INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

DESIGN 2014

TOWARDS CYCLE-ORIENTED TRACEABILITY IN ENGINEERING CHANGE MANAGEMENT

N. Chucholowski, T. Wolfenstetter, M. C. Wickel, H. Krcmar and U. Lindemann

Keywords: engineering change management, requirements management, traceability

1. Introduction

Dynamic markets, regulatory environments and other external and internal cyclic influence factors force companies to make changes and adaptions to their products (product as a general term also standing for product service systems, PSS) [Langer et al. 2011]. For instance, revised laws, emerging technologies, changing market needs, rising competition, errors or uncertainties during development lead to target deviations. That means the actual state of a product does not meet the desired nominal state anymore. Hence, companies need to adapt their products, processes and production continuously among the product lifecycle [Huang and Mak 1999]. Those processes of adaption are addressed by the concept of engineering change management (ECM) [Huang et al. 2001] and are interpreted as internal cycles within product development [Langer and Lindemann 2009].
These engineering changes (ECs) not only affect the product and its design but also other related artifacts such as production processes [Huang and Mak 1999]. While changed requirements are commonly seen as possible triggers for engineering changes, the effect of an engineering change on other requirements is considered only by little literature. However, changing one part of a product can have unforeseen effects on the fulfilment of requirements which are not directly related to that part. Changed requirements are handled by requirements management. In this work we examine how an integration of engineering change management and requirements management could look like and how companies can benefit from traceability in their development cycles.
The following section describes our research methodology before section 3 gives an overview on requirements management and engineering change management, ending with the description of issues on the interface between the two topics. In section 4 we then present implications for the interface before the approach towards traceability in engineering change management is described in section 5. Section 6 concludes the paper and gives an outlook for future research.

2. Research Methodology

In order to analyze the current state of the art in research on how engineering changes and requirement changes are considered integrally we conducted a systematic literature review about engineering change management and requirements management. We started our analysis by initially investigating already known publications [Jarratt et al. 2011, Lindemann and Reichwald 1998, Pohl 2010, Wiegers 2009], that give a good overview about the respective topics. In order to find publications which directly associate engineering change management and requirements management, we additionally searched in online literature databases. As recommended by [Webster and Watson 2002] we further conducted a backward and forward search within the publications initially selected. In total we selected 95 publications that form the sample for our analysis. For this set of publications we

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

performed a qualitative content analysis in order to draw a summarizing picture of the state of the art in engineering change management and requirements management.
Additionally to the analysis of literature, we got insights into current practice through discussions with practitioners who are part of our industrial focus group on engineering change management.
In literature and in the focus group discussions, we particularly investigated how engineering change management and requirements management interface with each other. Altogether we want to answer the following research questions:

1. Which knowledge items are produced and required in engineering change management and requirements management?
2. How do cycles in engineering change management and requirements management interface with each other in terms of process management and organization?
3. How can traceability among different development artifacts contribute to managing engineering changes?

3. Overview: Requirements Management and Engineering Change Management

3.1. Requirements Management and Requirements Engineering

In the early phases of product development the problem needs to be stated in a form that can be understood by engineers and used to find a solution. The part of product development that is concerned with defining the problem domain is commonly known as requirements engineering (RE). When talking about RE it is important to clear out what is meant by the term requirement. Requirements describe qualitative and/or quantitative properties or conditions for a product [Ehrlenspiel 2009]. A more detailed definition of the term requirement can be found in the IEEE standards: "Requirements are statements of what the system must do, how it must behave, the properties it must exhibit, the qualities it must possess, and the constraints that the system and its development must satisfy" [Radatz et al. 1990]. These requirements not only stem from customers but also from other stakeholders like the engineers developing the product [Pohl 2010].
Leffingwell and Widrig [1999] do not distinguish between the terms RE and requirements management (RM) and describe it as the eliciting, organizing, and documenting of the requirements of the system in a systematic way. It further aims to establish and maintain agreement between customers and the project team on the changing requirements of the system. As illustrated in Figure 1 requirements engineering can also be divided into requirements development and requirements management [Berkovich et al. 2009]. Since we focus on requirement changes after they already have been developed, we further use this definition of RM which is described in the following.

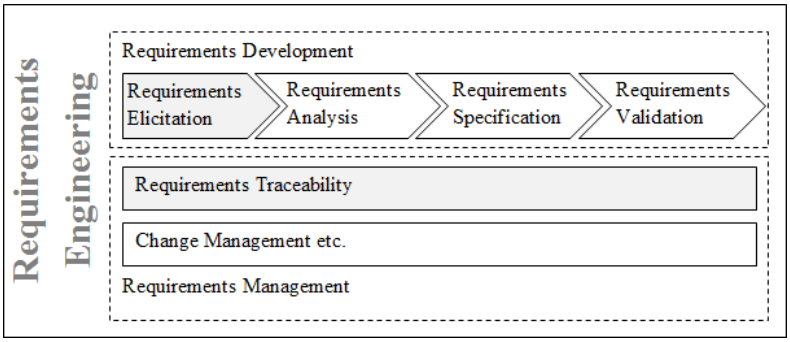

Figure 1. Considered activities of RE based on [Berkovich, et al. 2009]

The management of requirement changes (RCs) and the traceability of requirements are the two main tasks of **requirements management**. For instance, when a customer desires a new or changed functionality or product property a requirement is addressed and a potential change is identified. After identifying the encountered situation, a change request leads to several activities in the change management process: Create change request, determine attainability, plan, implement and evaluate changes. Each activity consists of several sub-activities to ensure customers', stakeholders' and producers' requirements. This step is supported by requirements traceability, which deals with the life

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

cycle of a requirement (traceability will be explained in more detail in section 5.1.). Changes in requirements are continuous and inevitable, because - while the product is being developed - customer needs evolve, competitors introduce products and processes that help to give them a competitive advantage, and political, organizational and technical environments change.
In general, changes are noticeable in two ways. Either existing requirements change or new requirements emerge after the requirements specification has been considered complete. Requirements usually change with a monthly rate up to 5%, normalized to the total project effort. Over the entire project progress requirements vary between 30% and 60% of the initial requirements analysis [Ebert 2010]. New or changed requirements during the project period must follow the same process and procedures as original requirements. Any change in requirements must be analyzed, evaluated and decided. For a controlled change of requirements in running projects a fixed change process must be defined [Versteegen et al. 2003]. Any change must be agreed through a specified instance. If changes are accepted or received in an uncontrolled manner, this might lead to even more requirements or a constantly re-tuning of previously recorded requirements by stakeholders. As a result, there will be additional costs and delays in the project process. The setting of deadlines for the adoption of requirements and a transparent communication of the effects of changes are possible solutions.

3.2. Engineering Change Management

The definitions of the term engineering change (EC) slightly differ in literature. While in the past ECs often only referred to modifications made to product components that were already in production [e.g. Wright 1997], nowadays any alteration to released parts, documents or software during the design process is considered as an EC [Jarratt, et al. 2011]. The handling of these changes is called engineering change management (ECM) [Jarratt and Clarkson 2005]. Other authors additionally see the documentation of all impacted product data [Rouibah and Caskey 2003], the documentation of the history of all changes of products and its associated documents [Huang and Mak 1999], or also the avoidance and anticipation of ECs [Lindemann and Reichwald 1998] as part of ECM. This shows that there are different perspectives on ECM in literature. Jarratt, et al. [2011] categorize the EC literature by a process perspective, tool perspective and product perspective. This categorization neglects the view on the different strategies regarding all perspectives and the fact that in some literature the documentation is reckoned as the main objective of ECM. Hence, we differentiate between a process, documentation and strategic perspective in the following. For a more detailed categorization we refer to [Hamraz et al. 2013] who performed a comprehensive literature review and categorized 427 ECM publications.
From a **process perspective**, ECM is the processing of engineering changes, i.e. starting with an EC request and finishing with its successful implementation (or its disapproval before) [e.g. Jarratt and Clarkson 2005] and providing respective methods and tools within the process steps. Jarratt and Clarkson [2005] suggest a generic change process in six basic steps shown in Figure 2.

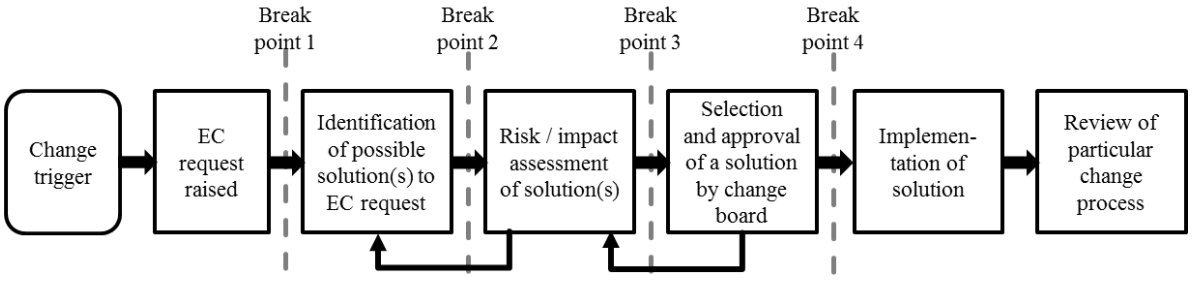

Figure 2. Generic engineering change process based on [Jarratt and Clarkson 2005].

ECM also deals with the constant **documentation** of changes, regarding change activities (EC processes) as well as old and new states of product documents [e.g. Huang and Mak 1999, Rouibah and Caskey 2003]. This can also be seen as the view of industrial standards (e.g. ISO 26262 or the German DIN 199-4 and DIN 6789-3) which aim to ensure that companies comply with other industrial standards such as ISO 9001 or with the requirements of their OEM customer (e.g. VDA 4965). Yet, as indicated above, ECM additionally pursues **strategies** besides the effective and efficient processing

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

and documentation of ECs. It further aims to avoid and anticipate ECs and to learn from ECs from the past.

3.3. Issues regarding the interface between requirements management and engineering change management

In order to refer to an artifact that can be regarded as the object of an EC (e.g. parts, components or product documentation) we use the term ‘solution artifact’ (SA) which is part of the solution domain. The term ‘requirement artifact’ (RA) represents a goal or a requirement. The problem domain is the counterpart to the solution domain and characterizes the problem that is addressed by the developed product by requirement artifacts.

A planned change of a solution artifact (i.e. an EC) should not be considered without looking at the requirements, because several requirements can be affected indirectly and lead to the need to change another solution artifact. The same is valid the other way around. When a requirement artifact is changed (i.e. a RC) it can affect other requirement artifacts directly or indirectly via several solution artifacts. As illustrated in Figure 3, there are known interrelations within requirement artifacts and solution artifacts and also dependencies in-between. For instance, RA 1 and RA 2 are related to each other, SA 1 and SA 2 fulfill RA 1. When there is an EC of SA 1, it could affect RA 1, which is related to RA 2. Then, if RA 2 has to be adapted, also SA 4 could be affected. Hence, there is an unknown dependency between SA 1 and SA 4 (an illustrative example for these correlations is given in [Koh et al. 2012]).

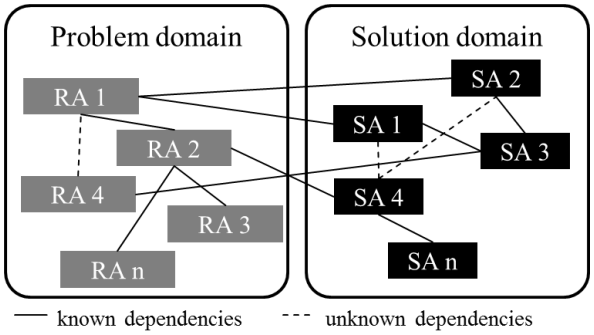

Figure 3. Dependencies between requirement artifacts (RAs) and solution artifacts (SAs)

Within the often referenced ECM and RM literature there is no interface directly addressed. Only the link of ECM to configuration management gives a small hint to consider both, solution artifacts and requirement artifacts, since configuration management aims to overlook all functional and physical characteristics and their changes and to verify the compliance with product specifications (i.e. requirements) [Jarratt, et al. 2011].

Only few publications are found that directly associate RM and ECM [Koh et al. 2008, Koh, et al. 2012, Morkos et al. 2012]. They initially come from ECM and research on change propagation. Indirect dependencies through requirement and solution artifacts are investigated with the help of matrices, where known dependencies are modeled and indirect dependencies are derived. However, using matrices bears some weaknesses, as they quickly become confusing if they grow in size or if weightings for the dependencies, different kinds of dependencies, conditional dependencies and several propagated indirect dependencies have to be considered. Moreover, the authors look at the dependencies in order to estimate change propagation, without describing implications for ECM and RM from the process and organizational perspective.

Further articles regarded as relevant mainly deal about product lifecycle management and product data management [e.g. Andersson et al. 2003]. They refer to RM as well as to ECM, but they rather just mention the terms without investigating the implications for the interface.

When looking into practice, the treatment of RCs and ECs can be very different. For instance, in one company that is represented in our focus group RCs are not considered integrated with ECs, besides seeing them as a trigger for ECs. In another company requirements also have a code number and are thereby treated the same as product components or its documentation in their product data

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

management (PDM) system. In a third company, requirements are stored in a special IT system. However, the consideration of all necessary solution artifacts and requirement artifacts that could be affected by one change request does not follow a structured formal process. On an organizational level, teams are built who have to assess and decide on the request.

4. Implications for the interface between requirements management and engineering change management

This paper does not investigate change propagation in order to assess change effects, but looks at it from a procedural and organizational perspective. This is important, since every propagated change theoretically leads to another change request (regardless if it concerns a RC or EC), which somebody has to decide upon. The conceivable procedural interactions between ECs and RCs are depicted in Figure 4. When there is a target deviation, the change procedure is triggered. ECs are either triggered by a deficient actual state (i.e. product does not meet requirements) or by a deficient or changed nominal state (for example misunderstood or changed customer requirements) [Fricke et al. 2000, Herberg et al. 2010]. In both cases an initial change is necessary and requested either in the solution domain (EC) or in the problem domain (RC). However, the change can make other changes necessary both in the solution domain and in the problem domain (i.e. further ECs or RCs are requested as depicted with gray arrows in Figure 4). This leads us to the conclusion, that the management of ECs and RCs should not be seen separated.

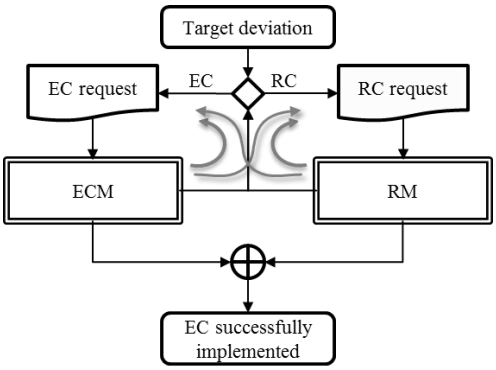

Figure 4. Change cycles within and between RM and ECM.

The information that is needed for a holistic estimation of the effects of one change is often distributed over a large variety of documents or it is only tacit knowledge that single engineers have. With the concept of traceability this information on each artifact that is generated during the development process can be stored systematically. If companies are able to use this hidden and distributed information they can improve the efficiency of their change processes. Furthermore, they can avoid errors or additional work by reducing unforeseen changes and hence avoid or shorten their development cycles.

5. Cycle-oriented traceability for the management of changes

Even though neither ECM literature nor RE/RM literature addresses the synchronization of ECs and RCs directly, it is encountered indirectly. There are approaches [Koh, et al. 2012, Morkos, et al. 2012] to assess change propagation not only on a component level, but also via requirement relationships. Furthermore, in ECM it is suggested to build an engineering change board including people from all relevant domains who could be affected by the change. Members in our focus group from industry state that people who control the requirements are also part of that committee in their companies. However, since the synchronization is not prescribed in a formal process, there is potential for errors. Errors may originate e.g. because some domains affected by the change can be forgotten or people (and thereby their knowledge) exit the company. Traceability bears the potential to formalize the necessary investigation of interrelations between ECs and RCs.

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

5.1. Theoretical background to traceability

With the ongoing digitalization more and more know-how is stored in documents as presentations, reports or construction plans. The implementation of a systematic organization and reuse of those documents awards companies with a competitive advantage [Liebowitz 1999]. Traceability aims to reuse such knowledge. Hence, experiences of individual and organizational knowledge become available and can be provided for future activities in order to improve processes and engineering designs [Hicks et al. 2002]. Moreover, knowledge-based product development aims to reuse best practices, reduced cycle times and improvements in product quality and variety [Rezayat 2000]. Traceability jointly connects single knowledge items or fragments in order to generate, dispose, retrieve, transform and apply them.

Especially the ability to follow the life of software artifacts has been used as a quality attribute for software [Winkler and Pilgrim 2010]. Defining, describing, capturing and following traces from and to artifacts of a software development are driven by requirements. Therefore the requirements engineering community has been the largest driver of traceability research. Traceability is defined in the IEEE Standard Glossary of Software Engineering Terminology as: [Radatz, et al. 1990]

1. "The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor–successor or master–subordinate relationship to one another. [...]
2. The degree to which each element in a software development product establishes its reason for existing."

To document the various dependencies, known or unknown, direct or indirect traceability links show the influence between the artifacts [Winkler and Pilgrim 2010]. Two kinds of linkages must be differentiated: A unidirectional depends-on or a bidirectional alternative-for link. Both can indicate an order in time or causality. [Spanoudakis and Zisman 2005] define eight different classes of traceability links listed in Table 1. They are partly used in our data model for traceability in change management in the following section.

Table 1. Classes of traceability links [Spanoudakis and Zisman 2005]

Traceability links	Explanation
Dependency	Indicates that the existence of an artifact depends on another
Generalization/Refinement	Shows the complexity of an artifact
Evolution	Reflects the change of an artifact
Satisfaction	Indicates, that an artifact was satisfied by another
Overlap	An intersection of two artifacts
Conflicting	Shows conflicts and inconsistencies between artifacts
Rationalization	Specifies the justification of the evolution of artifacts
Contribution	Shows the relationship between requirements and stakeholders

Traceability information helps to assess the effect of a requirements change and links to related requirements. It is the ability to verify an item by documented recorded identification. Model-driven development (MDD), an area where parts of the software development process are executed automatically, is able to leverage traceability by automatously generating these documented recorded identifications [Winkler and Pilgrim 2010]. It can be identified at any time, where, when and by whom solution artifacts are developed, manufactured, processed, stored, transported, used or disposed. This linking of requirements to system models increases the comprehensibility of the system. The impact of changing stakeholder requirements during the project is easier to assess. An explicit linking and traceability between requirement artifacts and solution artifacts facilitate crosschecking of system models with the associated requirements.

Outside the software and requirements engineering community the concept of traceability has attracted less attention among researchers. However, recently there has been significant progress in this area, for example on the field of tracing engineering information [Pavković et al. 2013]. While software development by its very nature produces artifacts (e.g. code or documentation) that are interpretable by machines so that traceable elements can be mostly created automatically and managed by common requirements engineering tools, the situation with the development of mechanical products is quite

different. Štorga [2004] argues that there are difficulties in achieving traceability in product development projects due to the incompatibility of information among heterogeneous design tools as well as human factors and the design process itself. An important issue regarding the implementation of traceability in engineering design is that only things that leave traces are traceable. This challenge is met by explicitly documenting different change states of an EC or RC on the one hand and all kinds of artifacts that are affected by the change on the other hand.

5.2. A data model for traceability in engineering change management

Štorga et al. [2011] present a reference model for traceability records that considers product and process related traceability elements regarding the four perspectives requirements, change, characteristics and decision traceability. Hence, they provide a useful holistic framework to achieve traceability of information objects within engineering design. However, the reference model does not address the different stages a change can evolve to. The evolving stages of ECs or RCs can be summarized as follows: A change proposal is triggered by an issue (i.e. target deviation), which then evolves to a change request if it is decided to go after it. Finally, when there is a promising option to solve the change issue, the change request becomes a change order which is the starting shot to implement the change. For every evolving stage, every domain directly or indirectly affected by the change has to be integrated, regardless if the initially triggered change was an EC or RC. This view can even be extended to other domains in the product lifecycle, such as production. Thus, the reference model by Štorga, et al. [2011] can be complemented by expanding the understanding of a 'change' from a single product component modification to a network of several related changes to requirement artifacts, solution artifacts or production artifacts. These correlations are modeled in a data model shown in Figure 5.

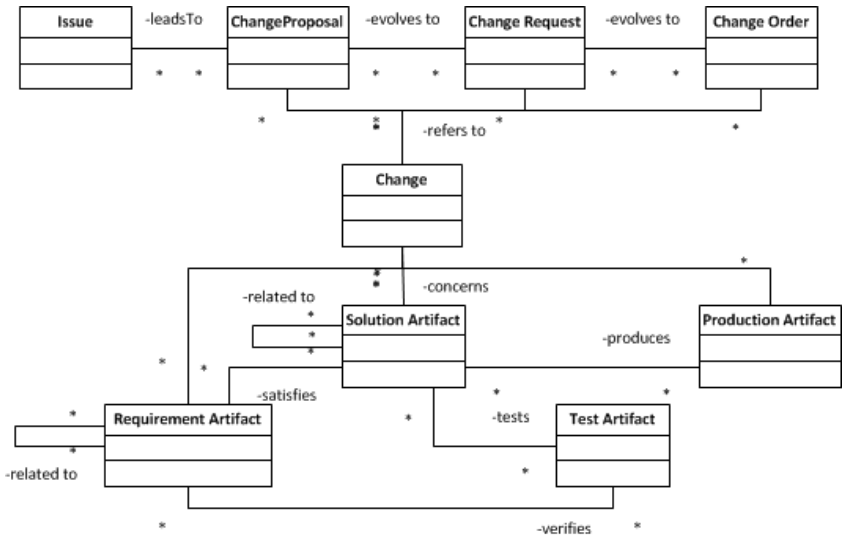

Figure 5. Data model for traceability in engineering change management.

The data model for traceability in ECM does not describe the change process itself but gives a static structure of how information regarding different artifacts within the ECM process is interconnected. An upcoming *issue* (i.e. target deviation) leads to a *change proposal* which then is connected to the entity *change*. The change proposal then evolves into a *change request* in the static model. The change request process determines attainability and plans the change. A successful change request then evolves into a *change order* where all information of issue, change proposal and change request are collected and associated with change request responsibilities, also for sub-activities for each part. The change order is also directly linked to the entity change through a *refers-to* connection. A change is further linked to three artifact types: requirement artifact, solution artifact and production artifact. The latter refers to all artifacts within production that are related to a product such as production processes or tools. They also have to be considered, since ECs can not only have direct impact on production processes or tools, but also changes in production artifacts often require other changes. The solution artifact is often related to itself and can offer various solution options. Requirement artifacts and

solution artifacts are linked through a relationship named *'satisfies'*, denoting which of the final solution artifacts are based on which requirements. A requirement artifact can additionally be related to another one. The solution artifact is tested through a test artifact, which verifies if the requirement artifact is satisfied by the solution artifact.
We believe that the tight connection between ECs and RCs should be reflected by deeply integrated tools or even a single tool for both RM and ECM. This way, analyzing the impact of ECs on requirements and vice versa can be facilitated. For solution, test and production artifacts on the other hand it seems promising to manage them in separate specialized tools and represent those artifacts as traceability records as proposed by [Štorga, et al. 2011]. The presented data model builds a basis to show the interrelations between different artifacts in the context of changes. With its help, traceability in ECM can be established, where not only ECs on a component level but all kind of product changes (including changes in service artifacts of a PSS), changes in requirements, changes in production processes or tools and changes in testing are considered. The following section gives an illustrative example of the data model.

5.3. Academic example

For an illustrative explanation of the data model we refer to an academic example of a pick and place unit of an industrial plant (for more details see [AIS 2014]). The unit grabs different work pieces (WP) that are stored in a stack and puts it into a stamping module where a specific note is stamped on the WP. Afterwards the WPs are transported to a sorting belt and are distributed to different slides depending on their material (white plastic, black plastic or metal). During the development of the system a customer changed one requirement so that not every WP had to be stamped anymore but only those made of metal. The changed customer requirement as an upcoming issue leads to a change proposal where the artifact that has to be changed is identified. This is the requirement "R1: All WPs have to be stamped" in our example. The change proposal then evolves into a change request that suggests changing the requirement to "R1*: Metal WPs have to be stamped, other materials can be stamped optionally". The required change leads to other potential changes of the already developed system. For instance, the three options "EC1: install additional inductive sensor that differentiates material", "EC2: modify software and use other existing sensors" or "EC0: no modification" are considered. The first two alternatives require further RCs, ECs or changes in production. Information about the affected artifacts and their respective relationships is stored in the entity change and thus can be retraced for the different options. After the decision on which option to implement, the change request evolves into a change order. The concept of cycle-oriented traceability helps to react to further cyclic influence factors that lead to changes by providing information about the history of past changes not only to solution artifacts, but also to requirement artifacts and production artifacts.

6. Conclusion and outlook

Engineering changes and requirement changes strongly interfere with each other. With the help of traceability in engineering change management this interface can be formalized on a process and organizational level. Thereby, unforeseen changes promoted by the initial change, errors and forgotten dependencies can be avoided. Further, the processing time of a change can be shortened.
By describing the organizational structure of potential interrelations of changes within the requirement and solution domain we aim towards a cycle-oriented management of changes. At the same time, the investigation of the underlying procedures for changes shows that the processes are very similar for all kinds of changes. Based on these findings, we presented a data model as research in progress where different evolving stages of a change and different artifacts related to the change are included. The data model should be seen as complementary to the reference model for traceability records elaborated by Štorga, et al. [2011]. The elements of the data model indicate knowledge items that are required or produced in a change process. The concept of traceability facilitates the access to necessary information about relations and dependencies between solution artifacts, requirement artifacts and production artifacts. Hence, the effects of a change to one artifact can easily be estimated and relevant people for the decision about a change and the implementation of a change can be identified. This leads to less errors and unforeseen effects of a change.

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/

The approach towards a cycle-oriented, integrated management of any kinds of changes with the help of traceability bears potential and has to be further developed. Moreover, also changes in other company departments such as sales, marketing, quality management, etc. that are influenced by various internal and external factors could be managed by a cycle-oriented traceability. Therefore we plan to extend the data model with artifacts regarding these departments and with their respective dependencies. The data model will then be tested as initial evaluation in a student research project. Another potential given by the use of traceability in ECM regards the learning from previous changes. [Sharafi et al. 2010] argue that especially in large organizations and in the context of complex products, engineering change management can be supported by hidden, but valuable knowledge. This knowledge can be discovered in the history of former change processes. The knowledge from former cycles can be utilized through successful data management to speed up iterative change and engineering processes [Sharafi, et al. 2010]. An important factor for the reuse of knowledge is traceability. Through the compound of elements the knowledge becomes contextualized. This in turn enables its transfer and successful reuse [Ramesh 2002].

Acknowledgement

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research center 'Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products'. Additionally, we thank the practitioners in our industrial focus group on engineering change management for the prosperous collaboration.

References

AIS, Institute for Automation and Information Systems: The Pick and Place Unit, available at: http://www.ais.mw.tum.de/ppu/, 2014.

Andersson, F., Sutinen, K., Malmqvist, J., Product Model for Requirements and Design Concept Management: Representing Design Alternatives and Rationale, International Conference on Systems Engineering, SE-412, Citeseer, 2003.

Berkovich, M., Esch, S., Leimeister, J. M., Krcmar, H., "Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review", Tagungsband der 9. Internationalen Tagung Wirtschaftsinformatik, 2009.

Ebert, C., "Systematisches Requirements Engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten", dpunkt. verlag, Heidelberg, 2010.

Ehrlenspiel, K., "Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit", Hanser, München, 2009.

Fricke, E., Gebhard, B., Negele, H., Igenbergs, E., "Coping with changes: causes, findings, and strategies", Systems Engineering, Vol.3, No.4, 2000, pp. 169-179.

Hamraz, B., Caldwell, N. H. M., Clarkson, P. J., "A Holistic Categorization Framework for Literature on Engineering Change Management", Systems Engineering, Vol.16, No.4, 2013, pp. 473-505.

Herberg, A., Langer, S., Netter, F., Lindemann, U., Characterizing triggers of reactive cycles within design processes based on process observation, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE, 2010, pp. 972-976.

Hicks, B., Culley, S., Allen, R., Mullineux, G., "A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design", International journal of information management, Vol.22, No.4, 2002, pp. 263-280.

Huang, G., Mak, K., "Current practices of engineering change management in UK manufacturing industries", International Journal of Operations & Production Management, Vol.19, No.1, 1999, pp. 21-37.

Huang, G. Q., Yee, W. Y., Mak, K. L., "Development of a web-based system for engineering change management", Robotics and Computer-Integrated Manufacturing, Vol.17, No.3, 2001, pp. 255-267.

Jarratt, T., Clarkson, J., "Engineering change", in: Clarkson, J., Eckert, C. M. (Eds.), Design process improvement, Springer, London, 2005, pp. 262-285.

Jarratt, T. A. W., Eckert, C. M., Caldwell, N. H. M., Clarkson, P. J., "Engineering change: an overview and perspective on the literature", Research in engineering design, Vol.22, No.2, 2011, pp. 103-124.

Koh, E., Keller, R., Eckert, C., Clarkson, P., "Influence of feature change propagation on product attributes in concept selection", International Design Conference - DESIGN 2008, Dubrovnik - Croatia, 2008, pp. 157 - 166.

Koh, E. Y., Caldwell, N. M., Clarkson, P. J., "A method to assess the effects of engineering change propagation", Research in Engineering Design, Vol.23, No.4, 2012, pp. 329-351.

Langer, S., Lindemann, U., "Managing Cycles in Development Processes - Analysis and Classification of External Context Factors", 17th International Conference on Engineering Design, Bergendahl, M. N., Grimheden, M., Leifer, L. (Eds.), Stanford University, California, USA, 2009, pp. 1-539 - 531-550.
Langer, S., Herberg, A., Körber, K., Lindemann, U., Integrated system and context modeling of iterations and changes in development processes., Proceedings of the 18th International Conference on Engineering Design (ICED11), 2011, pp. 499-508.
Leffingwell, D., Widrig, D., "Requirements Management. A Unified Approach", Addison-Wesley, München, 1999.
Liebowitz, J., "Knowledge management: handbook", CRC press, 1999.
Lindemann, U., Reichwald, R., "Integriertes Änderungsmanagement", Springer, Berlin, 1998.
Morkos, B., Shankar, P., Summers, J. D., "Predicting requirement change propagation, using higher order design structure matrices: an industry case study", Journal of Engineering Design, Vol.23, No.12, 2012, pp. 905-926.
Pavković, N., Štorga, M., Bojčetić, N., Marjanović, D., "Facilitating design communication through engineering information traceability", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol.27, No.02, 2013, pp. 105-119.
Pohl, K., "Requirements Engineering: Fundamentals, Principles, and Techniques", Springer Publishing Company, Incorporated, 2010.
Radatz, J., Geraci, A., Katki, F., "IEEE standard glossary of software engineering terminology", IEEE Std, Vol.610121990, 1990, pp. 121990.
Ramesh, B., "Process knowledge management with traceability", Software, IEEE, Vol.19, No.3, 2002, pp. 50-52.
Rezayat, M., "Knowledge-based product development using XML and KCs", Computer-aided design, Vol.32, No.5, 2000, pp. 299-309.
Rouibah, K., Caskey, K. R., "Change management in concurrent engineering from a parameter perspective", Computers in Industry, Vol.50, No.1, 2003, pp. 15-34.
Sharafi, A., Wolf, P., Krcmar, H., Knowledge Discovery in Databases on the Example of Engineering Change Management, Industrial Conference on Data Mining-Poster and Industry Proceedings, 2010, pp. 9-16.
Spanoudakis, G., Zisman, A., "Software traceability: a roadmap", Handbook of Software Engineering and Knowledge Engineering, Vol.3, 2005, pp. 395-428.
Štorga, M., Traceability in product development, Proceedings of the International Design Conference, DESIGN, Dubrovnik, 2004.
Štorga, M., Marjanovic, D., Savšek, T., Reference model for traceability records implementation in engineering design environment, Proceedings of the 18th International Conference on Engineering Design (ICED2011), Copenhagen, Denmark, 2011.
Versteegen, G., Heeler, A., Heßeler, A., Hood, C., Missling, C., Stücka, R., "Anforderungsmanagement: formale Prozesse, Praxiserfahrungen, Einführungsstrategien und Toolauswahl", Springer, New York, 2003.
Webster, J., Watson, R. T., "Analyzing the Past to Prepare for the Future: Writing a Literature Review", MIS Quarterly, Vol.26, No.2, 2002, pp. 13–23.
Wiegers, K. E., "Software Requirements", Microsoft Press, 2009.
Winkler, S., Pilgrim, J., "A survey of traceability in requirements engineering and model-driven development", Software & Systems Modeling, Vol.9, No.4, 2010, pp. 529–565.
Wright, I. C., "A review of research into engineering change management: implications for product design", Design Studies, Vol.18, No.1, 1997, pp. 33-42.

Dipl.-Ing. Nepomuk Chucholowski
Research Assistant
Technical University of Munich, Institute of Product Development
Boltzmannstr. 15, 85748 Garching, Germany
nepomuk.chucholowski@pe.mw.tum.de
http://www.pe.mw.tum.de

This paper was originally published in the proceedings
of the 13th International Design Conference, DESIGN'14
by the Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb: http://www.designconference.org/