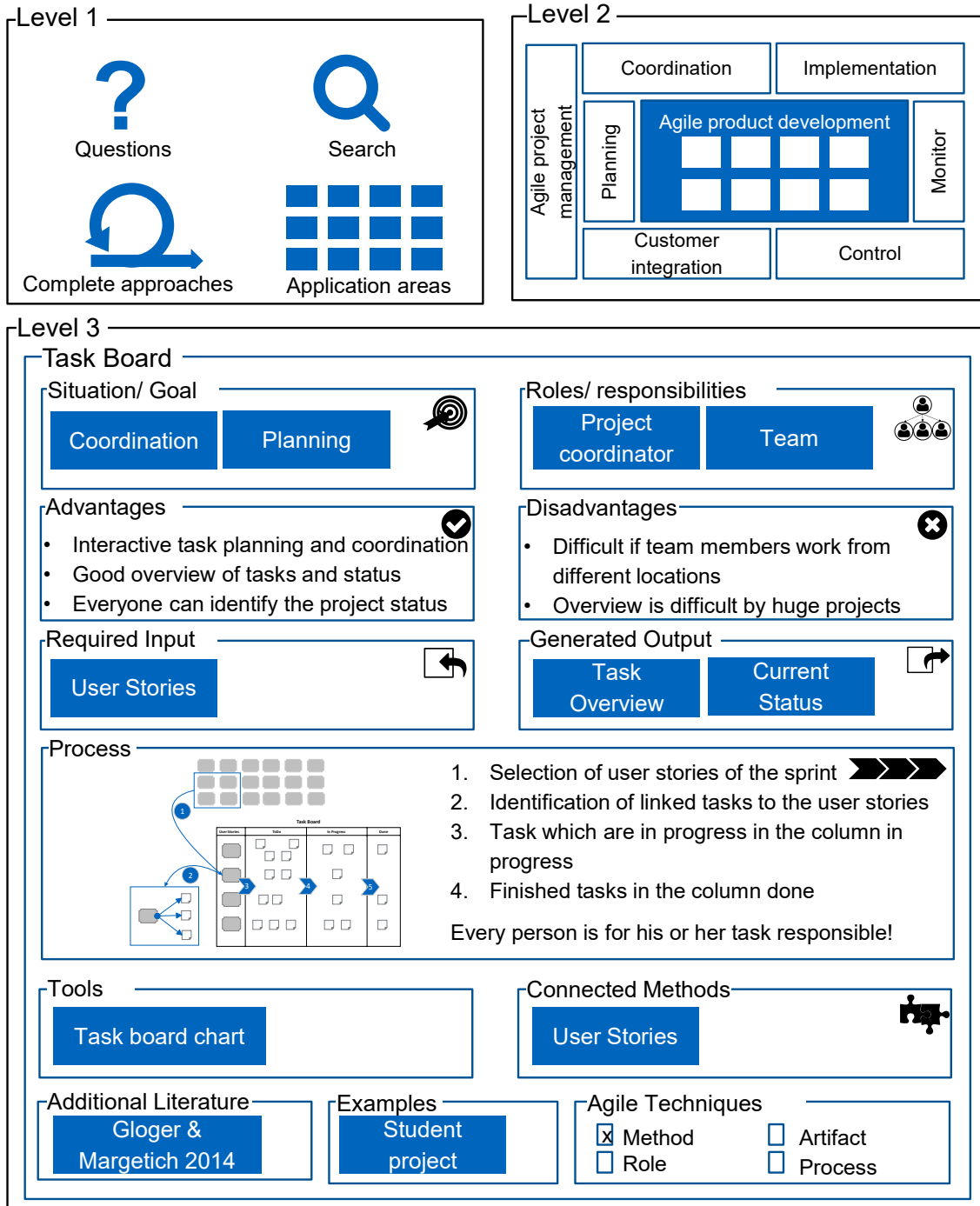
Figure 3. Example of the toolbox

6. Evaluation workshop and discussion

In an evaluation workshop in two German companies a mock-up of the toolbox was presented. The workshop participants were impressed of the toolbox and think the information is well structured and useful to decide if one technique is suitable or not. So the user stories 1 to 3 and the use cases 3, 4, and 6 are fulfilled. The focus was only on the application field part of the toolbox and not und the supporting questions. That is the reason why the user story 4 as well as use cases 2 and 5 are not fulfilled with this

part of the toolbox. The participants also criticized that the mock-up of the toolbox has no introduction to agile in general. This is part of the introduction workshop of the toolbox and the introduction workshop was not part of the evaluation (user story 5 and use case 1).

The requirements from the literature research R1 to R11 without the requirement R7 are fulfilled. The requirement R7 is only partly fulfilled. The requirement describes the need of selection criteria. Of course the application fields are selection criteria. But the detailed request and the close connection between them are included at the supporting questions.

7. Conclusion and outlook

This paper presents a digital agile technique toolbox for mechatronic product development. The paper focuses on the technique selection based on application areas. Depending on the chosen application area suitable techniques are suggested. The different techniques are described in standardized templates. The requirements, user stories and use cases on the toolbox are obtained from literature research and different industrial workshops. The content of the templates and the links between the techniques are based on a literature analysis. The templates are structured in situation/ goals, roles/ responsibilities, advantages, disadvantages, required input, generated output, process, tools, connected techniques, additional literature, examples, and what kind of agile technique it is. The resulting agile technique toolbox is evaluated in industrial workshops. From the workshops further research needs are derived.

At the moment, an implementation workshop to integrate the toolbox in companies is developed and a concept of the supporting questions are generated. If the user of the toolbox choose the opportunity selection with questions he or she can identify if agile development is really the right way and knowledge about the developing product takes into account. A pilot project in a company started. There the agile technique toolbox will be used as a starting point and then they will develop agile. The findings of the pilot project will be implemented in the toolbox after this pilot project is done. During the pilot project an additional method to adapt agile techniques on different development situations will be devised.

References

Abrahamsson, P. (2002), "Agile software development methods: Review and analysis", *VTT publications*, Vol. 478, VTT, Espoo.

Albers, A., Reiß, N., Bursac, N., Walter, B. and Gladysz, B. (2015), "InnoFox - Situationsspezifische Methodenempfehlung im Produktentstehungsprozess", *Stuttgarter Symposium für Produktentwicklung (SSP)*, Fraunhofer, Stuttgart.

Beck, K. (2005), *Extreme Programming: Die revolutionäre Methode für Softwareentwicklung in kleinen Teams; [das Manifest]*, Programmer's choice, Studienausg., 2. Nachdr, Addison-Wesley, München.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W. et al. (2001), *Manifesto for Agile Software Development*. [online] Available at: https://www.agilealliance.org/agile101/the-agile-manifesto/.

Blessing, L.T. and Chakrabarti, A. (2009), *DRM, a Design Research Methodology*, Springer, London. https://doi.org/10.1007/978-1-84882-587-1

Böhmer, A.I., Kosiol, M. and Lindemann, U. (2017), "Agile Mechatronics – Innovation Strategy for Cross-functional Teams in the Age of Uncertainty", *XXVIII ISPIM Conference: Composing the Innovation Symphony*.

Braun, T.E. (2005), Methodische Unterstützung der strategischen Produktplanung in einem mittelständisch geprägten Umfeld, Zugl., PhD thesis, Techn. Univ. München.

Choma, J., Zaina, L.A.M. and Beraldo, D. (2016), "UserX Story: Incorporating UX Aspects into User Stories Elaboration", *Human-computer interaction: 18th international conference, HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Lecture notes in computer science Information systems and applications, incl. Internet/Web, and HCI, Vol. 9731*, Springer, Cham, Heidelberg, pp. 131–140. https://doi.org/10.1007/978-3-319-39510-4_13

Cohn, M. (2010), *Succeeding with agile: Software development using Scrum*, , Addison-Wesley, Upper Saddle River, NJ.

Cooper, R.G. and Sommer, A.F. (2016), "The Agile-Stage-Gate Hybrid Model. A Promising New Approach and a New Research Opportunity", *Journal of Product Innovation Management*, Vol. 33 No. 5, pp. 513–526. https://doi.org/10.1111/jpim.12314

Ehrlenspiel, K. and Meerkamm, H. (2017), *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*, 6th ed., Carl Hanser Verlag GmbH & Co. KG, München.

Enard, Q., Gatti, S., Bruneau, J., Moon, Y.-J., Balland, E. and Consel, C. (2013), "Design-driven Development of Dependable Applications: A Case Study in Avionics", *PECCS - 3rd International Conference on Pervasive and Embedded Computing and Communication Systems, Barcelona, Spain*, SciTePress.

Gausemeier, J. and Plass, C. (2014), *Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen*, 2nd ed., Hanser, München. https://doi.org/10.3139/9783446438422

Goll, J. (2011), "Ausprägungen von Vorgehensmodellen", In: Goll, J. (Ed.), *Methoden und Architekturen der Softwaretechnik*, 1st ed., Vieweg+Teubner, Wiesbaden, pp. 81–128. https://doi.org/10.1007/978-3-8348-8164-9_3

Goll, J. and Hommel, D. (2015), *Mit Scrum zum gewünschten System*, Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-10721-5

Gövert, K., Baumgartner, M. and Lindemann, U. (2017), "The Agile Toolbox - Adapting of agileMPPs to the mechatronic development process", *21st International Conference on Engineering Design, ICED'17*.

Grande, M. (2011), "Vorgehensmodelle", In: Grande, M. (Ed.), *100 Minuten für Anforderungs - management*, Vieweg+Teubner, Wiesbaden, pp. 109–112. https://doi.org/10.1007/978-3-8348-8135-9_14

Hofert, S. (2016), *Agiler führen*, Springer, Wiesbaden. https://doi.org/10.1007/978-3-658-12757-2

Klein, T.P. (2016), Agiles Engineering im Maschinen- und Anlagenbau, PhD thesis, Technische Universität München, Forschungsberichte IWB, Vol. 323, Utz, München.

Komus, A. and Kuberg, M. (2015), Status Quo Agile: Studie zu Verbreitung und Nutzen agiler Methoden, GPM Deutsche Gesellschaft für Projektmanagement e.V., Nürnberg.

Kuster, J. (2011), *Handbuch Projektmanagement*, Springer, Dordrecht. https://doi.org/10.1007/978-3-642-21243-7

Lindemann, U. (2009), *Methodische Entwicklung technischer Produkte*, 3rd ed., Springer, Berlin. https://doi.org/10.1007/978-3-642-01423-9

Mueller, R. and Thoring, K. (2012), "Design thinking vs. lean startup: a comparison of two userdriven innovation strategies", *Leading Innovation Through Design / Proceedings of the DMI - 2012 International Research Conference*, dmi, Boston, pp. 151–161.

Ponn, J.C. (2007), Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte, PhD thesis, Technischen Universität München.

Rau, K.-H. (2016), *Agile objektorientierte Software-Entwicklung*, Springer Fachmedien Wiesbaden, Wiesbaden. https://doi.org/10.1007/978-3-658-00776-8

Reichwald, R. and Piller, F. (2009), *Interaktive Wertschöpfung: Open Innovation, Individualisierung und neue Formen der Arbeitsteilung*, 2nd ed., Gabler Verlag, Wiesbaden. https://doi.org/10.1007/978-3-8349-9440-0

Ries, E. (2011), *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*, 1st ed., Crown Business, New York, NY.

Schatten, A., Demolsky, M., Winkler, D., Biffl, S., Gostischa-Franta, E. and Östreicher, T. (2010), "Vorgehensmodelle", In: Schatten, A., Demolsky, M., Winkler, D., Biffl, S., Gostischa-Franta, E. and Östreicher, T. (Eds.), *Best Practice Software-Engineering*, Spektrum Akademischer Verlag, Heidelberg, pp. 47–69. https://doi.org/10.1007/978-3-8274-2487-7_3

Schröder, A. (2017), *Agile Produktentwicklung: Schneller zur Innovation - erfolgreicher am Markt*, Hanser, München. https://doi.org/10.3139/9783446452459

Schwaber, K. (2004), *Agile project management with Scrum*, Microsoft, Redmond.

Uebernickel, F., Brenner, W., Pukall, B., Naef, T. and Schindlholzer, B. (2015), *Design Thinking: Das Handbuch*, 1st ed., Frankfurter Allgemeine Buch, Frankfurt am Main.

Kristin Goevert, Research Associate
Technical University of Munich, Chair of Product Development
Boltzmannstrasse 15, 85748 Garching, Germany
Email: goevert@pe.mw.tum.de