INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018
https://doi.org/10.21278/idc.2018.0311

CPM/PDD AS AN INTEGRATED PRODUCT AND PROCESS MODEL FOR A DESIGN-THINKING BASED, AGILE PRODUCT DEVELOPMENT PROCESS

T. F. Luedeke, C. Köhler, J. Conrad, M. Grashiller, T. Ruf, A. Sailer and M. Vielhaber

Abstract

This contribution describes an approach for an agile product development process for technical products considering the outputs of Design Thinking. 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 approach transfers the agile development from software to hardware and focuses on solving the problems within the perspectives observed in practice.

Keywords: design thinking, design theory, design practice, agile development, CPM/PDD

1. Innovation pressure through market-oriented competition and digitalization

1.1. New challenges

When customers buy a product, they are mainly interested in meeting their needs. The questions of “how?” or “how long does it take?” to develop a product that fulfils their requirements do not matter to them (Ries, 2011). Furthermore, customers only buy a product when its properties are both perceived and significant to them (Kleinaltenkamp and Saab, 2009). So, the target of product development is to fulfil customer requirements in the most efficient way. But therefore it is critical-to-success to find out what customers really need. This problem is close to ISO 8402’s definition of quality, which furthermore states that there are stated and implied needs (ISO, 1994). Today's markets are characterized by market-oriented competition that leads to predatory competition (Kleinaltenkamp and Saab, 2009). This kind of competition is shaped by a vicious circle (Backhaus and Voeth, 2014): In order to stay ahead of their competitors companies develop new products and services with a simultaneous price competition that leads to an increase of customer requirements that leads again to further development activities and price competition that leads to higher customer requirements and so on. Within companies, this vicious circle is noticeable as a pressure to innovate.

This already existing pressure to innovate is further intensified by the technical possibilities of digitalization. In order to gain a better understanding of digitalization in its entirety, the Vice Executive President of Axel Springer SE Christoph Keese moved to Silicon Valley and, after his return, compared his findings with the state and the mind-set of the German economy. Keese (2016) found clear differences in the understanding of technical products and their development processes:

- In Germany, technical products are still predominantly understood as machines. In comparatively long development processes, product developers design products that are faultless at the time of delivery and prove to be reliable and robust during operation.

- In Silicon Valley, technical products are understood as solutions to customer problems. The focus is on the convenient and user-friendly interaction of the customer with the product. Due to the connectivity of the products developed there, fast update processes and thus bug fixes in the area of software and service functionalities are possible. This allows the local product developers to bring products to market in short development cycles that are not yet perfect to the point of delivery.

Taking a closer look at these differences, interesting conclusions about the development of "modern" technical products, which will mainly consist of hardware, software and networking services, can be drawn:

- The strength of traditional hardware dominated products (machines, cars etc.) is their zero-defect philosophy (Keese, 2016). This results in significant values for customers or users, such as convenience through perfect technology, which requires little attention during operation (e.g. error correction, maintenance, etc. - "install and forget"). Disadvantages, such as the complex handling of these products, have so far been accepted for lack of alternatives. The strength of the hardware-dominated product development processes lies in the ability to transform highly complex technical problems into reliable products, partly motivated by the fact that subsequent adaptations during the use phase are very costly and time-consuming.
- Software-dominated products have particular strengths in the areas of user-friendliness, networking and rapid adaptability (Keese, 2016). The customer receives products that can be operated in an intuitively manner and also participates in the further technical development after the purchase (through regular, usually free updates). The associated development processes are characterized by a deep understanding of customer interaction, speed and agility. When products are introduced to the market, they often consist only of basic functionalities and develop incrementally during the use phase. This means that modularity and flexibility in the product architecture are as important strengths of this process as distinct risk and change management.

The main difference can be visualized by the different development of the performance level of the products (Figure 1). From the customer's point of view, hardware-dominated products have a constant performance level in the course of a product generation. Improvements of the performance level take place at change of product generations (the same applies to hardware components). With software and thus also with networked products, the performance level improves due to shorter development cycles more often and also between two generations. In the case of modern products, it must therefore be possible to embed both advantages in the product model and in the product development process.

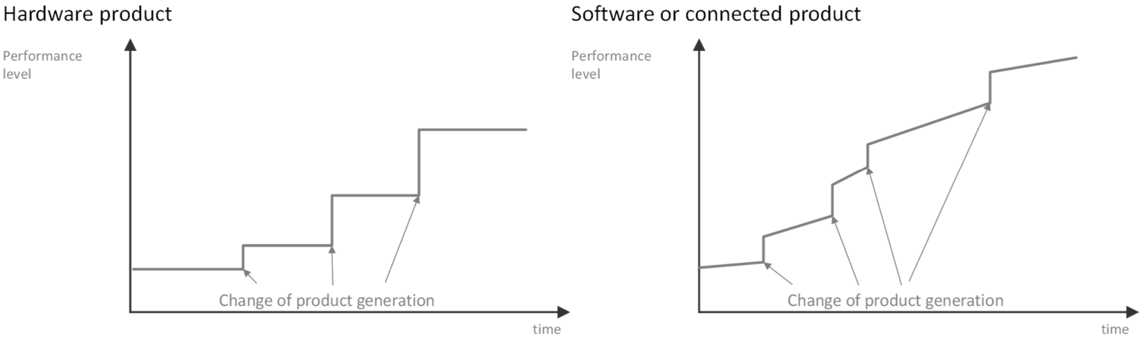


Figure 1. Performance level of hardware and software products

1.2. Scope of this contribution

This contribution is part of an interdisciplinary research work carried out by two research institutions and a product development service provider. The research partners have joined efforts to investigate more closely the practical application of innovative methods in product development, esp. in the machinery and plant manufacturing and automotive industry. After a short introduction to Agile Product Development (Section 2) and the CPM/PDD approach (Section 3), the contribution describes how to represent a practically tried-and-tested agile product development process within the CPM/PDD approach (Section 4). This is a first exploration to solve practical problems with a theoretical and scientific foundation by translating practical issues to the theory and returning the results back to practice. This helps to illustrate the practical challenges that exist and to identify further research needs (Section 5).

2. Design Thinking and agile product development

2.1. Human centeredness and Design Thinking

In our times of dynamic and radical change due to the increased innovation pressure in manufacturing companies, it is increasingly important to prepare managers for the corresponding operational tasks. Compared to traditional methods of product development, new innovation methods make a significant contribution to success because they provide the necessary basis for meeting the challenges in companies with extended competencies.

Human Centered Design is a further development of the User Centered Design, which adapts the development of new products to the user and focuses on the needs of the people in the context of "Feasibility", „Desirability" and "Viability" (Camacho, 2016). Another proven and scientifically researched approach that focuses on user needs research is Design Thinking. Over the past few years, it has become standard in innovation management, in which problems are first identified and then solved and validated by developing simple prototypes. Design Thinking according to Stanford University follows five iterative modes. The first mode, empathize, is focused on humans and their behaviours in the context of their lives using observations and interviews in order to obtain human centered insights. The define mode is the synthesis of all these empathy findings summarized in a point of view focused on specific users. From this point of view, the next mode, ideate, explores a wide variety of possible solutions using iterative ideation methods. This depository of ideas will be used in the prototype mode to transform them to the physical world. The last mode, test, refines the solutions and continually improves the design (d.school, 2013).

Several companies experimented with the individual methods with the intention of making their products more innovative for future customers - the results did not always correspond to the promised success. The central point of criticism of Design Thinking is that its use in companies fails if it is implemented without adaptation to their processes (Dark Horse Innovation, 2016).

Starting with Human Centered Design and Design Thinking, approaches have already been pursued which are based on this principle and can be easily integrated into existing product development processes. The principle of incorporating people's needs into new methods of product development seems to be ideally suited for agile project management methods. In this way, human needs can be incorporated into agile processes as requirements and subsequently worked out in incremental development steps. The Empagile process is an example of an approach that links the methodological foundations of user-oriented design with the approaches of agile project management as an implementation method. It integrates the fundamental principles of Design Thinking with its iterative process model and follows up with the incremental development sprints using agile project management frameworks. The ongoing challenge is to investigate the overlapping area between the creative stage and the product development stage in general and to add agile methods such as Scrum, in particular. (Grashiller et al., 2017).

Results from the practical application of Empagile in the past year show that especially the requirements of the early Human Centered Design phases in the later phases of agile project management cannot be fully taken into consideration. The aim is to introduce scaling in new processes that translates the requirements into properties and evaluates possible deviations in the development phase to cope with the challenges of the future.

2.2. Agile product development

Due to the iterative-incremental development, it is possible to react comparatively quickly to changing requirements. Concepts are rapidly validated via the product increment at the end of each phase. The close cooperation between customers and developers aims to achieve high product quality in a short period of time. Agile approaches are therefore particularly suitable for projects with little experience and unclear objectives. Examples include "Scrum", "Extreme Programming", or "Crystal" (Haberfellner et al., 2015).

The requirements analysis is of crucial importance in all development projects. It usually takes place at the beginning and during the development process in order to define which specifications the system to be developed should meet. In agile models, requirements change over the course of the project. This is a characteristic and thus desired feature of agility.

Through continuous delivery of product increments and the resulting early customer feedback during development, high engineering change costs in the later implementation and use can be avoided.

In general, priority is given to the implementation of increments on the basis of their receivable value. The business value for a function consists of the financial value for the company, the contribution to customer benefit and the risk. Furthermore, it is necessary to consider existing dependencies and necessary infrastructure work (Wirdemann and Mainusch, 2017). The business value from the company's point of view can be calculated precisely, whereas the customer benefit can only be estimated relatively. If the customer is well known, the estimation of customer benefit per feature works very well. The customer benefit must be estimated for unknown customers. The biggest advantage of agile approaches is that requirements with the greatest customer benefit are implemented first. However, it is not possible to quantify the degree of customer benefit.

During agile development processes, especially Scrum, it is resorted to the following artefacts and processes (Schwaber and Sutherland, 2017).

The Scrum framework specifies some fixed, recurring events which take place regularly and with a certain duration ("Time Box").

- **Sprint:** The Sprint contains all following processes and has always a constant duration. For each Sprint, a Sprint Goal is defined which must be achieved. Each Sprint results in a potentially deliverable product Increment. During a Sprint, no requirements should be changed that endanger the Sprint target.
- **Sprint Planning:** At the beginning of every Sprint the Sprint Planning takes pace. Taking into account the results of the previous Sprint, the planned capacity of the team and its previous performance, the development team, in cooperation with the Product Owner, determines the functionalities included in the product increment to be created. Unclear requirements can be clarified in this appointment. At the end of Sprint Planning, the development team commits itself to achieving the Sprint Goal.
- **Sprint Review:** At the end of each Sprint there is a Sprint Review. In this (time boxed) meeting, the "Scrum Team" presents the completed product Increment to the stakeholders. Feedback from stakeholders and other new findings flow into the product backlog and provide input for the subsequent sprint planning.
- **Sprint Retrospective:** The Sprint Retrospective takes place between Sprint Review and Sprint Planning of the next Sprint. It provides the team with an opportunity to review its processes, tools and practices and identify potential for improvement where appropriate.

Scrum uses the following artefacts that represent the value of the product and the work itself. The aim is to ensure and improve transparency and equal understanding of information created amongst all parties involved.

- **Product Increment:** The product Increment represents the result of the last and all previous Sprints. It is a potentially deliverable, usable product and meets the team's Definition-of-Done.
- **Product Backlog:** The Product Backlog is the only source of requirements for the product. The Product Owner is responsible for the Product Backlog. It does not contain all requirements from the beginning, but is continuously updated with functionalities, features or bug fixes throughout the product lifetime. The Product Backlog is a clearly ordered list. At the top there are items for the upcoming Sprint. These are formulated in such detail that it is possible for the team to estimate its effort and to fully implement the item in a Sprint.

- **Sprint Backlog:** In Sprint Planning, the development team transfers the Product Backlog items planned for a sprint to the Sprint Backlog and adds further necessary work to achieve the Sprint Goal. The Sprint Backlog represents the plan for creating the product Increment.
- **Definition-of-Done:** The development team determines which criteria must be fulfilled in order to describe a function as implemented. In addition to the development work, this includes all steps required for documentation, testing and integration into the overall product.

3. The CPM/PDD approach as a generic description of the product and product development process

3.1. A brief overview of the CPM/PDD approach

The approach of product and process modelling based on product characteristics and properties (CPM/PDD) was developed in the late 1990s at Saarland University by Weber. It arose from constructive project work with student groups in order to study the designers handling of methodologies and methods (Weber, 2012).

CPM/PDD is based on the distinction of characteristics and properties of a product. The concept of CPM represents the product model whereas PDD depicts the process of developing and designing products based on CPM. The characteristics (C) describe "the structure, shape and material consistency of a product. They can be directly determined and influenced by the designer (e.g. geometry etc.). The properties (P) describe "the product's behaviour". They cannot be directly determined and influenced by the designer, only indirectly by modifying the characteristics (e.g. weight, safety and reliability, aesthetic properties) (Weber, 2005).

The connections between characteristics and properties are represented by relations. Similar to the two main activities in the product development process, two types of relations can be distinguished: In analysis view, characteristics are known and – via analysis relations (R) – the product's properties are derived. In synthesis view, properties are known/required and – via synthesis/"inverted" relations (R-1) – the product's characteristics are established. The characteristics, properties, the two relation types and the resulting designing steps are shown in Figure 2. To describe constraints of certain characteristics, the model uses dependencies to represent them (D). Additionally, external conditions (EC) affect the context in which (analytical as well as synthetic) statements are valid.

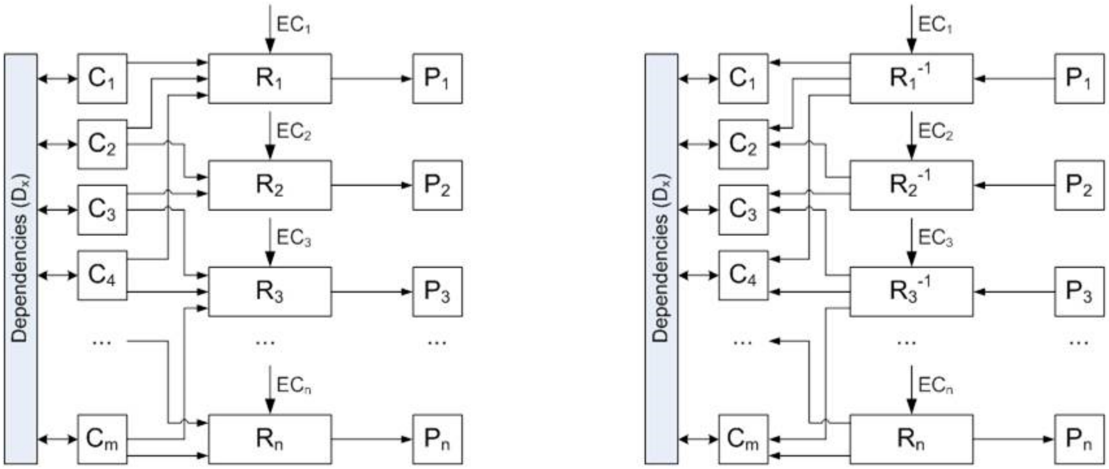


Figure 2. Analysis (left) and synthesis (right) in the CPM model (Weber, 2005)

PDD describes the product development process based on the CPM product model. Thereby, a given set of required properties should be met by the behaviour of the design solution. In a new product development project, (almost) nothing is known about structure and characteristics of the solution at the beginning. In case of an adaptive design, a list of characteristics is "set" from the beginning, together with the properties they bring along.

The PDD process consists of several iterations of the following four steps (Figure 3):

- **Synthesis step:** Determination and assignment of characteristics from the overall evaluation (in the first step of new product development, they are derived only from the properties required)
- **Analysis step:** Determines or predicts the properties from the given characteristics
- **Individual deviation:** Evaluates every property against its property required
- **Overall evaluation:** Draws conclusions from the individual deviations and drives the actual development process, i.e. strategy for synthesis or termination of the process.

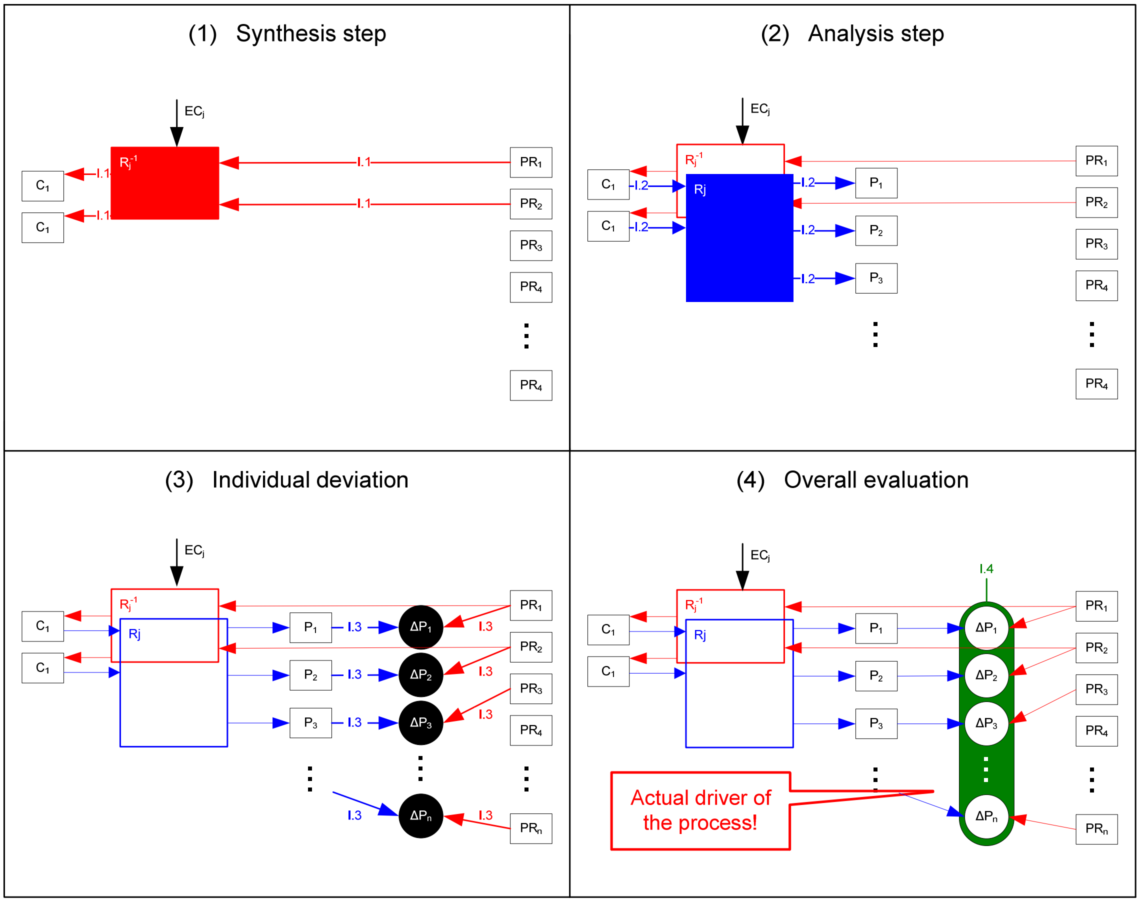


Figure 3. Visualisation of the PDD process (Weber, 2005)

The four steps of the PDD process are carried out as long as all properties fulfil the properties required sufficiently. Thereby, the properties have to be determinable and predictable with sufficient certainty and accuracy. Additionally, for termination it is necessary that all characteristics needed for manufacturing and assembly of the product are established and assigned.

3.2. Solution pattern in CPM/PDD

The term solution pattern is defined in the literature in different ways. VDI Guideline 2221 (VDI 2221, 1987) describes them as representational products or their parts for determining and displaying properties (especially design and function). Wanke defines them, based on Grabowski et al. (1993) as an application-neutral description of a solution that can be adapted to specific problems, since the basic functionality and the basic structure are described by the solution pattern. The key point is according to Wanke that a solution pattern is given without instantiation. This means that there are preliminary no concrete characteristic or property values (Wanke, 2010). To move from a solution pattern to a solution, an instantiation must be carried out. This means that characteristic and property values must be assigned to the solution patterns.

The use of solution patterns as well as the general reuse of components or information offers several advantages (Weber, 2007):

- Risk limitation (using proven and tested solutions)
- Facilitate and accelerate development / design
- Reuse of knowledge, standardization
- Enabling product modularization

Within the product development process based on CPM/PDD, recurring patterns occur in the characteristics/properties network, which can also be interpreted as solution patterns. The great advantage of using CPM/PDD is according to Wanke (2010) the separation of properties and characteristics of the Solution Pattern. This allows a behavioural description of the (partial) solution, which is not limited to a few individual behavioural elements. Due to the depicted characteristics, the developer can at the same time recognize and edit the parameters that can be directly influenced (Figure 4).

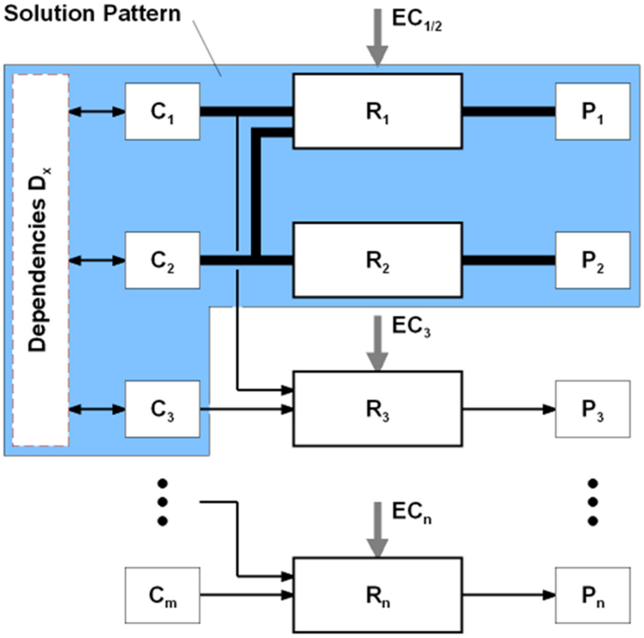


Figure 4. Design pattern in CPM model (Wanke, 2010)

4. Agile product development using CPM/PDD

In order to be able to represent Agile Product Development within in CPM/PDD approach, first of all, two fundamental questions have to be answered:

- From the point of view of the development process: How can the work steps and iterations of the two approaches be synchronized? Do PDD and agile product development together?
- From the point of view of the product model: What adjustments or extensions are needed within the existing CPM/PDD approach in order to be able to map Agile Product Development consistently?

4.1. Agile development process with PDD

Figure 5 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.

During the process the different perspectives have to be considered:

- **Stakeholder** (light red in Figure 5): needs insights/trends and delivers the product idea, gets product increment and does evaluation of it.
- **Product owner** (light blue in Figure 5): needs the product idea and delivers required properties and external conditions, is responsible for prioritization and monitoring of required product properties
- **Development team** (light green in Figure 5): needs technical requirements and delivers increment properties and functionality

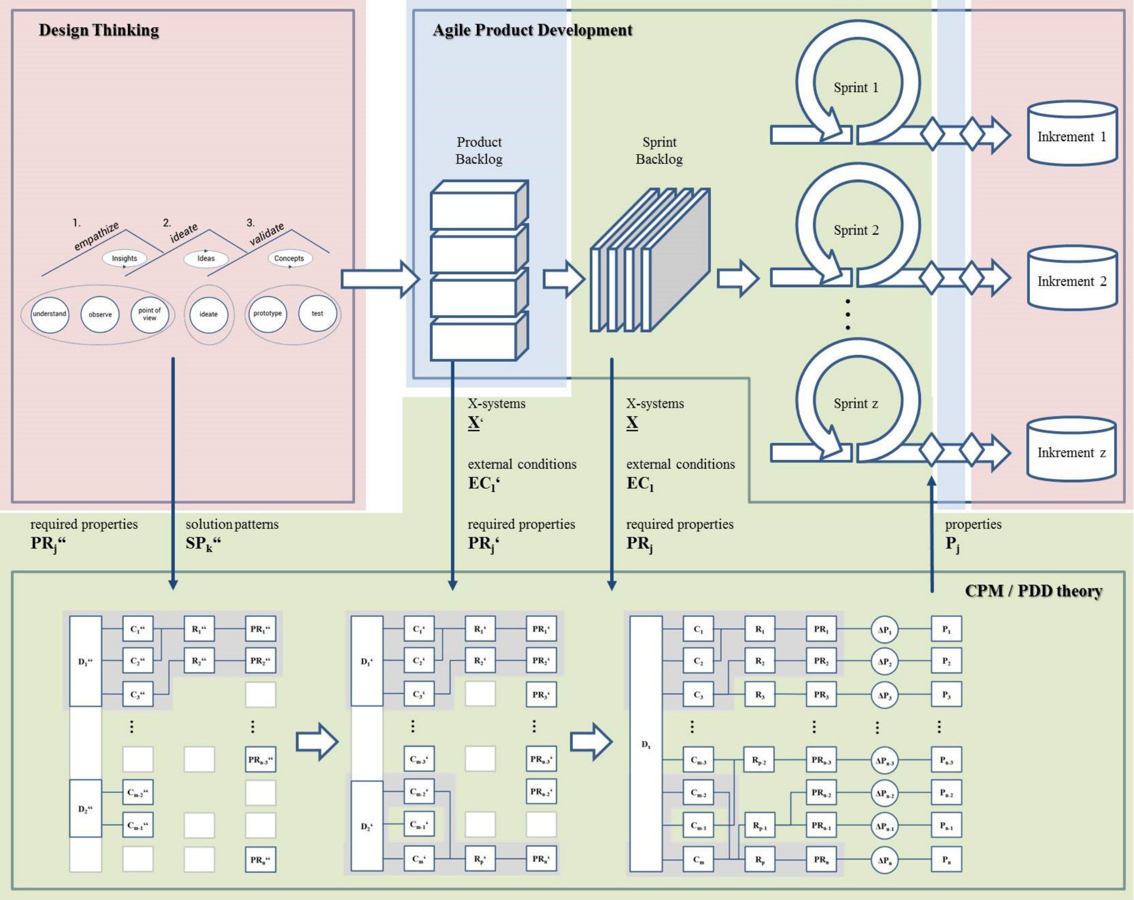


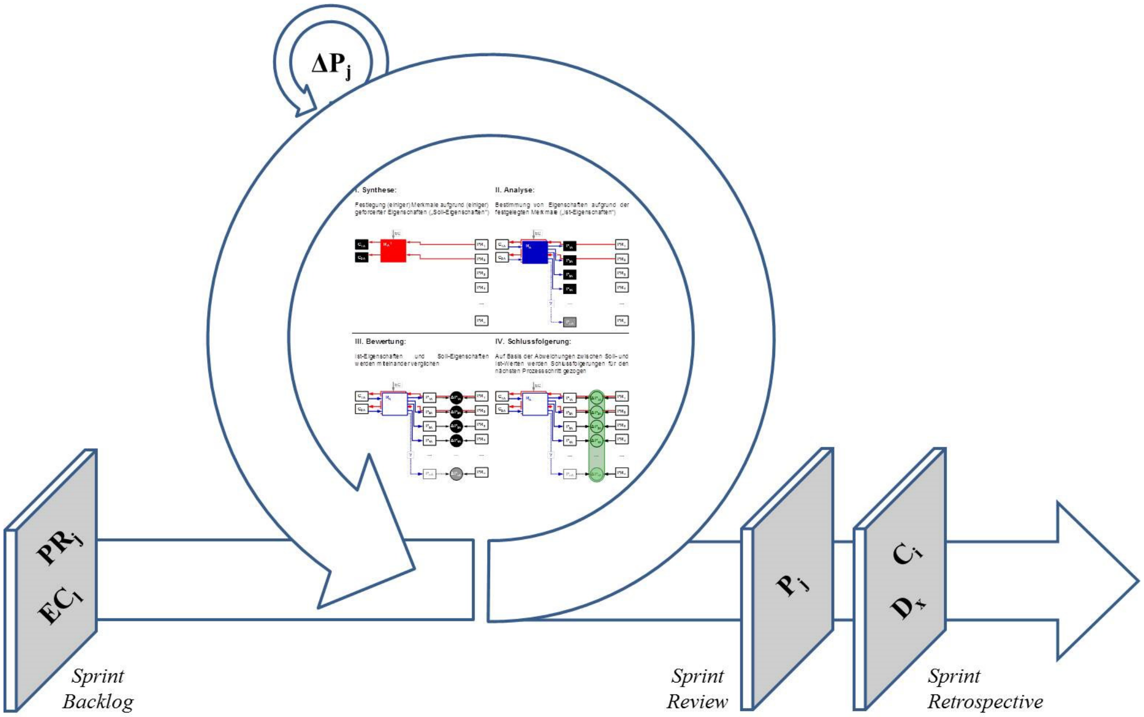
Figure 5. Overall process

The overall process is structured in the following different steps:

- During the **Design Thinking process** (stakeholder perspective) design concepts and prototypes are created and validated. In principle, that means that certain solution patterns on a fuzzy level with high uncertainty can be found when analysing for example functions and geometry of the prototypes. Required properties, characteristics and their relations can be derived from these rudimental solutions patterns and seen as the starting point for CPM product model (compare to Figure 5).
- Definition of the **Product backlog** is the task of the product owner. The stakeholder requirements from the solution patterns - which are rather qualitative than quantitative - are transformed into more quantitative required properties and additional required properties and external conditions are added. More solutions patterns - maybe not visible for the stakeholder - are considered to ensure the product functionality.
- After prioritization of the product backlog by the product owner - thus, the required properties - the development team plans the upcoming **sprint**: the required properties that are necessary to

build up a functional increment in a given time are chosen. Thus, a specific knowledge is required which can be supported by the rough CPM model from the creativity stage and by the engineer's experience. The Definition-of-Done is determined by the specification of targeted values of the set of ΔP (e.g. "ΔP_1 has to be smaller than 0.1").

- Running through a sprint (Figure 6) is the "real" development work in the way of **Property-Driven Development**, described in a previous section. The developer starts from the required properties, the external conditions and the X-systems, which are derived from the Sprint Backlog or transformed from those who are given by the product owner or the stakeholder. The developing process runs through the phases of PDD - starting from a synthesis of the required properties to characteristics over the analysis stage where the properties are determined to the deviation and the evaluation of the required properties against the developed properties. The validation of the required properties with designed properties (determination of ΔP) is conducted in micro-iterations, compared to Daily Scrum Meetings in Agile Development.
- In **Sprint Review** meetings the increment functionality which is described by the designed properties will be validated by all participants of the process (stakeholder, product owner, and Development Team).
- The **Sprint Retrospective** serves as review for the development team for evaluating their development process in the sprint just finished. A review of the creation process of the characteristics and the dependencies of the used characteristics is conducted.



ΔPj
I. Synthese:
Festlegung (einiger) Merkmale aufgrund (einiger) geforderter Eigenschaften („Soll-Eigenschaften")
II. Analyse:
Bestimmung von Eigenschaften aufgrund der festgelegten Merkmale („Ist-Eigenschaften")
III. Bewertung:
Ist-Eigenschaften und Soll-Eigenschaften werden miteinander verglichen
IV. Schlussfolgerung:
Auf Basis der Abweichungen zwischen Soll- und Ist-Werten werden Schlussfolgerungen für den nächsten Prozessschritt gezogen
PRj
ECl
Sprint Backlog
Pj
Sprint Review
Ci
Dx
Sprint Retrospective

Figure 6. Sprint

4.2. Adjustments and extensions of the CPM product model for agile product development

Solution patterns occur at several points in the process described in Section 4.1. As described in Section 3.2, solution patterns represent recurring combinations of characteristics (incl. dependencies), relations and properties within a CPM network that can among others be used to modularize products. But, in order to map the agile product development process described above in the CPM product model, the existing definition of solution patterns must be adapted.

The still very rudimentary solution pattern, which originates from the creative phase (e.g. represented by a Design-Thinking (pre-) prototype) differs structurally from a solution pattern in final product. Since development requirements manifest themselves in the prototypes, the structure must include properties required in addition to characteristics (incl. some dependencies), relations and properties. For this reason, this kind of a solution pattern is referred to as **type 2 solution pattern (SP'')**. Since the characteristics, dependencies, relations and properties of type 2 solution patterns do not or do not fully correspond to those of the final product, they are also referred as CPM elements of type 2. Type 2 CPM elements have the following features:

- **Type 2 properties (P'')** do not match to a high probability with the properties of the finished product. Moreover, they can even be very different from those of the finished product (e.g. wooden prototypes vs. steel products). Thus, they might not be of any significance for the further product development process except for an illustrative character.
- **Type 2 relations (R'')** will also not be significantly consistent with the relations of the finished product. Furthermore, they might even be very fuzzy, technically not yet solved or technically insoluble.
- **Type 2 characteristics (C'')** and **type 2 dependencies (D'')** form a rough structure of characteristics and dependencies that are important for the further development process (e.g. geometry). However, they are not yet as precise or fully defined as required by the final product.

In practice, the requirements from the creative phase generally become more concrete as the development process progresses. Therefore, these are also referred **as type 2 properties required (RP'')** in the presented model. However, these type 2 properties required are already of high relevance for the development process. The same applies to the external conditions which influence the type 2 solution pattern. In order to stimulate creativity, in the early phase, boundary conditions are often softened or faded out. As a result, these **type 2 external conditions (EC'')** also become more stringent with the progress of an agile product development project. The structure of a type 2 solution pattern is shown in Figure 7.

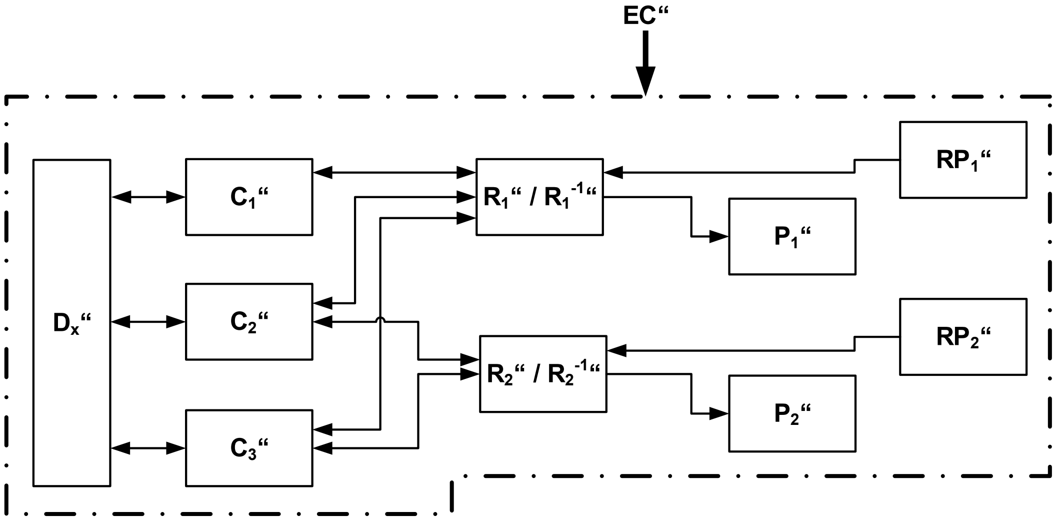


Figure 7. Structure of a type 2 solution pattern

As shown in Figure 5, the product owner adds further properties required and external conditions and, thus, makes them more concrete. Consistently, these are called **type 1 property required (PR')** and **type 1 external conditions (EC')**. The product owner also adds X-systems. Since these might be not as specific as those in later development, these are also called **type 1 X-systems**. It is worth to mention that the CPM elements used in the sprints correspond to those of the original CPM model and therefore do not get a name suffix.

The solution patterns are needed a second time within the depicted Agile Product Development process: they represent the product increments in the sprints. In order to make it possible to develop a complete product in sprints, the individual increments have to be able to be inserted in a constantly growing CPM product model. This results in two further requirements:

1. Solution pattern must be able to get composed of further solution pattern fractally.
2. In order to allow agile development and agile exchange of product increments in case of changing customer requirements (or bad customer feedback), it must be possible to easily identify the relationships between the (fractal) solution patterns.

The first requirement seems easy to solve within the existing CPM concept. This is merely a question of presenting the solution pattern (e.g. outer and inner frame). The second requirement must be examined in more detail. For example, as solution could be based on an extension of the CPM/PDD based engineering change process as presented by Köhler (2009).

5. Conclusion and need for further research

This contribution has shown that it is possible to represent agile product development in CPM/PDD. To do so, the CPM/PDD approach had to be extended by only a few elements: the introduction of the stakeholder and the product owner perspective, the extension of the idea of the solution patterns with CPM/PDD and an instantiation of the CPM/PDD elements, so that the increasing concretisation within the agile product development can be displayed. Thus, the fundamentals have been set in order to solve problems of Agile Product Development in CPM/PDD on a theoretical level and to return them later to real methods. Examples of these problems are:

- The support of the definition of done
- Content-related and temporal concretisation of requirements and boundary conditions
- The development and linking of increments in products consisting of hardware, software and connected services

Furthermore, it is necessary to integrate other approaches of the creative phase besides Design Thinking. Further research will focus on the detailed elaboration of the described perspectives, mainly on the interfaces among each other. Especially the consistent and holistic formulation of the entities of the integrated CPM/PDD product and process model is important.

References

Backhaus, K. and Voeth, M. (2014), *Industriegütermarketing*, Vahlen, München.

Camacho, M. (2016), "David Kelley: From Design to Design Thinking at Stanford and IDEO", *The Journal of Design, Economics, and Innovation*, Vol. 2 No. 1, pp. 88-101. https://doi.org/10.1016/j.sheji.2016.01.009

d.school (2013), *bootcamp bootleg*. [online] Stanford University. Available at: https://dschool.stanford.edu/resources/the-bootcamp-bootleg

Dark Horse Innovation (2016), *Digital Innovation Playbook*, Murmann Publishers, Hamburg.

Grabowski, H., Rude, S., Suhm, A. and Staub, G. (1993), "Lösungsmusterbasierte Produktmodellierung in wissensbasierten Konstruktionssystemen", *Rechnerunterstützte Wissensverarbeitung in Entwicklung und Konstruktion '93, Heidelberg, Germany, September 28-29, 1993,* VDI-Verlag, Düsseldorf, pp. 55-80.

Grashiller, M., Luedeke, T.F. and Vielhaber, M. (2017), "Integrated approach to the agile development with design thinking in an industrial environment", *Proceedings of the 21st International Conference on Engineering Design (ICED 17), Vol. 2: Design Processes | Design Organisation and Management, Vancouver, Canada, August 21-25, 2017*, The Design Society, Glasgow, pp. 239-248.

Haberfellner, R., Fricke, E., de Weck, O. and Vössner, S. (2015), *Systems Engineering - Grundlagen und Anwendung*, Orell Füssli Verlag, Zurich.

ISO (1994), ISO 8402 Quality management and quality assurance – Vocabulary, ISO.

Keese, C. (2016), *Silicon Germany: Wie wir die digitale Transformation schaffen*, Knaus Verlag, Munich.

Kleinaltenkamp, M. and Saab, S. (2009), *Technischer Vertrieb - Eine praxisorientierte Einführung in das Business-to-Business-Marketing*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-79533-9

Köhler, C. (2009), Technische Produktänderungen – Analyse und Beurteilung von Lösungsmöglichkeiten auf Basis einer Erweiterung des CPM/PDD-Ansatzes, PhD thesis, Saarland University. https://doi.org/10.22028/D291-22621

Ries, E. (2011), *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*, Portfolio Penguin, London.

Schwaber, K. and Sutherland, J. (2017), *The Scrum Guide*. [online] Scrum guides. Available at: https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf (accessed 12.11.2017).

VDI 2221 (1987), Systematic approach to the design of technical systems and products, VDI, Düsseldorf.

Wanke, S. (2010), Neue Konzepte zur Verwaltung und Bereitstellung von Lösungen im Produktentwicklungsprozess - CPM/PDD-Lösungsmuster als Grundlage eines verhaltensbeschreibenden Lösungskataloges, PhD thesis, Saarland University. https://doi.org/10.22028/D291-22645

Weber, C. (2005), "CPM/PDD—an extended theoretical approach to modelling products and product development processes", *Proceedings of the 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, Berlin, Germany, July 7-8, 2005*, Fraunhofer-IRB-Verlag, Stuttgart, pp. 159–179.

Weber, C. (2007), "What Makes Engineering Design Science „Applied"?", *Proceedings of the AEDS 2007 Workshop, Pilsen, Czech Republic, October 26-27, 2007,* Applied Engineering Design Science Special Interest Group (AEDS-SIG)*,* pp. 89-101.

Weber, C. (2012), "Produkte und Produktentwicklungsprozesse abbilden mit Hilfe von Merkmalen und Eigenschaften – eine kritische Zwischenbilanz", *Proceedings of the 23rd Symposium Design for X (DFX 2012), Bamberg/Erlangen, Germany, October 4-5, 2012*, TuTech Innovation GmbH, Hamburg, pp. 25-62.

Wirdemann, R. and Mainusch, J. (2017), *SCRUM mit User Stories*, Carl Hanser Verlag, Munich. https://doi.org/10.3139/9783446450776

Dr.-Ing. Tobias F. Luedeke, Design and Development Engineer
csi entwicklungstechnik GmbH, Innovations
Sachsstr. 5, 85080 Gaimersheim, Germany
Email: tobias.luedeke@csi-online.de