NordDesign 2018
August 14 – 17, 2018
Linköping, Sweden

CPM / PDD in the context of Design Thinking and Agile Development of Cyber-Physical Systems: Use Cases and Methodology

Tobias F. Luedeke[1], Christian Köhler[2], Jan Conrad[3], Michael Grashiller[1], Andreas Sailer[1], Michael Vielhaber[4]

[1]csi entwicklungstechnik GmbH, Germany
tobias.luedeke@csi-online.de
michael.grashiller@csi-online.de
andreas.sailer@csi-online.de
[2]htw saar - Saarland University of Applied Sciences, Germany
christian.koehler@htwsaar.de
[3]University of Applied Sciences Kaiserslautern, Germany
jan.conrad@hs-kl.de
[4]Institute of Engineering Design, Saarland University, Germany
vielhaber@lkt.uni-saarland.de

Abstract

This contribution offers an integrated view of an agile development process for cyber-physical systems which includes the creative stages and the technical implementation in one methodology.
First, three project examples from different engineering disciplines (machinery and plant engineering, software engineering, product development) are described. Based on the observation findings from these projects a first approach for an agile product development process for cyber-physical products considering the outputs of Design Thinking is presented. As backbone serves the integrated product and process modelling theory CPM/PDD. The overall process reflects three different perspectives: stakeholder, product owner and development team. The artifacts from the different process steps are formalized in the product model approach from the CPM/PDD theory, independent from the product disciplines. With transformation operations these elements from the different stages can be transferred to each other.

Keywords: CPM / PDD, agile development, design thinking, creativity, cyber-physical systems

1 Introduction

To deal with the increased innovation pressure in manufacturing companies, it is crucial for the development of new products to focus on the needs of the user in the context of "feasibility“, „desirability" and "viability". A common procedure here is the human centred design to meet the needs as far as possible. However, since many characteristics only gradually become visible, agile project management has a firm place in the repertoire of the product develpoment. They allow an iterative, incremental approach so that changes in requirements and additions can be considered relatively quickly. Ideation tools such as design thinking are often used in the early phases of product development. The results of design thinking sessions then flow into the further product development process. However, this transition is often not part of the agile approach, so that at this point there is a transition like in waterfall based approaches.

This paper examines the use of the Characteristics Properites Modeling / Property Driven Development (PM/PDD) approach to map the agile development process including design thinking. The CPM / PDD theory was introduced by Weber in the 1990‘s to model product and process based on product characteristics and properties (Weber, 2005). This modell is applied in use cases to develop cyber-physical systems (CPS). In addition to traditional embedded systems, a CPS is typically designed as a network of interacting elements with physical input and output (Lee, 2008). They describe physical and software components that are deeply intertwined, operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in many ways that change with context (US National Science Foundation, 2010).

This contribution is part of an interdisciplinary research work series carried out by two research institutions and a product development service provider. The goal is to investigate more closely the practical application of innovative methods in product development, esp. in the machinery and plant manufacturing and automotive industry.

First, three anonymized use cases are considered in which cyber-physical systems are designed (section 2). Chapter three presents the findings and the necessary fields of action. In chapter 4 two fundamental questions are derived from the previous chapters are explored: How can PDD as a tool for technical development and agile development methods as well as creative techniques as the human-centered component synchronize together and how can requirements, characteristics and properties from the three different disciplines be mapped to each other. Finally, a conclusion is drawn and future work is outlined.

2 Use Cases

This section describes real experiences in three different development projects. The results are mainly presented anonymously. The examples consider sub-disciplines that are relevant for the design of a CPS: design of a physical system, software engineering and system conception.

2.1 Use Case 1: Machinery and Plant Engineering: Development of a Human-Robot Collaboration Demonstrator

This use case describes the problem of designing an automatic part feeding system and how it was solved by working with physical mock-ups.

One goal of the research project "Hey Robi" is to develop a demonstrator for testing the effects of human-robot collaboration. There, a complex task was to design a part feeding system for Lego plates, from which a robot arm can remove single plates with a high reliability. The designer was aware of the following requirements resulting from a previous conception workshop, he took part in:

- Total storage capacity (one experiment run): 60 plates of a given geometry
- Time for separation: non-productive time of the robot arm movement
- Automatic separation with low investment budget
- The robot is equipped with a standard gripper. Additional sensors are not available.

The designer recognized that the geometry of the Lego plates promoted canting of the plates in the feeder. In a first design, this effect should be excluded by sloping the plates on a slide, a common tried and tested principle. In case the slope will not be sufficient to eliminate the canting, the designer suggested working with paper liners between plates. Although manufacturing drawings have already been prepared for all the parts of the feeding system, the project team was not sure that the required function was reliably fulfilled. For this reason, it was decided to build a physical mock-up to test the properties of the feeding system.

The results were sobering: The feeder was not able to separate the plates in way that allowed the robot gripper to take them out and moreover even supported their canting.

To solve the problem, the initial idea of the slide was adapted in a way that the plates are positioned vertically one after the other in an inclined position. The plate in front is pushed beyond the edge of the slide into a stopper and at the same time is turned slightly around its own axis, as shown in Figure 1. This provides enough space for the robot gripper to handle the plate. In order to let the remaining plates automatically slide forward in a defined position, they are pushed from behind through a triangular prism with a defined weight.

These results came about through a series of practical experiments with the physical mock-up. In the end, it was possible to design a cost-effective system for the automatic feeding of complex components, which also complies with Takeda's principle of Low Cost Intelligent Automation: up to 80% lower costs compared to full automation without significant impact on performance (Takeda, 2011). Based on the findings of the mock-up, two prototypes were built from MDF boards and proven in several test runs. As the prototypes worked very reliably and fulfilled the requirements sufficiently, it was decided to use the prototypes for demonstration operation as well.

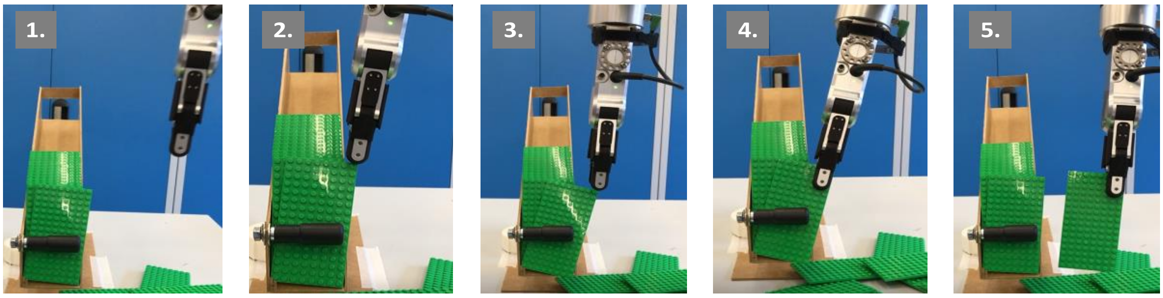

Figure 1. Handling system from use case 1

2.2 Use Case 2: Software Engineering: Development of a scheduling tool in the automotive industry

The task was to develop a software solution for scheduling within the automotive industry.

At the beginning of the project, the creative team was put together from different disciplines to start with the empathy phase. This also included end-users of the current software. Through the exchange of experience, the team was able to achieve a uniform level of knowledge on the subject area. The team shaped a future target group of the software application, taking into account future trends. This user-oriented information was summarized into a so-called persona. In addition, usage scenarios were generated via further media, from which future needs as well as applications and finally the actual problem definition emerged. Since the

current software has already reached its technical limits, it was decided to rely on a new solution. First ideas for the previously generated future persona were developed by different creativity methods. This was done in loops by working out and concretizing the ideas round by round. Ideas that turned out as a solution to the problem were presented to the entire team in order to ensure a uniform level of knowledge. In the final step of the empathy phase, potential ideas were further validated according to customer requirements. This user-oriented user stories were then translated into use cases.
Meanwhile, the development team started to define its own use cases. In doing so, they derived the use cases from their own experience as well as from past projects and began programming.
When the creative team handed over their use cases generated from the user's point of view to the development team, differences arose regarding the viewpoints on the needs of the users. On the one hand, the use cases defined by the development team were based on past experience. On the other hand, the use cases developed by the creative team were created together with end users and the help of future persona. As a solution to this disagreement, the development team began to try to bring their own use cases closer to the use cases of the creative team. From the development team's point of view, the content of the creative team's use cases contained inaccurate, non-quantifiable information that made the transfer of user requirements to technical features for programming more difficult. As the programming had already been advanced, needs could only be partially or not at all taken into account. The entire product was thus further developed in incremental development steps by the development team.
This example clearly shows the phenomenon of non-simultaneity, when the development team started to develop without coordination with the creative team on the basis of their own decisions. Another aspect was the lack of quantifiability of requirements from the creative phase. The applied methods for user-centered development from design and user experience conclude with requirements that could not be directly adopted by the development team as properties for the product. From the user's point of view, their expectations could not be fulfilled, despite the previous recording of their requirements.

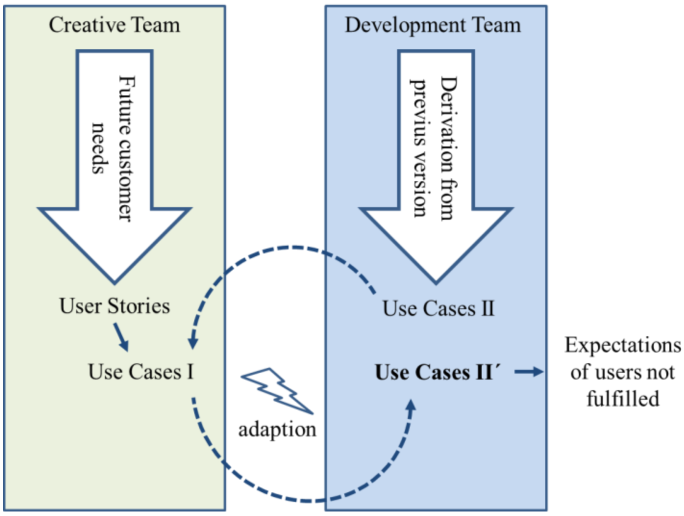

Figure 2. Interactions between creative and development team in use case 2

2.3 Use Case 3: Product Development: Development of an innovative mobility concept

In a different project, a creative team started the empathy phase of an innovation project focused on future mobility. Classic design thinking was used to generate first ideas and concepts in a heterogeneus team. The development team, which later had the task of

implementing the final concept in hardware, was not part of this phase. Rather, the focus was on the future application. In doing so, the creative team tried to find out the requirements of delivery services using observations and conducting interviews with representatives of this industry. This resulted in valuable insights from which ideas were iteratively generated. To receive initial customer feedback, representatives of the industry were invited to perform tests on the prototype. In this way, the user-oriented design could be compared with other user experiences and minor changes to the concept could be carried out quickly and cost-effectively. After several adjustment loops, the solution was a new type of mobility concept, which primarily aimed at transport and logistics in urban areas. In the later phases of prototyping and testing, a simple 1:1 wooden model was iteratively constructed, gradually adapted and finally validated with the user-centered use cases. The result was a design prototype without further details for technical implementation.
The transition to technical implementation began with the handover of the design prototype to the development team. Several design features had to be neglected in terms of feasibility. As the vehicle was gradually constructed, several ideas of the original concept were also discarded and important details of the user-centered design were lost.
In general, it was not possible to draw conclusions about the requirements from the creative phase in the later development phases. In addition, no quantifiable measurand could be found in the project that indicates to what percentage the requirements from the creative phase were achieved.

3 Findings and Need for Action

Although the case studies in the previous section are only excerpts from three development projects, they allow drawing conclusions about positive and negative influences on the development result in the relevant sub-disciplines of a CPS development.

3.1 Findings Use Case 1

The example of Use Case 1 shows a typical characteristic of a design process in mechanical and plant engineering: The technical problem-solving process is often characterized by reuse, reconfiguration or adaptation of proven solution principles (Weck & Brecher, 2006). From an isolated perspective, this is an efficient approach. However, the use case also shows that this approach is not necessarily effective when new requirements, previously unknown to the designer's pool of experience, are added. In the first draft, for example, the partial problem of plate supply was solved in principle, but not sufficiently for the additional "customer" robot. Consequently, the problem solution was not sufficiently adapted to the new application or the interaction with a new technology. This also shows that in practice it will be difficult to fully capture all requirements for new kinds of systems right from the start.
The fact that the designer was involved in the concept workshop, from which his list of requirements emerged, is positive. Furthermore, the example shows the necessity to constantly compare the partial solutions with the requirements of the overall system and not to design the feeding system in isolation as shown. If this had happened, a compensation measure that would have been unusable from an overall perspective, such as "separation of the plates by paper", would not have been proposed. However, this also illustrates once again that incompletely thought-through designs can lead to additional expenses to compensate negative effects in the later production or use processes.
A positive effect that can be seen in the use case is the collaborative and experimental approach that ultimately produced the solution that worked satisfactorily. Unfortunately, these

workshops represent additional negative expenses in this project because they were not scheduled in the original project plan. Furthermore, they were motivated by doubts about the functionality of the proposed solution. Especially when dealing with the unknown or unknown additional requirements, in this case, the workshops with physical mock-ups have proven to be very useful. During the workshops additional ideas were generated, tried out and evaluated with regard to the requirements. Several agile micro cycles of synthesis, analysis/testing and evaluation resulted in a practical solution that met the original requirements and, furthermore, requirements that had been unknown in the conception workshop.

3.2 Findings Use Case 2

Compared to use case 1, the example of use case 2 uses a lot of effort to work out the customer requirements completely and clearly and to approach the problem solution in a collaborative way. Interdisciplinary cooperation, a systematic and method-supported approach and the integration of the customer proved to be useful in developing a sustainable solution concept. The clearly formulated use cases also strengthened the customer's expectations of a satisfaction of its requirements. However, the use case also shows that a lack of process discipline and the return to old patterns of action can nevertheless lead to the customer not being offered a satisfactory product, even if its requirements were precisely formulated. As in use case 1, insufficient consideration of customer requirements leads to inefficiencies in the product development process, especially since requirements assumed or interpreted on the basis of own experience do not correspond to actual (and known) requirements. Since active customer integration has increased expectations of the result, such behavior has a negative impact on customer perception.

Use case 2 reveals another problem in passing on the requirements: The requirements developed with great effort were not sufficiently quantifiable for the development team, which also contributed to the fact that they were not sufficiently understood.

3.3 Findings Use Case 3

Use case 3 shows that methodically well-captured customer requirements can become diluted in the course of the product development process, especially if they were only recorded on one side. The presented negotiation processes indicate that a real collaboration between the phases did not take place and that the requirements determined in the first phase were not fully accepted as usable by the development team. It shows once again that especially in complex projects, which require different disciplines, a lack of early involvement of all parties involved can have negative effects on the further course of development.

This example also shows that in this project it became very difficult to evaluate the development result during the development process. The reason for this in this case was the lack of traceability of the originally intended and renegotiated requirements.

3.4 Conclusions and Need for Action

The following conclusions are based on the use cases presented and do not allow a generally valid conclusion due to the sample size. Nevertheless, they point to problems that the authors have also observed in other development projects.

Although these three examples come from three different disciplines and projects, they have some similarities that are typical for agile projects (cf. Miller, 2001):

- the associated development projects had a *modular* structure
- a more or less *collaborative* approach in requirements development or problem solving
- an *iterative* approach in the development process
- flexible *adaption* of the process in case of unforeseen events
- when things went differently than expected, an *incremental* approach was used
- *convergent* efforts have been made to solve the significant problems

Furthermore, these examples come from core disciplines that are relevant for the development of cyber-physical systems (US National Science Foundation, 2010): technical product development and software development.

On this basis, the authors try to define requirements for a development methodology for CPS:

- All examples have shown that the determination and fulfilment of requirements is crucial for product success. This applies in particular in the context of CPS, where additional requirements such as business concerns, trustworthiness or lifecycle issues of the CPS (NIST 2017) must be taken into account in addition to technical customer requirements. Consequently, an interdisciplinary and collaborative approach to requirements analysis appears to be a meaningful entry into the development process. The resulting requirements must be available, known, fully understood and accepted throughout the development process for a CPS.
- A methodical support of the requirement determination seems meaningful, above all because the user must stand in the centre of the application.
- Due to the technical complexity of CPS and the number of disciplines involved, the development process must follow a collaborative approach in which the individual disciplines work together and regularly compare their results with the customer requirements before the context of the behaviour of the overall system.
- To avoid efficiency losses, cooperation between the participating disciplines must be structured and coordinated in a joint process. Especially because the individual disciplines are methodically and culturally different.
- Due to the necessity of the error-free interaction of different technical disciplines in a CPS, the short cyclic iterative procedure of synthesis, analysis including experimental verification and evaluation, which is also used for complex problems in the individual disciplines, should be applied for the CPS development. This is particularly important, because especially for interdisciplinary, complex products to be developed, such as CPS, no discipline can fulfil the respective customer requirements independently and completely.

In summary, an approach is needed that attaches great importance to identifying and consistently fulfilling customer requirements, supports agile and incremental work and can be applied across disciplines.

4 Approach

In order to take the conclusions from the use cases into focus and to meet the requirements set in the previous chapter, the authors introduced a methodology (Luedeke et al., 2018). This methodology tends to answer two fundamental questions:

- Development process view: How can PDD as a tool for technical development and agile development methods as well as creative techniques as the human-centered component synchronize together?
- Product model view: How can requirements, characteristics and properties (in the sense of the CPM / PDD approach) from the three different disciplines be mapped to each other?

4.1 Creative Stage, Agile Development and PDD

Figure 3 illustrates the process from the stakeholder problem to the final product. Hereby, Design Thinking is used for the creativity and ideation stage and Agile Product Development for the distribution and the management of the development tasks. Simultaneously, the CPM/PDD theory offers an integrated product and process modelling. The product description is based on characteristics and properties as well as the development process is driven by the required properties.

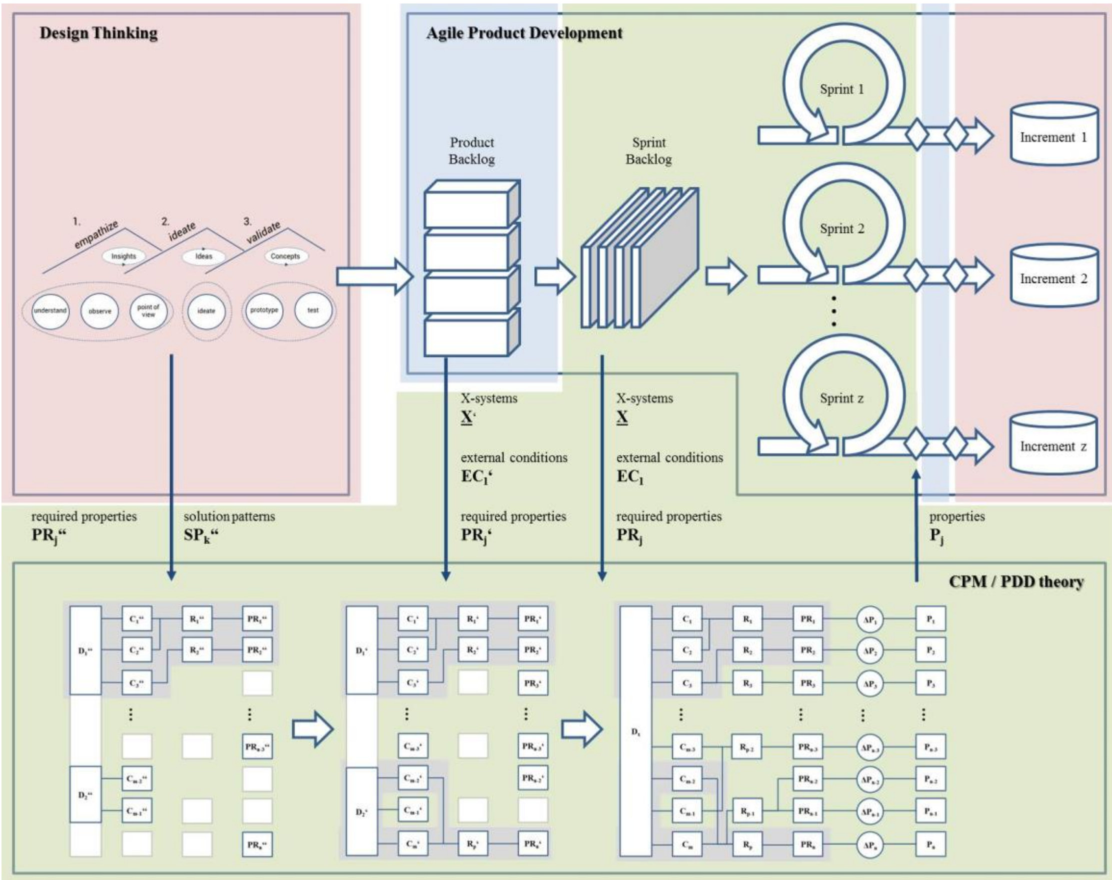

Figure 3. Overall process of the methodology (Luedeke et al., 2018)

During the process (figure 3) the different perspectives have to be considered:

- Stakeholder (light red)
- Product owner (light blue)
- Development team (light green)

The overall process is structured in the following different steps:

- Design thinking process (stakeholder perspective) with the output of design concepts and prototypes validated.

- Definition of the Product backlog (product owner perspective).
- Sprint Planning (development team perspective)
- Sprint (development team perspective)
- Sprint Review (all perspectives)
- Sprint Retrospective (development team perspective)

For further details, please see Luedeke et al. (2018).

4.2 Creative Stage, Agile Development and CPM

In order to map different artifacts from the creative stage to the agile development process and the technical implementation, the terms from the product model of the CPM / PDD approach are used. With this more or less generic description of a product model, the nature of the product does not matter. Thus, this methodology tends to be applied to different kinds of products, e.g. typical mechanical engineering products, software products, cyber-physical systems. As stated in the use cases above, the transformation of the product model elements between creative stage, agile development and technical implementation is very crucial and difficult.

Figure 4 shows the different types of CPM elements in the context of product maturity and project time. The area of the blocks shows the overall number of CPM elements which describe the product to be developed: the longer the project time, the higher the number of the elements and thus the higher the product maturity. Furthermore, we have to take into account, that customer needs can be changed on the basis of every product increment (analogous to a Minimal Viable Product) which does not mean, however, that the product maturity is reduced.

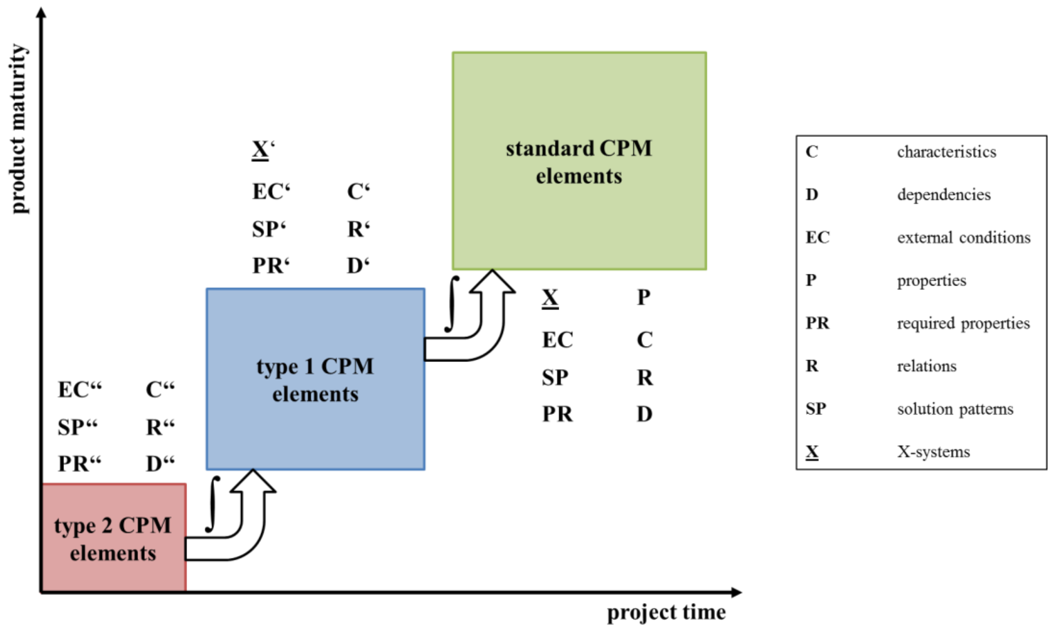

Figure 4. CPM elements in the context of project time and product maturity

The transformation between level 2 elements to level 1 elements as well as level 1 elements to standard elements can be seen as a type of integral function which has to be described in further work. The description of the transformation processes will be dependent of the product disciplines and the people involved. With a formalized transfer from requirements of the creative stage to the technical implementation, it will be possible to measure how many of the ideas of the creative stage are implemented into the real product. The more the implementation is progressed, the more user expectations are taken into account.

5 Conclusion and future work

In this paper, a methodology was presented which offers an integrated view of agile development, creative stage and the typical technical implementation. Based on three use cases taken from industry projects different problems are identifiedoccurred, especially transferring requirements throughout the projects – from user to the development team and back to the user. The methodology presented is based on the CPM/PDD approach and extends it to the creative stage and the agile project management stage by taking the perspective of the stakeholder and the product owner into account. Referring the questions, the synchronization between the different perspectives can be done with the PDD approach as backbone – starting with very fuzzy user requirements which have to be transformed to required properties. These are used by the development team which implements the product. The mapping of the product model elements is carried out by introducing type 2 and type 1 elements which are derivations of the standard CPM elements.

The focus of further work and research is the description of the transformation operations between the different perspectives and validation of the overall methodology in real industry projects. Furthermore, it has to be worked out how technical (product requirements, technical boundary conditions,...) and non-technical (supplier network, business models,...) influences on the product can be described and integrated. Finally, the creation and use of supporting tools is subject of further investigation

References

Lee, E. (2008): Cyber-physical Systems: Design Challenges. University of California, Berkeley Technical Report No. UCB/EECS-2008-8

Luedeke, T. F., Köhler, C., Conrad, J., Grashiller, M., Ruf, T., Sailer, A. & Vielhaber M. (2018): CPM/PDD as an Integrated Product and Process Model for a Design-Thinking Based Agile Product Development Process. In: DS92: Proceedings of the DESIGN 2018 15th International Design Conference, 2018, May 21-24, pp. 2063 – 2074, DOI 10.21278/idc.2018.0311

Miller, G. G. (2001): The Characteristics of Agile Software Processes. In: Proceedings of the 39th Int'l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS'01), IEEE

NIST (2017): NIST Special Publication 1500-201 - Framework for Cyber-Physical Systems: Volume 1, Overview. Issued by U.S. Department of Commerce and National Institute of Standards and Technology, online available: https://doi.org/10.6028/NIST.SP.1500-201

Takeda, H. (2011): LCIA – Low Cost Intelligent Automation, mi-Wirtschaftsbuch, Finanzbuchverlag, München

US National Science Foundation (2010): Cyber-Physical Systems (CPS), https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm, last visit: 31.03.2018

Weber, C. (2005), CPM/PDD – an extended theoretical approach to modelling products and product development processes. Proceedings of the 2nd German-Israeli symposium, Fraunhofer-IRB-Verlag, Stuttgart, pp 159–179

Weck, M. & Brecher, C. (2006): Werkzeugmaschinen 2 – Konstruktion und Berechnung. Springer Vieweg, Berlin/Heidelberg