Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 60 (2017) 458 – 463

www.elsevier.com/locate/procedia

27th CIRP Design 2017

Technical change management for the maintenance of product platforms

Günther Schuh[a], Michael Riesener[a], Christian Tönnes[a,*], Sasa Aleksic

[a] Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Steinbachstraße 19, 52074 Aachen, Germany

* Corresponding author. Tel.: +49-241-80-20617; fax: +0-49-241-80-620617. *E-mail address:* c.toennes@wzl.rwth-aachen.de

Abstract

In order to react to an increasing global competition on the one hand and an increasing demand for individual products on the other hand, product platforms became a well-established design concept for developing product families. By using the principle of modularization companies are able to achieve synergies within development and production and at the same time offer a wide variety of products. Therefore, approaches and methods for the initial setup of modular product architectures are well covered in the scientific literature. Nevertheless, a lack of research can be identified for the sustainable maintenance and continuous development of product platforms. Accordingly, most companies do not have a structured release management for the systematic introduction of technical changes and modified modules of physical products. In order to preserve the success of an implemented product platform, it is mandatory to plan the introduction of changes. In this paper, a new method is introduced for synchronizing the lifecycle of modules with the lifecycle of the respective platform products. The goal of this approach is to maximize the number of technical changes which can be implemented into products of an existing modular product platform at a certain time while minimizing development costs as well as changing efforts within production. At the end strategies are shown how to release physically changed modules into different product lines of a modular product platform.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference

Keywords: modular product platforms; lifecycle synchronisation; release strategies; technical change management; release management

1. Introduction

Nowadays, manufacturing companies are facing two major challenges concerning the development of new products [1]. On the one hand there are fast evolving technologies and decreasing product lifecycles, while on the other hand customers are expecting highly individualized products [2, 3]. Especially competitors from low wage countries force European companies to offer cost-efficient products with a high consideration of individual customer requirements [4, 5]. This leads to an increasing variety of external products as well as internal processes [5, 6]. One solution to handle this dilemma are product platforms [7]. They permit realizing economies of scale due to a repetition of modules or functions as well as economies of scope in terms of customer specific products [8]. As a study conducted by the WZL of RWTH Aachen University shows, many companies are already using a product platform and more than 40% of these companies are in a stage of advanced implementation [9]. Once a product platform is implemented it is not a static structure but a dynamic system which underlies changes in terms of upgrades or technical developments during the lifecycle[10]. Nevertheless, most companies do not follow a defined process to implement changes into an existing product platform [9]. So the question of handling technical changes is becoming more important. In context of release management detailed knowledge of change requests as well as relationships between physical modules have to be used for computing an optimal composition of new or changed modules which can be released at a certain point. Thus, this paper introduces a new method for synchronizing the lifecycle of modules with the lifecycle of physical products which are based on a common modular product platform. Strategies are shown how to release changed modules into different product lines of a modular product platform.

2212-8271 © 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
doi:10.1016/j.procir.2017.01.025

2. Related Work

As it was outlined in the introduction, there is a strong demand for developing a method for systematically introducing technical changes in product platforms [11, 12]. Research approaches of this topic are published under the term of release engineering or release management [12].

According to Belener, there are two different reasons why technical changes occur during the product platform's life cycle [13]. The first reason is a technical change due to an improvement of the platform. The second trigger of changing requests is an error which can be divided into two different sub categories. On the one hand a module was not defined correctly during the concept phase, while on the other hand a module was not designed according the specification sheet. Aßmann defines every change during the phase of development as an iteration [10]. Whereas technical changing requests after start of production are called updates.

After defining reasons for changing requests and different types of technical changes, there is still the need for their systematical introduction into existing product platforms. In information technology the term "release" is defined as the "collection of one or more new or changed configuration items deployed into the live environment as a result of one or more changes" [14]. A similar definition is used by Stahlknecht and Hasenkamp, who describe a release as a periodical revision, improvement or expansion of software [15]. Also defining standardized timing and prioritizing tools are part of the research in that field [16–18].

Using this understanding release management can be defined as the process of planning, timing and controlling releases in test as well as live environments [16]. Research in this field focuses on the optimal point of market entrance of new variants. Thereby is the conflict between time consuming testing and introducing products to the market as early as possible to realize advantages against competitors [19]. That is why it is crucial to determine the earliest date of release while considering technical maturity and benefit for the customer.

The approach developed by Schuh aims at consolidating all technical changes into. Due to this synchronized realization, companies are able to organize and optimize their internal development and production processes. Furthermore, it can be used for a dynamic adoption of competition and market conditions [20]. Within the methodology technical changes are classified considering whether the chances are time sensitive, avoidable or necessary changes. For every cluster a specific process and timeline is defined. Criteria for evaluating changes are implementation costs, urgency and sustainability [21].

Carlshamre defines requirements for a certain release and provides an automated algorithm which selects certain customer requirements that have to be fulfilled at a certain time. Furthermore, Carlshamre considers the level of quality that should be realized. As many of this requirements are dependent on each other, the process of selection is a complex problem [17].

The recent approach by Wright uses eight case studies for describing release processes in the software industry. After analyzing faults within these processes Wright proposes four hypotheses for describing the essentials of successful release management [18].

Concluding it can be said that these approaches focus on releasing more than one product. Nevertheless, Schuh, Carlshamre and Wright lack on considering lifecycles for single modules. While Schuh and Wright take interactions between modules into account, this aspect is missing in Carlshamre's approach too. Also Carlshamre as well as Wright do not classify different types of changes. None of the approaches above considers different strategies for releasing technical changes in products. In conclusion, there is no method to determine a release strategy for technically changed modules of physical products which are based on a modular product platform taking the change efforts as well as the benefits into account. Such a methodology is needed as studies show that industry do not have established any processes for introducing technically changed physical modules into existing modular product platforms [9]. Furthermore Lindemann et al. showed that most technical changes are not time sensitive so that change requests can be bundled and released at an optimal date [22].

3. Technical change management for the maintenance of product platforms

To fill the gap of identified shortcomings a new method for the maintenance of modular product platforms will be introduced. This method contains three steps to synchronize the lifecycle of modules with the lifecycle of the respective platform products systematically. First, the flexibility of a module of the product architecture needs to be analysed. Next, efforts and benefits of changing requests need to be quantified, in order to identify the optimum bundle of changing requests. Last, a release strategy for the introduction of the changes needs to be chosen. Following, all steps will be explained using an industry case study. Therefore, a hydraulically released brake from an industry project will be analysed. The break consists of ten parts which are shown in figure 1.

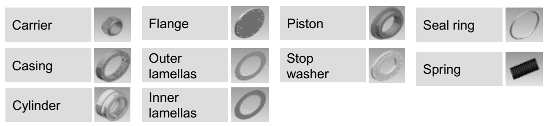

Fig. 1: Parts of hydraulically released brake (based on [23])

The proposed method is based on a systematic analysis of relationships between modules of a modular product platform. The central part is the identification optimal bundle of technical changes which is an optimization problem. This is solved by algorithms for a knapsack problem with restricted capacities.

3.1. Flexibility of modules

First of all, it is essential to analyze the current state of the product structure for future planning and upcoming technical changes. Therefore, an onion peel model is used to visualize the overall flexibility of a module and its direct correlation to release frequencies. For an objective and quantitative measurement, the flexibility of modules is evaluated in terms of four dimensions which are:

- customer features,
- interactions between modules,
- complexity costs and
- technology roadmap

The first dimension analyzes the market demands and customer requirements. In order to identify the impact of features the Kano model is a well-established method for clustering customer perception and is used to classify three types of requirements [24]. The first type, must-be requirement, fulfills the basic features and can lead only to non-dissatisfaction but is not able to enthuse the customer. In contrast to the must-be requirement, a one-dimensional requirement shows a linear proportionality between customer satisfaction and degree of fulfillment. This means, that the second type has either the possibility to satisfy or dissatisfy the customer. The so called attractive requirements generate customer enthusiasm and lead to a high customer satisfaction if fulfilled. By aid of the Kano model it is possible to evaluate the flexibility, as modules that address requirements with lower perception and enthusiasm have the potential for standardization.

The second dimension evaluates the interactions between modules, as modules are used in compositions for different products. There are mainly two types of interactions. On the one hand a module can affect others, on the other hand a module can be influenced by other modules. By analyzing all active and passive interactions the strength of dependency can be considered. With the overall sum of active and passive weighted interactions of a module, one can draw conclusions about its flexibility. The higher the score of active interactions, the higher the influence of a technical change of this module and thus lower is the flexibility.

The third dimension quantifies the effort due to a technical change within a module. Thus modules, that cause high costs due to technical changes, are ranked with low flexibility.

The last and fourth evaluation dimension regards the period, in which the to be changed module will be replaced by a new technology. In order to obtain information about technology replacement a technology roadmap is useful. As a result, it can be concluded that modules with a forthcoming and urgent technology replacement have to be more flexible than modules, whose replacement does not take place in near future.

As mentioned above, the onion peel model is used for visualizing the flexibility of modules. Therefore, the flexibility of each module is ranked from high to low by each of the four dimensions. For calculating the final degree of flexibility a weighting factor can be used [25]. This factor describes the position of a module within the onion peel model (Fig. 2).

Modules placed in the center of the model have a low flexibility, whereas modules which are located on the outer peel have higher flexibility. Based on the classification and position of modules in the onion peel model release cycles can be derived for each module. Modules which show a high flexibility (outter peel) can be changed easily such as the seal ring, cylinder or flange.

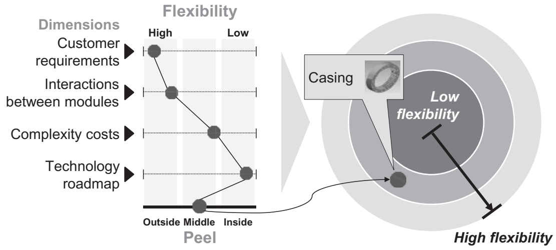

Fig. 2: Onion peel model

This means they have a high frequency of possibilities for being implemented. Conversely, this means that modules located within the innermost peel of the model should rarely be changed which are the inner and outer lamella or the carrier. Those modules have the slowest release frequency. As a consequence, there are two basic types of releases. During “major releases” modules with the lowest flexibility and modules with a high flexibility can be implemented. Whereas “minor releases” have a much higher frequency and only implement less complex technical changes. In order to minimize the efforts of testing changed modules all frequencies have to be multiples of each other (Fig. 3).

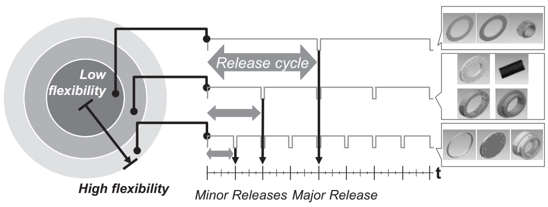

Fig. 3: Relationship between flexibility of modules and release cycles

3.2. Identification of optimum bundle of technical changes

As a second step, a mathematical algorithm is used to identify the optimum composition of modules to be changed during certain releases. Therefore, every technical changing request has to be measurable, representative and comparable. The benefit is quantified as a cost reduction or cost decrease in the production process. Whereas effort is measured in terms of necessary capacity for the development, like man-hours and needed machine tools, due to technical changes on modules. In figure 4 four exemplary change requests are listed. The shown effort and benefit values were determined within workshops with designers and production engineers. Thereafter, the algorithm is used to identify which composition of technical changing requests has the most benefits while triggering minimal efforts.

3.2.1. Pareto-optimal changing cluster

At first, all possible compositions of technical changing requests are formed and evaluated according to the Pareto-method in order to form release bundles. Therefore, every combination of changing requests is valuated by adding the benefits and efforts of the certain composition and comparing the overall score with scores of other compositions.

Compositions, whose overall benefits can not be realized with lower overall efforts, are defined as Pareto-outcomes and visualized as a binary vector called changing vector (Fig. 4) [23].

		Change of casing size	Design of tooth spaces	Design of spline geometry	Design of seal ring grooves	...
Effort	Benefit	CR1	CR2	CR3	CR4	CR5
0	0	0	0	0	0	0
5	2	1	0	0	0	0
4	3	0	1	0	0	0
9	5	1	1	0	0	0
3	4	0	0	1	0	0
8	6	1	0	1	0	0
7	7	0	1	1	0	0
12	9	1	1	1	0	0
⁞	⁞	⁞	⁞	⁞	⁞	0

Key: E= Effort; B= Benefit; CRx = changing request x

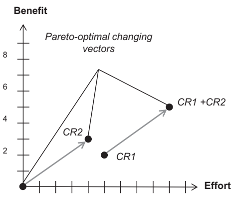

Fig. 4: Pareto-optimal changing cluster visualized as changing vector

3.2.2. *Potential matrices of changing cluster*

Next, the potential matrix of each changing cluster is set up in order to calculate the overall effort and benefit of very changing cluster in chapter 3.2.5. Therefore, each changing vector is multiplied with its transpose to get the symmetrical n×n potential matrix, which contains only the binary numbers 1 and 0. Fields of the potential matrix are overlaps of rows and columns. As shown in Fig. 5 composition "A" consists of changing request 3 and 4 which means, that a redesign of the spine geometry and seal ring groves should be conducted together.

Effort	Benefit	CR1	CR2	CR3	CR4	CR5
⁞	⁞	⁞	⁞	⁞	⁞	0
7	7	0	1	1	0	0
12	9	1	1	1	0	0
2	5	0	0	0	1	0
5	9	0	0	1	1	0 (A)
9	12	0	1	1	1	0 (B)
14	14	1	1	1	1	0

A:

$$\begin{pmatrix}0\\0\\1\\1\\0\end{pmatrix}(0 \quad 0 \quad 1 \quad 1 \quad 0) = \begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&1&1&0\\0&0&1&1&0\\0&0&0&0&0\end{pmatrix}$$

B:

$$\begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix}(0 \quad 1 \quad 1 \quad 1 \quad 0) = \begin{pmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&0&0&0&0\end{pmatrix}$$

Key: E= Effort; B= Benefit; CRx = changing request x

Fig. 5: Potential matrix

3.2.3. *Identification of interactions between technical changes*

Due to the fact, that each changing request influences different modules by its implementation, the interactions between different changes need to be considered as well.

First, a matrix is set up, that shows which modules are affected by a technical change request by the numbers "1" and "0". When evaluating the degree of interaction, the scalar product of two row vectors is used. It is called the technical-change-coupling-factor (TCCF) and shows how many common modules are affected by two changing requests. As an example, changing request 2 (*Design of tooth spaces*) and 4 (*Design of seal ring grooves*) influence component 1 (*seal ring*) and 3(*outer lamella*). Thus the number of common affected components is 2 and equates to the scalar product of changing request vector 2 and 4. A TCCF of 0 means that the respective changing requests do not influence common modules as it would be for changing request 1 and 5 (Fig. 6).

	Seal ring	Inner lamella	Outer lamella	Cylinder	Casing
	M1	M2	M3	M4	M5
CR1	0	1	1	0	1
CR2	1	0	1	0	1
CR3	0	0	0	0	1
CR4	1	0	1	1	0
CR5	0	0	0	1	0

$$\begin{pmatrix}1\\0\\1\\0\\1\end{pmatrix} \cdot \begin{pmatrix}1\\0\\1\\1\\0\end{pmatrix} = 2$$

Key: CRx = changing request x; Mx = Module X

Fig. 6: Technical-change-coupling-factor (TCCF)

3.2.4. *Calculation of adjusted efforts and benefits*

In the next step, the efforts and benefits of each defined changing cluster need to be adjusted. This step is necessary as the efforts might decrease and the benefits might increase when two changing requests are executed in a bundle.

For this, two matrices are used (Fig. 7). Both matrices consist of change requests as rows and columns. The numbers quantify the influence of the implementation of one changing request on the other request(s). The TCCF supports this procedure by showing which fields in the matrix need to be considered. But the actual values need to be estimated by experts of different departments (Fig. 7). Attention should be paid to the effort matrix, because all fields have to be filled with negative values. The negative sign must be interpreted as a reduction of effort if two changing requests are proceeded at a time. For example, a value of "-3" means that the composition of the two influencing changing requests reduces the effort in amount of 3, whereas a benefit value of 6 increases the benefit in amount of 6.

Effort matrix

	CR1	CR2	CR3	CR4	CR5
CR1		0	0	-1	-1
CR2			-1	-3	0
CR3				-1	0
CR4					-1
CR5					

Benefit matrix

	CR1	CR2	CR3	CR4	CR5
CR1		1	2	2	7
CR2			3	6	0
CR3				1	0
CR4					4
CR5					

Key: CRx = changing request x

Fig. 7: left: Adjusted effort matrix; right: Adjusted benefit matrix

3.2.5. *Calculation of optimal composition of changing requests*

Based on the adjusted effort and benefit matrix as well as the potential matrix, the new efforts and benefits of each changing request can be calculated. Therefore, the effort matrix and the benefit matrix are multiplied on a per-element basis by the potential matrix. Finally, the value of benefit and effort of each Pareto-optimal changing request composition is available (Fig. 8). The value "-1" means that the effort of implementing the shown bundle changing requests is reduced by 1 compared to executing them separately. In that case dependencies between the casing which is affected by CR 3 and the outer lamella which is affected by CR 4 leads to a lower effort performing both changing request at a time instead of implementing them separately.

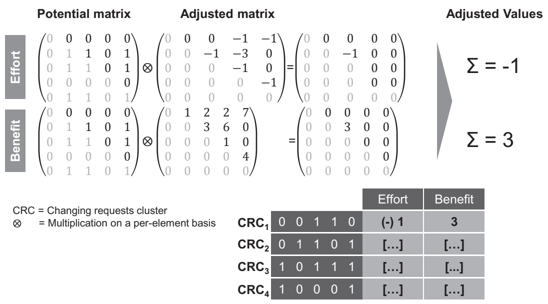

Fig. 8: Potential matrix multiplied by potential effort and benefit matrix

Finally, the optimum composition of changing requests for the next release can be identified. Therefore, all Pareto-optimal changing clusters have to be adjusted and visualized as scenarios in graphics, plotting effort on the abscissa and benefit on the ordinate (Fig. 9). The capacity of resources is used to create an upper limit for needed effort [23]. All changing requests located left to the value of maximum effort can be realized which means those changing requests are the optimal composition for the next release.

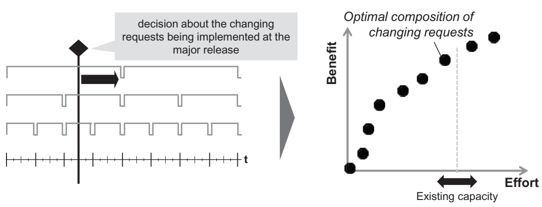

Fig. 9: Decision about composition of changing requests

3.3. Scenarios of implementation

After defining an optimal composition of technical changes as well as its release cycle the next challenge is to synchronize it with the regular release cycle of new products of the modular product platform. As 73% of technical changes are not time sensitive, they can be bundled and implemented in consolidated releases [22]. Therefore, different strategies can be used to introduce new product features in form of modules into an existing product portfolio. The automotive industry usually uses a “trickle-down”-strategy. That means that new features are first implemented in their most expensive product line. Afterwards those technologies are integrated into other product lines step by step [26]. In general, there are two dominating strategies for releasing new modules into a product environment: An ad-hoc implementation or different degrees of time lag: [23]

3.3.1. Ad-hoc Implementation

An ad-hoc implementation means that the bundle of change requests will be implemented at the moment as they are completely developed and in all products of a product platform at one time (Fig. 10). There is no consideration of lifecycles or product strategies. That’s why an ad-hoc implementation usually contradicts any orderly management [11].

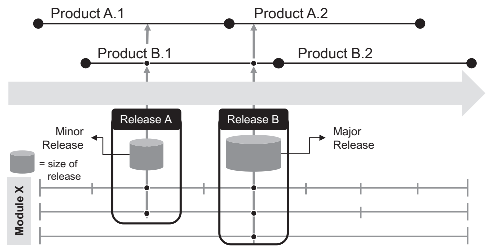

Fig. 10: Ad-hoc Implementation [25]

But in cases of safety or quality issues of a product it is absolutely essential to release the optimized modules to the market even though there are huge costs of change. Another challenge occurs in sales where new products features have to be explained towards the customer without much time in advance. The ad-hoc implementation can also be used to avoid a debased position in competition. By an immediate synchronization of changed modules and the product life cycle enormous advantages can be realized in case of a sudden change of customer.

3.3.2. Gradual Implementation

A gradual implementation is more appropriate concerning a well-structured release management. It is characterized by the consecutive introduction of the modules into the product portfolio following the defined lifecycle of each product (Fig. 11). This strategy allows to realize certain advantages such as low costs of change as not all products of one platform have to be changed at a time and the capacity of utilization is well balanced. In addition, a gradual implementation saves money because of lower overhead costs, e.g. lower expenses for testing and quality management. But integrating the changed modules with a time lack implies also disadvantages, e.g. some products are exposed to a delay of innovation. There is also a decrease in economies of scale as two different variants of one module are produced for a certain time.

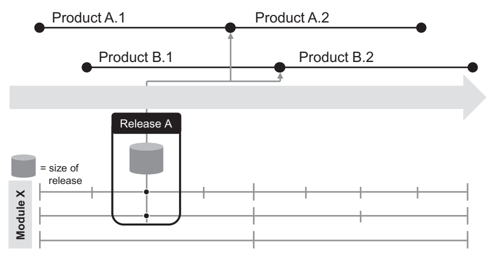

Fig. 11: Gradual Implementation [25]

3.3.3. Change of product release

The last strategy is characterized by an early release of a new product variant. This strategy is close to a technology push as a new or changed module triggers to preponing the market introduction of new product variants (Fig. 12). A reason for this strategy is an avoidance of falling behind competitors through missing out on innovation. Also, companies are not losing any economies of scale as the changed module is implemented directly. Costs of changing for this module are comparatively low as there are enough reserved capacities for the product release. Nevertheless, there are also disadvantages such as e.g. unplanned usage of resources that might lead to dissatisfaction among employees or the reduction of the lifespan that leads to financial burdens as amortization periods change.

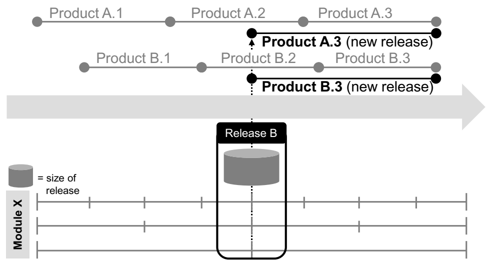

Fig. 12: Change of product release [25]

4. Conclusion

Surveying related work in the field of release management, release engineering and technical change management as well as related studies showed the initial necessity for a method addressing the systematical maintenance of product platforms. The proposed method is able to optimize the composition of modules for a certain release into the product platform. It is capable of determine the flexibility of all modules through one value and allows the matching of modules to different release cycles. Furthermore, an algorithm is presented to calculate the optimal bundle of technical changes that can be implemented at a time. Finally, different strategies for implementation are shown. Thus, the methodology addresses the research gap and aims at researchers as well as practitioners in the industry

Further research is demanded for establishing a link to the requirements management to identify the change requests which trigger new modules. In that way the causes and urgency of changes can be tracked back and analyzed. Furthermore, release management needs to be embedded into the context of consistent data connections of PLM systems.

Acknowledgements

The presented results have been developed within the research project "Development of a methodology for the systematic planning and development of product architectures through release management" funded by the Deutsche Forschungsgemeinschaft (DFG).

References

[1] Eversheim W., Editor. *Innovation Management for Technical Products*. Springer Berlin Heidelberg, 2009.

[2] Schuh G., Arnoscht J., Bohl A., Nussbaum C. Integrative assessment and configuration of production systems. CIRP Annals - Manufact. Tech. *60*, 2011, p. 457.

[3] ElMaraghy H., Schuh G., ElMaraghy W., Piller F. *et al.* Product variety management. CIRP Annals - Manufacturing Tech. *62*, 2013, p. 629.

[4] Schuh G., Rudolf S., Vogels T. Performance Measurement of Modular Product Platforms. Variety Management in Manufacturing Proceedings of the 47th CIRP Conference on Manufact. Systems *17*, 2014, p. 266.

[5] Schuh G., Potente T., Thomas C., Schmitz S. *et al.* Design principles for an integrated product and process development approach for rotationally symmetric products. 2014 Proceedings of PICMET '14: Infrastructure and Service Integration, 2014, p. 2126.

[6] Brecher C., Editor. *Integrative Production Technology for High-Wage Countries*. Springer, 2012.

[7] Schuh G., Arnoscht J., Rudolf S. Integrated development of modular product platforms. Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET '10, 2010.

[8] Schuh G., Behr M., Brecher C., Bührig-Polaczek A. *et al.* Individualised Production, in *Integrative Production Technology for High-Wage Countries*, Springer Berlin Heidelberg, 2012, p. 77.

[9] Schuh, G. *Modular Product Platform and Variant Management: Conclusions of the Consortial Benchmarking*, Aachen, 2015.

[10] Aßmann, G. *Gestaltung von Änderungsprozessen in der Produktentwicklung*. Herbert Utz, München, 2000.

[11] Schuh G., Aleksic S., Arnoscht J. Module based release planning for technical changes. 2013 Proceedings of PICMET '13: Technology Management in the IT-Driven Services (PICMET), 2013.

[12] Schuh G., Rudolf S., Tönnes C., Aleksic S. Module-based release management for technical changes, in *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*, Springer International Publishing, 2015, p. 293.

[13] Belener, P.M. *Technisches Änderungsmanagement modularer Produkte und Prozesse*. Shaker, Aachen, 2008.

[14] IEEE Standard. Standard, Adoption of ISO/IEC 20000-2:2012, Information technology - Service managmntt - Part 2: Guidance on the application of service management systems, 2013(IEEE Std 20000-2-2013).

[15] Stahlknecht, P., Hasenkamp, U. *Einführung in die Wirtschaftsinformatik,* 8th edn. Springer, Berlin, 1997.

[16] van Bon, J., Wilkinson, J. *Foundations of IT Service Management: basierend auf ITIL V3.3,* 3rd edn. Van Haren, Zaltbommel, 2008.

[17] Carlshamre P. Release Planning in Market-Driven Software Product Development, 2002, p. 139.

[18] Wright, H. *Release Engineering Process: Their Faults and Failures*, Austin, 2012.

[19] Hayakawa, Y., Irony, T., Xie, M. *System and Bayesian Reliability.: Essays in Honor of Professor Richard E. Barlow on His 70th Birthday*. Singapore: Scientific Publishing Co. Pte. Ltd., Singapore, 2001.

[20] Schuh G., Desoi J.C., Lenders M., Witte V. Release-Engineering: An Innovative Approach to Handle Complexity of Mechatronic Products, in *Proceedings of the 1st International Conference on Electrical/ Electromechanical Computer-Aided Design & Engineering*, 2004.

[21] Dart, S. *The past, present, and future of configuration management*, Pittsburgh, 1992.

[22] Lindemann, U., Langer, S., Wilberg, J., Maier, A. *Änderungs-management-Report 2012*, München, 2012.

[23] Aleksic, S. *Nachhaltige Weiterentwicklung von modularen Produktarchitekturen durch Release-Management,* 1st edn., Aachen, 2016.

[24] Kano N., Seraku N., Takahashi F., Tsuji S. Attractive Quality and Must-Be Quality. Quality (Hinshitsu): The Journal of the Japanese Society for Quality Control, 1984, p. 39.

[25] Schuh G., Rudolf S., Tönnes C., Aleksic S. Release frequency for technical changes of modular product platforms. 2016 IEEE International Conference on Industrial Technology (ICIT), 2016.

[26] Paris Auto Show: Car Technologies Like Lane Assist, Rear Cameras Have Become Mainstream. International Business Times. http://www.ibtimes.com. Accessed 11 October 2015.