NordDesign 2018
August 14 – 17, 2018
Linköping, Sweden

Managing Systems of Objectives in the agile Development of Mechatronic Systems by ASD – Agile Systems Design

Albers, Albert[1]; Heimicke, Jonas[1]; Hirschter, Tobias[1]; Richter, Thilo[1], Reiß, Nicolas[1], Maier, Alexander[1]; Bursac, Nikola[1]

[1]Karlsruhe Institute of Technology (KIT)
IPEK Institute of Product Engineering
(Albert.Albers, Jonas.Heimicke, Thilo.Richter, Nicolas.Reiss, Nikola.Bursac)@kit.edu
Tobias.Hirschter@partner.kit.edu
Alexander.Maier2@student.kit.edu

Abstract

In the development of mechatronic systems and the associated product services and business models, the system of objectives contains all relevant objectives of various stakeholders, as well as requirements, boundary conditions and interdependencies between objectives. It serves as basis to define the solution area for the technical implementation of products and provides the foundation to validate resulting virtual and physical objects in the process of product development. In particular, the objectives of potential customers and users regarding the product change with an increasing dynamic, which leads to a lack of transparency. Accordingly, the product development process is characterized by constant uncertainties. To be able to counteract these uncertainties, companies are increasingly using agile approaches in order to identify potentially changed objectives at an early stage and to adapt the product along the product development process accordingly. An early validation of prototypes with low functionalities is also carried out under consistent customer integration to identify further objectives early on. As a result, the overall process gains in robustness. One approach that supports the developer throughout the entire development process of mechatronic systems is ASD - Agile Systems Design. It consists of principles, methods and processes of the PGE - Product Generation Engineering and contains a constellation of structuring and flexible elements that can be adapted to the respective development context. To support the process of product development, it is imperative to understand the behavior of systems of objectives in order to be able to derive resulting recommendations for the development. Based on a real development project that has been completed according to ASD, the paper introduces an approach to identify the development of the maturity of the systems of objectives. On this basis, a methodology is presented which supports the agile management of systems of objectives to meet the associated requirements. Thus, a handling of systems of objectives in connection with prototypes generated during the process is presented, which leads to a robust safeguarding of the development direction.

Keywords: Agile product development, ASD – Agile Systems Design, PGE – Product Generation Engineering, system of objectives, customer integration

1 Introduction

Today's markets are characterized by high dynamics (Eisenhardt et al. 2000). New competitors appear on the market, companies offer services within a very short period of time, of which it was generally believed that these companies lack the relevant expertise (amazon wants to offer bank accounts), and customers determine the distribution of market shares by their purchasing behaviour (Athanassopoulos, 2000) and their explicit and implicit demands for products. In addition to the desire for individualisation, the systems of objectives - the totality of all objectives, their interrelationships as well as requirements and boundary conditions - of various customer groups are subject to high dynamics (Albers, Klingler & Ebel, 2013). Even small incidents can have a major impact on the system of objectives and thus on the purchasing behaviour of customer groups, if their systems of objectives are changing by various influences (e.g. Snapchat lost $1.3 billion in market value after Jenner Tweet). These market constellations have a direct influence on the product development process. Only the company, which correctly anticipates the future systems of objectives of its customers and which is able to develop a product today that the customer wants to buy tomorrow, can sustainably operate successfully in the market (Abolhassan, 2016). In addition, the lifecycles of products on the market are becoming shorter and shorter, as companies must include products with unique selling propositions in their portfolios at higher frequencies in order to differentiate themselves from the competition in terms of technology (Kumar & Promma, 2005). This inevitably leads to shorter development times (McGrath, 2012). Clearly, an increasing complexity of the development process can be observed. This is characterized by continuous decision-making in the face of uncertainty (Albers, Ebel, & Lohmeyer, 2012). However, companies are increasingly trying to act flexibly and appropriately in a dynamic development environment by implementing agile approaches also in the field of mechatronic systems development to make the development more robust against changes (Schmidt, Weiss & Paetzold, 2017). The development always focuses on the customer and the user as well as their systems of objectives. Only a product that satisfies all the relevant customer needs has the potential to be successful on the market and thus become an innovation (Albers, Heimicke, Walter, Basedow, Reiß, Heitger, Ott, & Bursac, 2018). This paper uses the concept of innovation based on SCHUMPETER (Schumpeter, 1939): A product is only an innovation if it is successful on the market. In the innovation process, the right product profile (customer-, user- and provider benefit) must be identified, technically satisfied and implemented (invention) and successfully introduced to the market (See Figure 1).

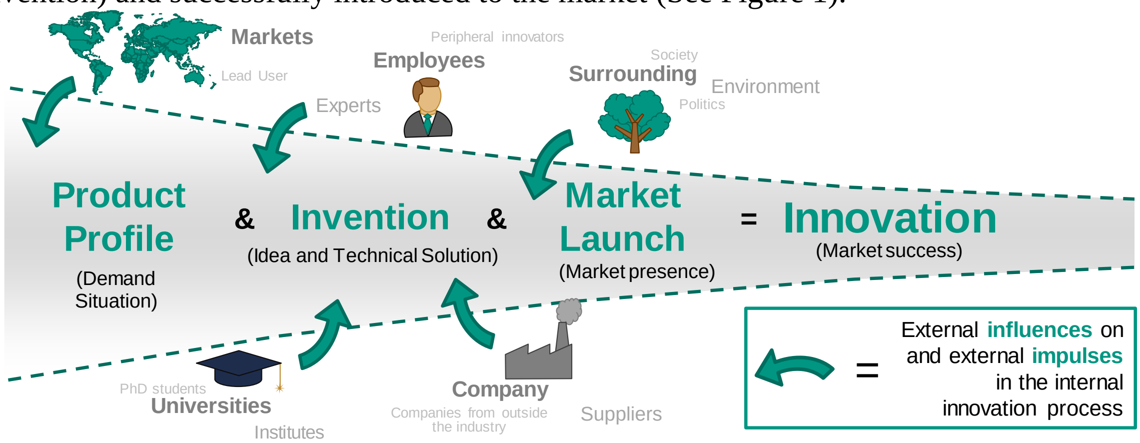

Figure 1: The Elements of an Innovation (Albers et al., 2018)

To anticipate the customers' systems of objectives of the future in the best possible way, a multitude of methods exists (e.g. persona method, sounding board, etc.) in agile approaches. Many of these approaches start with a systematic empathy phase, in which the members of the development team try to put themselves in the customer's shoes (Plattner, Meinel & Leifer, 2011). However, since the system of objectives for a product does not consist exclusively of the condensate of the systems of objectives of different customer and user groups, but contains all relevant objectives of all stakeholders affected by the product (especially the provider himself), conflicts of objectives that need to be resolved regularly occur in projects. Real development projects are characterized by the fact that systems of objectives are continuously concretized by a continuous validation of developed systems of objects through a stringent increase in knowledge (Albers, Behrendt, Klingler, Reiß, & Bursac, 2017). In practice, however, it can be observed that component-oriented development in particular pursues a very static objective management process and thinking in specifications is established. In industries like the automotive industry, for example, the focus is on objects such as the specification sheet, which, however, only represent a shortened representation of the target system at a defined point in time and thus do not serve the continuous tracking of objectives. The handling of systems of objectives is often extremely sensitive due to the large variety of information on the one hand and the lack of relevant information on the other hand. It is difficult to predict the development of systems of objectives that are constantly maturing in each process. The different understanding of the target term in the development team can also result in misunderstandings. Consequently, the question arises as to how an adequate handling of systems of objectives can take place in agile processes. To this end, this paper presents a systematic approach to the investigation of the development of the maturity of systems of objectives in agile projects in practice. In addition, a tool will be presented that goes beyond a conventional, rigid specification and meets the requirements of agile development. The aim is to make it easier to find and understand objectives and to identify the interactions between them quickly and gain robustness according the fulfilling of objectives.

2 State of the art

2.1 Agile processes in the context of PGE – Product Generation Engineering

ASD - Agile Systems Design is an approach to the development of mechatronic systems and the associated product services and business models, derived from empirical observations (Albers, Bursac, Heimicke, Walter, & Reiß, 2017). It bases on the understanding of PGE - Product Generation Engineering, which illustrates the development in the real world through two core theses. There is no development of a new product, that starts on a white sheet of paper but on at least one reference system. The reference system consists of various elements that are used to develop a product generation (e.g. Google Glasses used glasses and Android software as elements of reference products). In addition, the development of new products (product generations) is carried out by systematically combining the three activities of Carryover Variation (CV), Embodiment Variation (EV) and Principle Variation (PV). The sum of the subsystems, which are carried over into the new product generation through EV and PV, describes the respective parts which have to be newly developed (Albers, Rapp, Birk, & Bursac, 2017). Using the knowledge from reference products and previous processes, ASD includes situationally adaptable structuring and agile elements. It is based on 8 principles and can be operationalized by a generic metaprocess (Analyze, Identifying Potentials, Conception, Specification, Realization, Release) and selected methods (See Figure 2). The principles, ASD is based on are: *Consistent unification of the PGE concept and agile development*, *Human-Centeredness*, *Combination of structure and agility appropriate to the respective situation*, *Continuous Validation*, *Thinking in product profiles* (Albers et al., 2018), *Scalable to different kinds of problems and project lengths*, *Situational use of mechanisms of Intentional Forgetting* and *Procedure in product development using the iPeM – integrated Product engineering Model* (Heimicke, Reiss, Albers, Walter, Breitschuh, Knoche, & Bursac, 2018). Product development using ASD makes it possible to display, plan and execute activities iteratively and simultaneously (Albers, Reiß, Bursac, & Richter, 2016). During the process the developers validate the products from the user's point of view, the management from the customer's point of view. Figure 2 shows the ASD with its core elements, whereby it can't be reduced only to the phases. Rather, it represents a situation-specific interaction of various elements. Besides the structuring elements (or main activity cluster) and different principles, ASD provides situation- and demand-oriented provision of design methods to support the developer in his activities (Albers, Reiß, Bursac, Walter, & Gladysz, 2015).

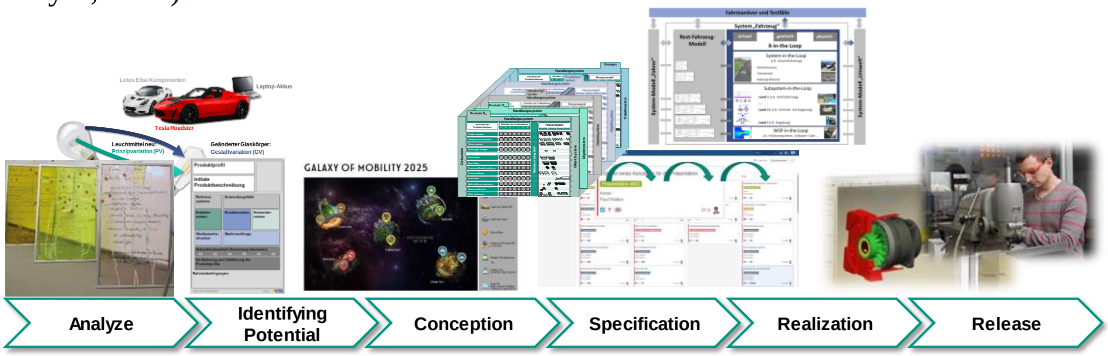

Figure 2: The structuring elements of ASD – Agile Systems Design (Heimicke et al., 2018)

The focus is less on agile project management methods, such as scrum, but rather on agile methods for synthesis (Heimicke et al., 2018) or analysis of systems.

2.2 Objectives in the context of product development

In manufacturing companies, however, objectives of the product interact strongly with a large number of other objects (See Figure 3).

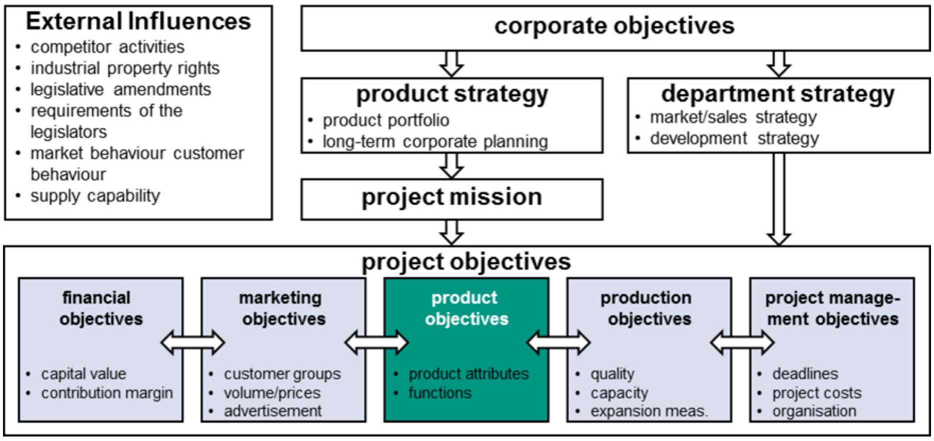

Figure 3: System of objectives in typical product development projects (Bader, 2007)

BADER states that the top-level objectives of a company like *increasing growth* secure the medium- to long-term success of the company. In companies with a diversified product portfolio, various product strategies can be derived from the corporate strategy, which in turn are the starting point to create project orders. Defining the objectives of these projects also contains the consideration of both the different sub-strategies and the project orders of the company's various departments. In accordance to BADER, the objective of the project can be divided into five interacting sub-systems of objectives: Financial, market/sales, product, production and project management objectives. The financial objectives specify both the budget to be used and the revenue to be generated from the product. Market/sales objectives are, for example, the definition of objectives of customers and markets, sales prices or sales figures (Bader, 2007). The different objectives and objective levels listed here are mapped and modeled in the system of objectives.

2.3 Objectives, requirements and boundary conditions

The pertinent literature does not provide clearly separated definitions for the terms *objectives* and *requirements* – in accordance to OERDING the distinction is feasible just to a limited extend (Oerding, 2009). In addition to that, LINDEMANN points to the restrictive role of requirements when searching for solutions and their function of serving a basis of assessment for the selection of a suitable solution concept (Lindemann, 2009). He assigns a decisive role to the development of objectives and the definition of requirements in product development, as wrong or incorrectly defined requirements lead to additional effort in the course of the project and tie up unnecessary resources (Ponn & Lindemann, 2009). COSS define objectives as needs. Therefore, objectives include anything the product is supported to achieve. In contrast to that, the term *requirements* describe anything a product should be capable of (Cross, 2008). Following the findings of EI-LETZ, objectives can be achieved or not be achieved through active action, whereby these are collectively agreed nominal conditions. Requirements, on the other hand, formulate desired facts or properties of the solution (Eiletz, 1999). BADER point out that requirements can be derived from both objectives and boundary conditions (Bader, 2007). LOHMEYER argues that a boundary condition cannot be the responsibility of the development team itself - however, one of the main tasks in the early phase of the development process is to identify the boundary conditions. Nevertheless, these cannot be defined and modified independently. Additionally, POHL provides a distinction between the terms objectives, boundary conditions and requirements: an objective is the earmarked description of a feature of the system or process. Requirements are properties (or conditions) that a system or process must have in order to solve a problem and achieve the objective or meet a standard/norm/contract. A boundary condition is difficult to or cannot be changed at all, but it restricts the system in development (Pohl, 2007). Clearly defining the objectives upfront improves the stakeholders' understanding and acceptance of the overall system, promote the derivation and definition of requirements, support the identification of irrelevant or even incorrect requirements and promote the systematic (resolution) of conflicts (Pohl, 2007). Following EILETZ's example, an objective is the collectively agreed description of a nominal condition that can be achieved by taking action (Eiletz, 2007). A requirement can be derived from both objectives and boundary conditions. Boundary conditions (e. g. technical, organizational or political boundary conditions), on the other hand, are restrictions from the environment of the development project and cannot be influenced. In order to deal with the complexity and dependency on objectives, boundary conditions and requirements as part of the system of objectives, a systemic understanding is necessary. The iPeM - integrated Product engineering Model (Albers, Reiß, Bursac & Richter, 2016) supports operational activities and agile planning of the process of product development. This model is based on the ZHO model according to ROPOHL (Ropohl, 1979) with which the product development is described by three interacting systems: "Through the system of objectives, the operation system and the system of objects product development can be described as the transformation of a (vague at the beginning) system of objectives into a concrete system of objects by the operation system" (Albers et al., 2012). In order to support the developer, the iPeM can be used to model the development process of different product generations, the production system, the validation system and the strategy. This makes it possible to take a holistic view of product development, while at the same time targeting the system in development (Albers et al., 2016). A successful exchange of knowledge between the different stakeholders is necessary to create a holistic, consistent system of objectives. In modern processes, knowledge gain is partly distributed over sub-processes and takes place simultaneously with the participation of various stakeholders from different disciplines. If it is possible to distribute the resulting knowledge or to make it possible to access other people's knowledge, the holistic solution of problems can be supported (Probst, Raub, & Romhardt, 2012). New knowledge is not only generated by individual participants, but also arises from the combination of knowledge of several participants through communication (Moskaliuk, 2008). In order to support the developer in the consistent creation of the system of objectives, specific methods can be helpful. One possibility is a situation- and demand-oriented training concept. This enables the exchange of knowledge about the objectives in the respective subject area. The acquired knowledge about the handling of the system of objectives should enable the participants to exchange knowledge about systems of objectives more effectively in order to synthesize a more consistent holistic system of objectives (Richter et al., 2018). Another tool is the *Pitch 2.0*, which is a framework that combines tools, structures of arguments, teaching presentation skills and recommendations for the transfer of knowledge. This can improve joint creation of systems of objectives, because the concept supports the congruent creation of mental models and enables a transfer of knowledge beyond conventional barriers of communication (Richter, Heimicke, Reiß, Albers, Gutzeit & Bursac, 2018).

3 Research questions and research methodology

The agile strategy of product development described in Chapter 1 is widely used and successfully applied in software development, although it poses several obstacles to the development of mechatronic systems. As this strategy has clear benefits in terms of customer integration and agility compared to the initial strategy, it is intended to be integrated into the development of mechatronic systems despite existing obstacles. To be able to successfully achieve this integration, systems of objectives must also be capable of being mastered in agile development processes. To make systems of objectives controllable in agile processes and thus be able to manage them efficiently, it is necessary to investigate beforehand how they behave in agile processes. To answer the overarching question of behavior and thus also the management of systems of objectives in agile processes, the following research questions were derived:

- How can the maturity of systems of objectives be recorded?
- How does the maturity of system of objectives develop in agile processes?
- How can a modeling of systems of objectives be achieved and supported?

The procedure to answer these questions was: **1)** Based on preliminary work a metric was identified to determine the maturity of systems of objectives. A definition of the maturity of systems of objectives was introduced, which is used for the following research questions. **2)** Through the identified metrics to determine the maturity of systems of objectives, the system of objectives of a real project was observed as a reference for an agile product development process and its development was presented and interpreted. **3)** Based on the findings of research questions 1 and 2, a recommendation for action was made on how the management of agile systems of objectives can be supported. The software tool Scope was developed, which efficiently supports the management of systems of objectives in agile product development processes. Within the project *AIL - Agile Innovation Lab*, a course developed at the *IPEK - Institute of Product Engineering*, a company presents a development task to a team of students. Through methodical support of the *IPEK's* scientists, this task is systematically solved following the ASD developed by ALBERS. Within this real development project, the combination of process, research and product development in the same environment opens a wide range of possibilities for the development team to quickly compile and validate research results and to dynamically and iteratively improve their maturity (Albers et al., 2017).

4 Results

4.1 Metric

The analysis and comparison of the maturity level of systems of objectives implies the presence of a metric to determine these maturity level. Various metrics of this kind are known in literature (Costello & Liu 1995; Davis et al., 1993), which have been analyzed and tested whether they are practicable for this application. During this analysis, the following conclusions were drawn as to how a practical and correct metric for determining the maturity of systems of objectives in agile processes must be designed: **1)** Metrics used in practice to determine the maturity of systems of objectives rely on a quantitative comparison of previously defined objectives with the total number of objectives in the system of objectives. **2)** Objectives have different degrees of maturity and therefore do not have a uniformly large influence on the progress of maturity development, therefore objectives with a uniform degree must be defined in order to achieve meaningful results through a quantitative approach. The prerequisites of a metric identified above imply the need for a uniform target framework and a uniform representation of objectives, which enables a uniformly pronounced degree of maturity of the individual objectives. To develop such a target framework, insights from agile processes of software development and insights from research on design methods were identified and harmonized.

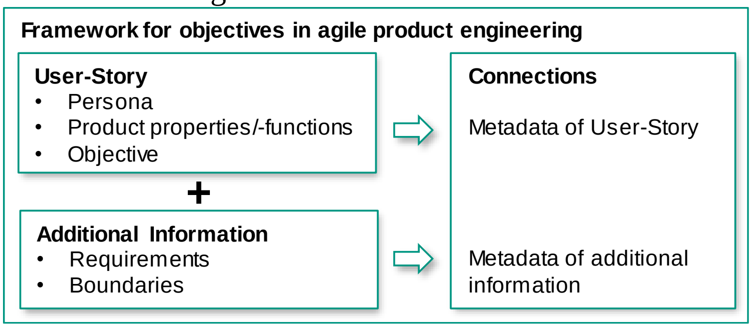

Figure 4: System diagram of the Framework for objectives in agile product engineering

Figure 4 shows this framework in the form of a system image. If all objectives are documented in a development project according to this framework, all objectives have the same degree of maturity and are also described at the same hierarchy level. Therefore, they are standardized to the growth of maturity of the system of objectives. Accordingly, the degree of maturity of a system of objectives can be determined with the following metric developed in this work:

If all the objectives considered and the requirements and boundary conditions associated with them have the same degree of maturity, the maturity of a system of objectives can be determined using the following formula:

$$Maturity\ (t)[\%] = \frac{\sum_{i=0}^{i=t} Objective_i}{\sum_{j=0}^{j=tmax} Objective_j} * 100\%$$

The sum in the numerator represents the number of objectives at time t and the sum in the denominator represents the number of objectives at the end of the project. Therefore the percentage maturity level of a system of objectives can be determined at a point in time t.

4.2 Description of the Observations

Using this metric, the development process of *AIL* was examined and the development of the maturity of the corresponding system of objectives was presented. For this purpose, the objectives were recorded and collected during the project in accordance with the developed framework of objectives in order to show the development of the maturity level over the duration of the project. With help of the later described software-tool *Scope*, each objective was documented according to the developed framework and saved in a database. Additional to the information about the objective itself, each objective was marked with the date and project phase when it was added to the system of objectives. Under these two premises, objectives with the same level of maturity and a logged date of each objective, the software-tool *Scope* allowed an automatic analysis of the maturity level over the duration of the project. The project team of *AIL* consisted of four developers from which one of them was the manager of the system of objectives. He was responsible for logging the objectives and communication each change or new objective to the other team members. Therefore, the risk of misunderstandings with the project partner that could lead to a definition of wrong objectives was minimized. Figure 5 shows the development of the maturity of the system of objectives in *AIL* with three characteristics: leaps in maturity at project milestones, increasing maturity during the project phases and the approximation of maturity to a saturation limit. The leaps can be traced back to the fact that a particularly large number of new objectives was included in the system of objectives at the milestones. This high number over a short period of time can be explained by the validation of the system of objects in cooperation with the customer. This validation consisted during the *AIL* project, in accordance to other agile processes and the extended ZHO model (Albers et al. 2012), of the construction of a prototype which represents the actual functions of the product and a demonstration of the prototype together in collaboration with the customer (Matthiesen, Grauberger, Nelius, & Hölz, 2017). By a tangible interaction of the future product, the communication of the development team with the customers took place on a completely different level. This enabled the developers to communicate the current system of objectives to the customers and enabling the customers to associate further objectives and needs that they expect in the future product. In addition to new objectives, however, the project also had objectives that became obsolete, especially after milestones.

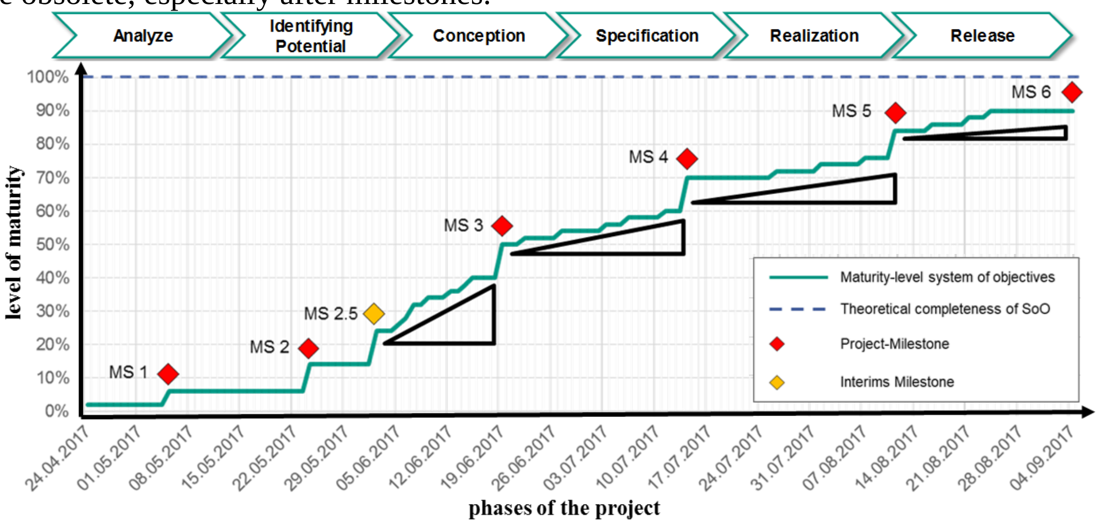

Figure 5: Development of the maturity-level of the system of objectives during the AIL project

The team had to remove these from their minds through mechanisms of Intentional Forgetting (Schüffle et al. 2017). The second characteristic, the continuously increasing maturity level during the project phases, is based on changing targets and adding new targets to the system of objectives through validation activities of the development team. The continuous validation of the system of objectives in accordance to the extended ZHO model occurs parallel to the development process. Similar to the validation at the project milestones, the development team continuously carried out analysis and synthesis activities between system of objectives and object system. As a result, errors are discovered, potentials are uncovered or new objectives are associated and this is reflected in an increase in the maturity level of the system of objectives during

the project phases. The level of maturity increases in the later phases of the project, both at the project milestones and within the project phases. This can be explained by continuous detailing of the solution space during the process. In later project phases, the system of objectives has thus reached a sufficient level of detail and the number of new targets decreases.
With the findings of this analysis and the identified characteristics of systems of objectives in an agile environment the software tool *Scope* was developed.

4.3 Systematic and agile Approach for the Modelling of Systems of Objectives

Based on the analyzed behavior of systems of objectives in agile processes, requirements for a management tool for a system of objectives were identified for such an application. These requirements were implemented in the *Scope* software tool, developed and validated in *AIL*. *Scope* is a VBA-based tool, which supports managing systems of objectives with the following functions: Guidance on documentation of targets, storing targets, linking them, displaying the system of objectives, filtering targets by specific tags, and sorting targets by importance. An important focus of *Scope* is to enable a practicable way of linking of objectives. This is achieved through assigning specific tags to each objective and thus enabling *Scope* to create a network of objectives, and the developer to use this network in further steps of the development process. The validation of the software tool *Scope* was carried out within the framework of the Live-Lab *AIL*, which was optimally suited for a validation, due to the implementation of an agile product development process. In analogy to the agile processes in *AIL*, *Scope* was developed agile, through several iteration stages and continuous validation. According to the extended ZHO (Albers et al. 2012) model, an iteration consisted of the identification of a required function, the implementation of this function, the reanalysis and an adaptation or modification of this function. In this way, *Scope* was developed iteratively and due to the continuous validation, only functions were implemented which create the greatest benefit for the manager of the system of objectives.

5 Findings

The contribution has improved the understanding and handling of systems of objectives in agile processes. This is achieved by means of a clear systematic, which makes objectives comparable through a uniform and simple description. In addition, interactions between objectives can be continuously registered and checked by tagging, which requires little effort. In this way, it is also possible to filter out these interactions easily, so that all the conflicting objectives can be displayed directly and taken into account. It could be shown that successful product development doesn't necessarily need a thick requirement specification, but can be supported with a light and flexible tool. Nevertheless, a clean documentation of objectives is still essential for success in the process. Particularly at the milestones it could be demonstrated that the customer's interaction with different prototypes increases the associations for new objectives, or the adaptation of obsolete objectives in the course of validation. Through mechanisms of the ASD - Principle Intentional Forgetting, these objectives were hidden in *AIL* according to the situation and well-founded, in order to focus on other, for the respective situation more relevant knowledge. The aim of this contribution was not to establish a connection between the recording of an objective and the implementation of objectives with regard to the timing in the process.

6 Future Works

The approach was developed and validated as part of *AIL* and made a significant contribution to the understanding of systems of objectives and how they evolve during development processes. This understanding is necessary to be able to (further) develop and optimize methods, which concern in particular the development of systems of objectives. In observational studies in mechanical and plant engineering as well as in the automotive industry it was observed that changes to systems of objectives in conventional processes show similar sharp increases – however milestones are not the only trigger for this. Nonetheless, sharp increases in systems of objectives are also strongly correlated to decision-making rounds within the top management. In order to improve the management of systems of objectives in conventional processes, it is necessary to use the method in a longer process (development of a vehicle). In addition, it is necessary to extend the developed method with regard to the hierarchization of objectives in order to understand, e.g. the effects of changes to objectives at subsystem levels on the overall system. Due to the constant increase in complexity, e.g. in the development of vehicles, it is necessary to master systems of objectives along the entire product development process in order to be able to develop the right product according to customer requirements. In addition, the principle of Intentional Forgetting can be anchored even more purposefully in the development process of ASD in order to remove further, obsolete objectives from the development focus.

References

Albers, A., Behrendt, M., Klingler, S., Reiß, N., & Bursac, N. (2017). Agile product engineering through continuous validation in PGE – Product Generation Engineering. Design Science, 3(16).

Albers, A., Bursac, N., Heimicke, J., Walter, B., & Reiß, N. (2017). 20 years of co-creation using case based learning: An integrated approach for teaching innovation and research in Product Generation Engineering. In M. E. Auer, D. Guralnick, & J. Uhomoibhi (Eds.), Proceedings of the 20th ICL Conference. Springer.

Albers, A., Heimicke, J., Walter, B., Basedow, G. N., Reiß, N., Heitger, N., Ott, S. & Bursac, N. (2018). Product Profiles: Modelling customer benefits as a foundation to bring inventions to innovations. Proceedings of CIRP Design 2018.

Albers, A., Klingler, S., & Ebel, B. (2013). Modelling systems of objectives in engineering design practice. International Conference on Engineering Design, ICED13, Sungkyunkwan University, Seoul, Korea.

Albers, A., Ebel, B., & Lohmeyer, Q. (2012). Systems of objectives in complex product development. Proceedings of TMCE 2012.

Albers, A., Rapp, S., Birk, C., & Bursac, N. (2017). Die Frühe Phase der PGE – Produktgenerationsentwicklung. Stuttgarter Symposium für Produktentwicklung, SSP2017.

Albers, A., Reiß, N., Bursac, N., & Richter, T. (2016). iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering. Procedia CIRP, 50, 100–105.

Albers, A., Reiß, N., Bursac, N., Walter, B., & Gladysz, B. (2015). InnoFox – Situationsspezifische Methodenempfehlung im Produktentstehungsprozess. In H. Binz et al. (Eds.), Proceedings of Stuttgarter Symposium für Produktentwicklung.

Abolhassan, F. (2016). Was treibt die Digitalisierung?. Springer Fachmedien Wiesbaden.

Athanassopoulos, A. D. (2000). "Customer satisfaction cues to support market segmentation and explain switching behavior." Journal of business research 47(3), pp. 191-207.

Bader, J. (2007). Die Hauptstellgrößen des Produktzielmanagements am Beispiel der Automobilentwicklung. Fakultät für Maschinenbau, Technische Universität Dresden.

Costello & Liu. (1995). Metrics for requirements engineering. In: Journal of Systems and Software 29, Nr. 1, 39–63

Cross, N. (2008). Engineering Design Methods – Strategies for Product Design. 4. Edition, Wiley,New Jersey.

Davis, A., Overmeyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Lederboer, G., Reynolds, P., Sitaram, P., Ta, A. & Theofanos, M. (1993). Identifying and

measuring quality in a software requirements specification. In: Proceedings of the First International Software Metrics Symposium.
Eiletz, R. (1999). Zielkonfliktmanagement bei der Entwicklung komplexer Produkte – am Beispiel PKW-Entwicklung, Fakultät für Maschinenbau, Rheinisch-Westfälische Technische Hochschule Aachen, Shaker Verlag.
Eisenhardt, K. M., & Jeffrey A. M. (2000). "Dynamic capabilities: what are they?." Strategic management journal, pp. 1105-1121.
Heimicke, J., Reiss, N., Albers, A., Walter, B., Breitschuh, J., Knoche, S., & Bursac, N. (2018). Agile Innovative Impulses in the Product Generation Engineering: Creativity by Intentional Forgetting. *Proceedings of ICDC18*.
Kumar, S., & Promma, P. (2005). Research methodology. Springer US.
Lindemann, U. (2009). Methodische Entwicklung technischer Produkte. 3. Aufl., Springer, HD.
Matthiesen, S., Grauberger, P., Nelius, T. & Hölz, K. (2017). Methodische Unterstützung des Erkenntnisgewinns in der Produktentwicklung durch Konstruktionshypothesen. KIT Scientific Working Papers, Karlsruhe: KIT Scientific Publishing
McGrath, M. E. (2012). Setting the PACE in product development. Routledge.
Moskaliuk, J. (2008). *Konstruktion und Kommunikation von Wissen mit Wikis: Theorie und Praxis*, *Web 2.0*. Boizenburg: vwh, Hülsbusch.
Oerding, J. (2009). Ein Beitrag zum Modellverständnis der Produktentstehung – Strukturierung von Zielsystemen mittels C&CM. Fakultät für Maschinenbau, Universität Karlsruhe.
Plattner, H., Meinel, C., & Leifer, L. (2011). Design Thinking: Understand – Improve – Apply. Berlin, Heidelberg: Springer Berlin Heidelberg.
Pohl, L. (2007). Requirements Engineering – Grundlagen, Prinzipien, Techniken. dpunkt Verlag, Heidelberg.
Ponn, J.; Lindemann, U. (2011). Konzeptentwicklung und Gestaltung technischer Produkte – Optimierte Produkte: Systematisch von Anforderungen zu Konzepten und Gestaltlösungen. 2. Auflage, Springer Verlag, Berlin.
Probst, G. J. B., Raub, S. P. & Romhardt, K. (2012). Wissen managen: wie Unternehmen ihre wertvollste Ressource optimal nutzen. 7. Auflage. Aufl. Wiesbaden : Springer Gabler.
Richter, T., Albers, A., Birk, C., Rapp, M. & Bursac, N. (2018). A Training Framework for the Synthesis of a Consistent System of Objectives in Modular Design; 28th CIRP Design Conference, Nantes, France.
Richter, T., Heimicke, J., Reiß, N., Albers, A., Gutzeit, M. & Bursac, N. (2018). Pitch 2.0 – Concept of early evaluation of product profiles in product generation engineering. TMCE 2018. Las Palmas, Spain.
Ropohl, G. (1979). Eine Systemtheorie der Technik: zur Grundlegung der allgemeinen Technologie.
Schmidt, T. S., Weiss, S., & Paetzold, K. (2017). Agile Development of Physical Products: An Empirical Study about Motivations, Potentials and Applicability. University of the German Federal Armed Forces.
Schüffler, A., Schwier, I., Kluge, A., & Gronau, N. (2017). Organisationsentwicklung: Gewolltes Vergessen als Jungbrunnen. SW Sozialwirtschaft, 27(2), pp. 32-33.
Schumpeter, J. A. (1939). Business Cycles. A Theoretical, Historical and Statistical Analysis of the Capitalist Process., McGraw-Hill, New York