ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

SYSTEM ARCHITECTURE CHANGE DECISIONS IN MULTI-VARIANT PRODUCT PORTFOLIOS

Maximilian KISSEL, Udo LINDEMANN
Technische Universität München, Germany

ABSTRACT

Decisions in complex design environments like in system architecture design can have wide-ranging effects on the operations of a company. The contribution of this paper to the body of research is to clarify the nature of decision-making on engineering changes and to explore success factors towards more rational and sustainable decisions. Our results were obtained through a literature review and evaluated in the context of a research project with six industry partners who formed an industry focus group. We propose a research framework that systemizes the objectives, challenges and possible enablers of complex decision situations. A proceeding model is proposed where possible support in decision-making is highlighted. As implications we claim to support complex decisions by making relevant entities and their dependencies more transparent and accessible. Therefore, a consistent database could serve for consistent model generation, visualization, analysis and evaluation. This will have a positive impact on development costs, resource allocation, product quality, flexibility towards the market and delivery time.

Keywords: system architecture, engineering changes, decision-making

Contact:
Maximilian Kissel
Technische Universität München
Institute of Product Development
Garching
85748
Germany
kissel@pe.mw.tum.de

1 INTRODUCTION

In complex environments like in system architecture design, easy questions can demand difficult-to-give answers. An example arose in one industry case as a manufacturer of power tools decided to develop a new tool for its product portfolio. The representative of the responsible business unit had to decide if there is a need for a new powertrain for the new product or if an existing design can be reused. The answer to this question has wide-ranging consequences e.g. for the resource-allocation of the design department, the prioritization of other development projects and even the production planning. But there could also be indirect effects of this decision e.g. for the supply chain, logistics, marketing, and services if this change propagates on to their relevant planning parameters. So, how could the responsible be supported in making a rational decision with low impact on costs, resources, development time, and product quality?

Which change has low impact on the rest of product portfolio and on other stakeholders? Can an existing design be reused? Or can carry-over parts of existing designs be used to reduce the overall development effort? How much effort is it to change an existing design, and how critical does a change propagate in the neighborhood of its origin? At what point do we need to make a complete new design and how to integrate a new design in an existing product portfolio most efficiently? Should innovations or other changes be anticipated when designing a new powertrain? What kind of change can be realized with available resources, causes acceptable cost, can be delivered in time and with the required quality?

Of course, there might be several fractions of the answer available in some distributed databases, in the minds of experienced developers or product managers. For some information, the effort to gather it is simply too high. Then, decisions are often based on heuristics or even gut feeling. Due to the complexity of the design decision described above and a lack of transparency of the dependencies in the system environment, high effort is required to compile all necessary information to come to a sustainable and rational decisions how to answer to these questions.

Methods and tools are needed to sustainably develop, change, maintain, assess, and improve complex product portfolios. Decisions on engineering changes can be supported through transparency of dependencies between relevant system elements. But how can the required transparency be provided? What are possible success factors to a decision-making support?

> *Therefore, the contribution of this paper is to clarify the nature of decision-making on engineering changes and collect requirements how system architects can be supported in order to make more rational and sustainable decisions in system architecture design.*

This report is organized as the following. First, we explain our research methodology and define the basic concepts of this paper: system architecture, engineering change and the nature of decision-making. Then, we start building up our framework of decision-making in system architecture design. We clarify the challenges and objectives of decision makers in system architecture design. Then, we discuss which enablers could lead to success and achieve the objectives. After that, we propose an approach to support the decision-making process in system architecture design systematically. We discuss the effects of the measures and propose implications for industry and academia. In the conclusion provide an outlook on future steps.

2 RESEARCH METHODOLOGY

An aim of this paper is to describe the existing situation in system architecture design and highlight the problems of decision-making in this field. The presented work is based on qualitative research to investigate the nature of the problem, relevance of the research topic, and to identify success criteria and key factors that are most suitable to address possible support for the problem (Blessing & Chakrabarti 2009).

This research is embedded in the project AMISA funded by the European Commission. Six industry partners form an industry focus group in the project. Two small-size and four large-size companies are operating in various branches, i.e. automotive, aerospace, manufacturing equipment, packaging, optical engineering, and communication technology. The aim of the project is to develop methodologies that support Design for Adaptability (cf. Hashemian 2005; Li et al. 2008), meaning *Adaptable Design* (e.g., design of a product portfolio based on modular building blocks) and *Adaptable Products* (e.g. efficient upgrade of products in use). This paper should set Design for Adaptability in a larger context and clarify the design situation. Both varieties of adaptability – product and design adaptability – are proactive engineering changes, because distinct design effort is spent to allow possible (under uncertainty) changes (here: adaptations) in the future. Here, this type of change is contrasted to reactive changes, where the cause of a change is already in place.

To clarify the problem of decision-making on system architecture changes in complex development environments, we conducted an extensive literature review on decision-making in engineering design, engineering change management, and system architecture design. The research framework deducted and proposed in this paper was presented to, evaluated and complemented by the industry focus group and academic colleagues in the consortium. The results of this research were introduced in six industrial case studies. In addition to that, we are currently conducting another case study at a project-independent company (i.e. a manufacturer of power tools) that was not involved in the methodology development in order to evaluate unbiased application and success of the methodologies. The evaluation of the framework is work in progress.

3 DEFINITION OF KEY CONCEPT

The Nature of Decision-Making

Laux et al. (2012) differentiate between *Prescriptive* and *Descriptive Decision Theory*. Descriptive decision theory tries to explain how and why decisions are made in reality. Prescriptive (or normative) decision theory explores the logic of decision-making and how to come to rational decisions. It provides guidance how to decide between alternative solutions with regards to several conflicting objectives and – if applicable – how to decide under uncertainty. The focus of our research is on prescriptive decision theory to support rational decision-making in system architecture design. Although, we are aware that the decision made at the end can highly be influenced by individual believe, political and/or strategic aspects and deviate therefore from the rational input. But the utmost rational input to the final decision should be based on transparency of the relevant dependencies and boundary conditions.

Krishnan and Ulrich (2001) describe typical product development decisions for *setting up* and *within a development project*. Especially in the concept phase, there are crucial decisions to be made by system architecture design responsibilities like e.g. "What is the product architecture? What variants of the product will be offered? Which components will be shared across which variants […]?", etc. In this early phase, decisions set already the course also for subsequent decisions of supply chain design, product design, performance testing and validation, and production ramp-up and launch. The architectures of both the product and the product portfolio also trigger the decisions in setting up the project – namely in product strategy and planning, product development organization and project management.

Scherpereel (2006) built a taxonomy for decision types. "First-order problems or decisions typically have static properties and are associated with high levels of certainty and simplicity. These problems/decisions […] have well-established solution methodologies, characterized by rational deterministic rules and deductive procedures. Second-order problems/decisions have probabilistic uncertainty, are often complicated, and follow definable dynamic processes. These problems/decisions […] rely on probability theory and inductive logic for solutions. They are typically approached using axioms, computer simulations, and a constrained model of the actual phenomena of interest. Third-order problems/decisions are those with genuine uncertainty, complexity, and dynamics. These problems/decisions rely on abductive logic and heuristic solutions. The objective is to find acceptability and effectiveness in the results." In this paper, we want to focus on first and second order decisions.

Steffens et al. (2007) describe that decisions in the system architecture design processes can be operational or strategic, and have a short-, mid- or long-term perspective. While there are predominantly financial criteria in place to assess change decisions, they found in a survey among change managers that the most important criteria for decision-making on engineering changes regard *project efficiency*, *customer impact* and the *project portfolio*. *Business success* and *preparing for the future* were considered less frequently. In all cases, the interviewees highlighted the impact of a change on the technology platform and resource dependencies. During a change evaluation, also "impact of changes on *product or technology roadmaps*, *development of other products in the same product line* and *ongoing developments in other product lines"* were considered.

They also found that traditional product development decision-making literature focuses decisions at projects gates but neglects decision-making on changes between gates. We want to complement this aspect with the consideration of *who prepares and who takes the decision*. A coworker in system architecture department or design department can prepare the clarification of the goals, the analysis and evaluation of alternative solutions. She/he has a different view on the decision-making problem than a product/project manager or a change board who have to approve the decision. Both parties need distinct ad individualized support in the process.
Laux et al. (2012) describe the decision-making process (cf. Figure 1) as an iterative sequence of (1) problem definition, (2) specification of target system, (3) exploration of alternative solutions, (4) selection of one alternative, and (5) realization of decision.

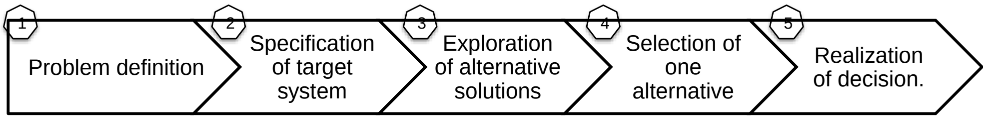

Figure 1: The decision-making process from Laux et al. (2012)

The basic logics of both the engineering change process and the generic decision-making process are similar.

Engineering Changes

The body of Engineering Design literature comprises an extensive and well-accepted work elaborating methodologies, tools and proceeding models for *ab initio* design. Typically, it is assumed rather to start a development process from sketch than changing an existing design. All this work has its *raison d'être* to educate and to guide engineers in the design of products holistically. However, *ab initio* design is quite seldom in practical engineering. The typical design task is to change an existing design rather more often evolutionary than revolutionary (as it was for example the case at the power tool manufacturer in the introduction).
For the definition of Engineering Change, we refer to Jarrat et al. (2011) who defined it as "making alternations to (parts, drawings or software of) a product and Engineering Change Management to organizing and controlling of this process. […] A change may encompass any modification to the form, fit and/or function […] and may alter the interactions and dependencies of the constituent elements of the product." Product changes occur during the whole product life cycle and may have an impact on other components or products in the portfolio.
Two types of causes for change are differentiated (Jarratt et al. 2011): "**emergent changes** arise from the properties of the product itself": error correction, safety, change of function, product quality problems. Different stakeholder – namely customers, sales and marketing, product support, production, suppliers, product engineering, company management, and legislators – might **initiate changes** to a product in order to improve, enhance or adapt them. When a change initiates another change then it propagates. The change propagation was discussed intensively over the last decade by (e.g. Hamraz et al. 2012; Clarkson et al. 2001).
Figure 2 illustrates a generic change process (Jarratt et al. 2011). One or more change triggers lead to a (1) change request. Before the approval of the request, (2) possible solution(s) are identified and (3) risk and impact are assessed. (4) A change board selects and approves a solution that is (5) implemented afterwards. After approval, there should be (6) a review of the particular change process. At several break points there can be iterations or termination of the change process.

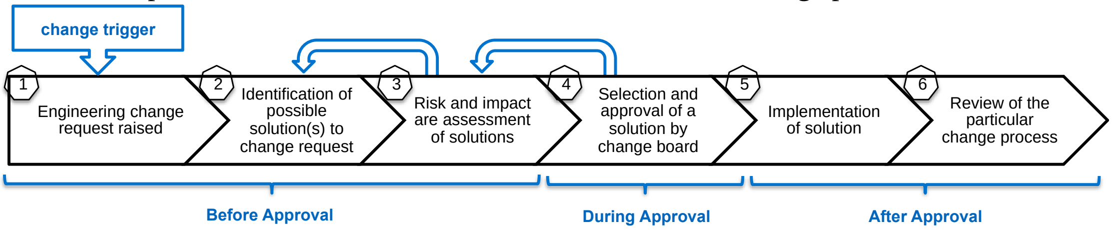

Figure 2: A model of a generic change process from Jarrat et al. (2011)

As parts and/or sub-modules may be re-used in other products of the whole product portfolio, we believe that the impact of the change must be managed beyond the borders of the single product. Jarrat et al. (2011) further define that "the impact a change has on a product is governed by three factors: (1)

the complexity of the product, (2) the architecture of the product, and (3) the degree of innovation within the product". From our observations in industry cases, we want to complement this definition by enlarging the scope from single products to product portfolios. There are multiple dependencies from one single design to other products beyond the system border of a single product. E.g., when a supplier announces discontinuation of supply of a component that is reused in various products and is supposed to be replaced, the impact of this change has to be integrated and tested among all the affected products individually. The experts from industry stated that it is crucial for an efficient change like this to overcome the lack of transparency of logical dependencies.

The property of a system of "being changed easily" is named changeability. Like in changeability, the basic concept of an "ility" (De Weck 2011; Ross et al. 2008) is normally to combine a verb or an adjective with the appendix "-ability". Here, the *ability of changing something* expresses to perform this task in a reasonable time with a rational amount of effort. Therefore, we determine the amount of changeability of a system as the triangulation of time, effort, and costs to perform a change. The faster, easier, and the cheaper the change is, the higher the changeability of a system (Kissel et al. 2012). Two subsets of changeability are *adaptability* and *flexibility*. Gu et al. (2009) state that adaptions are conducted by a person outside of the product/system, like the user or the designer. In contrast to adaptable systems, flexible ones have the ability to change internally to fit changes from the environment. Hence, flexibility is the extreme case of adaptability in terms of effort from outside to change the system. The changeability of a system like a product or a product portfolio can be shaped on an architecture level which is explored in the following section.

System Architecture Design

Intentionally, we speak of *systems architecture* rather than of *product architecture* in this paper. Product architecture is defined by the "arrangement of functional elements; (2) the mapping from functional elements to physical components; (3) the specification of the interfaces among interacting physical components" (Ulrich 1995). While a product can be a system, a system does not necessarily need to be (only) a product. Product architecture describes the principles of design of a product, whereas the system architecture can describe the design principles of a product family, a product portfolio or any other system (Jiao & Tseng 2000). In the system architecture, it can be defined how a product is embedded in the portfolio and how the dependencies of the components and functions of a product interrelate with other components, functions and products within the portfolio.

Why is the problem of decision-making such a challenge especially in the systems architecture environment? The INCOSE handbook describes the role of architecture design as a central technical process in systems engineering (Haskins et al. 2010). After ISO/IEC 15288:2008, the purpose of architecture design process is to transform stakeholder needs into "a set of separate problems of manageable, conceptual and, ultimately, realizable proportions". That incorporates the definition, analysis, evaluation documentation and maintenance of the architecture, and the definition of integration strategies and test cases. In a company, the individual manifestation of this purpose is highly dependent on the nature of its business and products. Typical task to be made in system architecture design observed in our research projects were (cf. Plaikner et al. 2012 – list makes no claims of being complete): Definition of portfolio and product families, generic product structure and product decomposition, platform management, modularization, variants and configuration management, and system and function definition etc.

The high number of different entities, numerous different dependencies, several external constraints and the lack of transparency about that all cause the complexity of decision-making in system architecture design.

Due to its role in the product development process, system architecture design has naturally many interfaces to a number of various stakeholders. Most of the decisions taken in the early product life cycle phases - where systems architecture department operates – have direct impact on the work of departments subsequent in the life cycle like design, production, supply chain, sale, and service/maintenance. These decisions have to be coordinated carefully as they conflict with constraints and objectives of the stakeholders. Unfortunately in real life, the awareness of these dependencies to other stakeholders is often not anywhere near. Time and cost pressure and the lack of awareness of possible degrees of freedom of these wide-ranging decisions make a rational decision even more difficult and have to be made viscerally.

Despite of the importance of the decisions to be made in system architecture tasks, we rarely observed at companies that one clear responsibility is defined. Some are establishing this central role in their company; some still operate system architecture tasks uncoordinated with unclear responsibilities. Unclear responsibility of system architecture design and lack of central coordination of the tasks exacerbate the situation to make rational and sustainable decisions. On the question who takes these decisions and should be supported, we assume here that there is one department, person of similar responsible for system architecture design decisions.
In the next section, we will set up a framework that consolidates the insights of the literature research and highlights how the process of decision-making in system architecture design could be supported. In the industry cases, we observed challenges to overcome and derived critical success factors and requirements towards a transparent, rational, and replicable decision-making when engineering changes appear that have an impact beyond the borders of a product.

4 RESEARCH FRAMEWORK FOR SYSTEM ARCHITECTURE CHANGE DECISIONS

In this section, we propose a framework that systemizes decision-making in system architecture design and highlights possible spots for support in the process of decision-making. Goal of the framework is to illustrate a target-oriented proceeding when a change occurs. The typical change process is governed by overall objectives like e.g., cost- and time-efficiency. To achieve these objectives certain challenges have to be overcome. The enablers to achieve the objectives are first success criteria towards the development of decision support in system architecture design.
As some *challenges* were already mentioned in the introduction of this paper, we want to name a few more that we observed among our industry focus group:

- Decisions under time- and cost-pressure
- Decisions under uncertainty and unclear risks
- Complexity of decisions
- Conflicts of objectives in multi-objective solution spaces
- Conflicts with stakeholder interests
- Constraints that have to be considered
- Unclear impact of decisions
- Information abundance and right selection of relevant data
- Accessibility to relevant information and data consistency
- Unclear responsibility of who prepares the decision and who makes it?
- Misinterpretation of information due to different terminology among stakeholders
- Alignment of decisions with corporate strategies

These challenges have to be overcome in order to achieve the objectives in system architecture design.
In order to develop sustainable support for system architects in decision-making, we collect critical success factors and requirements in the next section.

General Objectives and Enablers in System Architecture Design

Predominant objective is the *reduction of costs*. Costs can be reduced, e.g., by efficient allocation of resources, re-use or parameterization of existing designs, exploitation of economies of scales, or reduction of administrative costs. At the end, all objectives in architecture can be aligned and should contribute to reduction of costs directly or indirectly.
Further goals are *shorter delivery time* to enhance time-to-market. Therefore, it is critical how to enable the market need with the available resources and how to remain flexible to changing market needs. Several measures are discussed in literature to increase the effectiveness and efficiency of the internal variance and complexity of the product portfolio in order to provide the expected external variance on the market. Beside others, the group of Prof. Krause published comprehensive work on this issue (cf. Blees et al. 2010; Eilmus et al. 2012; Jonas et al. 2012).
Another goal to be addressed is *overall product quality*. Through re-use of existing already tested and validated designs the risk of quality issues is less than of a new design.
As discussed above, a main challenge is the complexity of managing multi-variant portfolios. Hence, a goal is to *mitigate complexity* and *make the complex dependencies manageable*. Complexity originates from a big number of different elements and dependencies, a lack of transparency of relevant

coherences and uncertainty about the impact of changes on the system. Decisions in such complex environments could be supported by

- Reducing the information abundance to only relevant information,
- Providing an overview of alternative actions and solutions,
- Providing functionality to analyze and visualize alternatives,
- Evaluating viable options in terms of costs, time, quality, risk etc.
- Monitoring goal attainment of key indicators of the product/system architecture (e.g. in a System Architecture Balanced Scorecard)
- Serving with methodologies to support distinct decision-making processes like e.g. modularization methodologies (e.g. Blees et al. 2010; Erixon 1996).

The issue of uncertainty of changes occurring in the future and ensuing change effects is impeding solid planning and is a considerable risk to success of system architecture design. Thus, a goal is to *reduce risk and uncertainties*. This issue can be addressed by

- Support of decision-making under uncertainties and risk (Laux et al. 2012),
- Estimation of change propagation and impact assessment (e.g. Clarkson et al. 2001), and
- Design and manage the multi-variant portfolio for better adaptability (Kissel et al. 2012).

An additional goal might be to make *sustainable, rational, and replicable (comprehensible) decisions*. Therefore, the effort to document and reason the decision could be reduced through easy access to relevant data, support or even automation of report generation and storing.
We think that the main success factor to achieve all these goals is in *providing transparency* about *setscrews* and *degrees of freedom in complex design environments* where and how to change the system efficiently and to avoid aftermaths of unexpected change propagation.

Approach to Handle Engineering Changes in Systems Architecture Design

In this section, we propose a generic approach how to handle changes in system architecture design. The approach is consolidated from the work above and based on the generic change process published by Jarrat et al. (2011) and the decision-making process by Laux et al. (2012). The approach is shown in Figure 3.

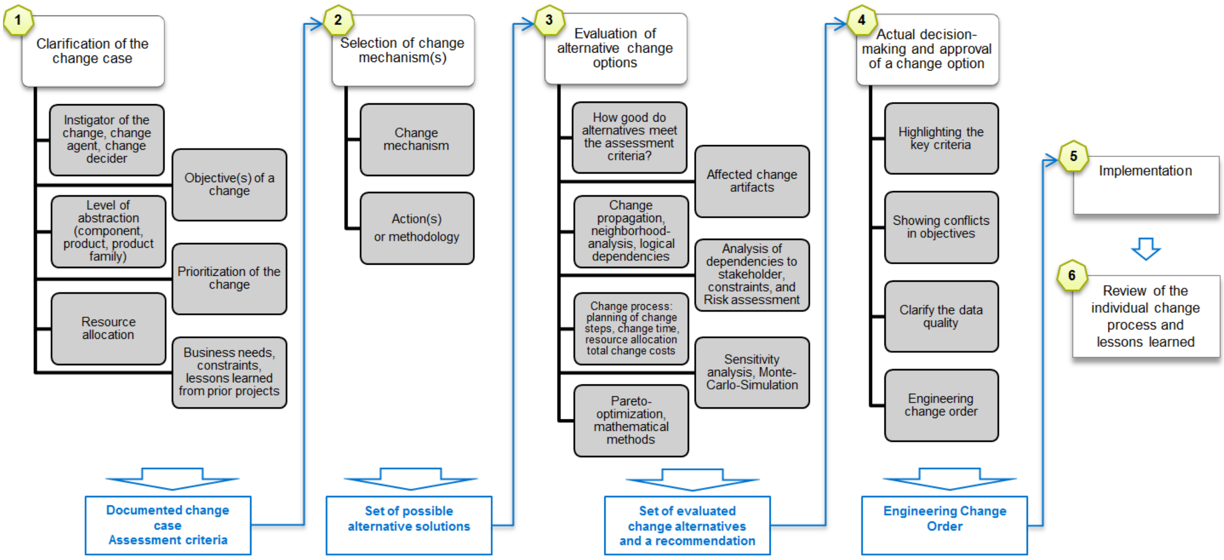

Figure 3: A generic approach to support decision-making on changes in system architecture design

Not only an engineering change request – may it be an emergent or an initiated change – can trigger the procedure. It can also be triggered if the monitoring (e.g. in a system architecture balanced scorecard) indicates deviation of a key indicator. The procedure is divided in six phases: (1) The first step is the clarification of the change case, the second step (2) is the selection of change mechanism(s), step (3) is the evaluation of alternative change options, step (4) is the actual decision-making and approval of a change option, step (5) is the implementation, followed by step (6) a review of the individual change process.
The **clarification of the change case** (1) is a crucial step to assure a target-oriented decision support. It has to be defined who the *instigator of the change* is (e.g. a stakeholder, the system itself, a key

indicator in the monitoring system), who will prepare and implement the change (i.e. *change agent*: designer, system architect) and who will decide about the change and the solution how to implement it (i.e. *change decider*: the designer, change board, project or product manager). Additionally, the *objective(s) of a change* must be clear. That could be clarified in a target-actual comparison taking into account, e.g., characteristic of the product structure or key indicators. And, it is stated on which *level* of the product-portfolio the change should be implemented, i.e. on component-, module-, product-, product family-, or product-portfolio level. Also, the *prioritization of the change* must be defined, if the change has to be implemented immediately or later. The prioritization is also an indicator for the possible *resource allocation* for the change process. Knowledge about *business needs*, *constraints*, *lessons learned* from prior similar cases could also be documented if available. The result of this step is a *documented change case* and a list of possible *assessment criteria* for the alternative solutions.
After the clarification, we collect and (2) **select possible change mechanism(s)**. A *change mechanism* is the way or path, how the new state is reached (Ross et al. 2008). It can be an action like *add*, *delete*, *merge*, *separate*, *scale*, or *modify* an element in the system (cf. Kissel et al. 2012) or a methodology like e.g. re-modularization (Erixon 1996; Engel & Browning 2008). The result of this step is a *set of possible alternative solutions*.
In the next step, – (3) **evaluation of alternative change options** – the results of step (2) are evaluated how good the *alternatives meet the assessment criteria* of step (1). To do that, the direct affected *change artifacts* (e.g. certain components, modules or products in a product family) have to be examined. To give an example, a change of a certain requirement was described in the change case. This requirement is fulfilled by certain functions and components in the system. A *network analysis* – e.g. matrix- or graph-based – could support the identification of artifacts that are directly affected by the change. According to the scope of the analysis, the network analysis could comprise a single product or all products that share common or similar parts. To analyze possible *change propagation* a *neighborhood-analysis* can be done and *logical dependencies* (e.g. components that are logical and – or even physical – related through functions) can be analyzed. Furthermore, other *analyses of dependencies to stakeholder*, *constraints*, or *risk assessment* can be performed. This is highly dependent on the available data or the effort to obtain this data. To assess the alternative change scenarios in terms of cost and time, the change case has to be translated into a *change process*: this comprises the planning of *change steps,* the estimation of *change time*, estimation of *resource allocation* and estimation of *total change costs*. If the change case describes a possible change in the future like an upgrade of a product, it can be evaluated how much money can be invested today (option costs) to have the option in the future to perform a cost-efficient upgrade (upgrade costs). This was investigated in more detail by Schrieverhoff et al. (2012). Computational tools, e.g., *sensitivity analysis* or *Monte-Carlo-Simulation,* can address the issue of uncertainty. The issue to resolve multi-objective problems could be addressed by *Pareto-optimization* or other mathematical methods (cf. Laux et al. 2012). The result of this step is a *set of evaluated change alternatives* and *a recommendation*.
Dependent on the criteria set, the visualization of the results can strongly support the decider what alternative to choose. The (4) **actual decision-making and approval of a change option** can be supported by visualizations that are *highlighting the key criteria*, *showing conflicts in objectives* and *clarify the data quality* on which the recommendation is based (e.g. by a visual representation of a sensitivity analysis). The decision has to be documented and communicated among the affected stakeholders. The result of this step is a detailed *engineering change order*.
In the following, this order can be implemented. As support in the (5) **implementation** phase, all the *information* about change artifacts, mechanisms, effects, process steps, costs, etc. *can be used* by the change agents actually implementing the change.
Jarrat et al. (2011) suggested to close the change process with a (6) **review of the individual change process**. This makes sense in order to profit from lessons learned in future change processes. The effort for documentation should be small, when all steps above were performed on integrated computational tool and based on a consistent database. This could help to sustainably improve the change processes in a company. When typical cost drivers in the actual cases can be identified, strategies could be derived accordingly to reduce costs sustainably in the future. This could be done, e.g., by creating an adaptable, flexible product portfolio.
The requirements and success factors highlighted in this paper were intensively discussed and approved by experts among our industry focus group. In the case studies, first results of the application

of the approach to handle engineering changes in systems architecture design revealed positive feedback: the extensive clarification of the change case was helpful to focus further steps in the process. The network analysis of the correlations between requirements, functions and components can help in better estimation of costs and the affected areas by the change. The analysis of indirect dependencies of change artifacts with other artifacts disclosed unseen problems and helped to refine the estimation of the change impact. Also the deduction of alternative change scenarios is considerably supported by this approach. Nevertheless, the final assessment of the results of the case study will show the benefits and shortcomings of the approach presented in this paper more comprehensively.

5 DISCUSSION

Managerial Implications

Crucial for the success of a rational and sustainable decision support is the availability and accessibility to the relevant product data and transparency of the surrounding structures. Corporations should spend reasonable effort to elaborate these product data structures by adding and organizing relevant dependencies and attributes to their existing product data management systems. This would help to support workflows of system architecture design with tools providing analysis, visualization and evaluation functionality.

With these product data structures, the product development processes can be sustainably enhanced through the possibility of computer-aided decision support, workflow automation and structure optimization. The information generated in a tool like this could further be processed and adopted in managerial monitoring tools. Key indicators of system architecture could be continuously updated and monitored in order to accelerate and qualify managerial decisions.

General heuristics of system architects may fit for some decisions coincidentally. Heuristics cannot – by nature –guarantee success. Through the identification of cost drivers in the complex system and degrees of freedom, the flexibility towards the market is more transparent and can be optimized target-oriented by focusing on the most promising spots in the system structures.

Another issue in decision-making in complex environments is decisions under uncertainty. Uncertainty in decision-making may arise when a change event may come up in the future (e.g. change of requirements, upgrade of a product) under certain probability. Another aspect of uncertainty may be, that decisions are made under a *lack of information*. Probabilistic models of decision theory (Laux et al. 2012) can support the first issue coping with probability of future events. The authors of this paper assume that the second issue of lack of information can be improved by application of this framework.

Like other wide-ranging approaches to improve the corporate efficiency, the success of the framework proposed in this paper is – last but not least – heavily depended on the managerial support and strengthen of the role of system architecture design in the company.

Research Implications

Many researchers narrow the focus of their methodologies on one product and assume rather new product development than change of existing design. These assumptions facilitate the discussion and comprehension of new methodologies. Nevertheless, these assumptions only apply in very limited cases in an industrial environment. The predominant design paradigm is change existing products and structures (compare also Jarrat et al. 2012).

To allow a right degree of changeability at the strategically right spots of the product portfolio, the focus of design methodologies should not be too narrow. The system border of a system under research should be set dependent on the degree of dependencies of the complex system to its environment. In multi-variant product portfolios the dependencies between two variants of a product sharing common parts, suppliers, responsibilities etc. are simply too many to focus on only one of the products and assign methodologies only there.

6 OUTLOOK AND CONCLUSION

This paper highlights the added value of dependency management in complex decision-making environments. We clarified the role and typical tasks of system architecture design, discussed engineering changes in that domain and elaborated the nature of decision-making. Decisions can be made more rational and sustainable on the basis of transparent dependencies in complex networks like product portfolio management. Therefore, we collected a list of challenges and goals to identify success factors of decision support. The knowledge acquired from literature and discussions in the industry focus group was consolidated in a framework for decision-making in system architecture design. The next steps will be the final evaluation of the framework in six plus one industry cases. The evaluation has to prove the applicability and usability of the proposed work.

ACKNOWLEDGMENTS

Partial funding for this research work was received from the European Commission: project AMISA (Call identifier: FP7-NMP-2010-SMALL-4, Grant agreement: 262907). More information about the AMISA project is available at Website: http://www.amisa.eu/.

REFERENCES

Blees, C., Jonas, Henry & Krause, Dieter, 2010. DEVELOPMENT OF MODULAR PRODUCT FAMILIES. *Design*, (July), pp.169–182.

Blessing, L. & Chakrabarti, A., 2009. *DRM, a Design Research Methodology*, London: Springer.

Clarkson, P John, Simons, C. & Eckert, C., 2001. PREDICTING CHANGE PROPAGATION IN COMPLEX DESIGN. In *Proceedings of DETC'01*. pp. 1–10.

Eilmus, S. et al., 2012. EVALUATING A METHODICAL APPROACH FOR DEVELOPING MODULAR PRODUCT FAMILIES IN INDUSTRIAL PROJECTS. , pp.837–846.

Engel, A. & Browning, T.R., 2008. Designing Systems for Adaptability by Means of Architecture Options. *Systems Engineering*, pp.125–146.

Erixon, G., 1996. Modular Function Development (MFD)-Support for good product structure creation.

Gu, P, Xue, D & Nee, a Y.C., 2009. Adaptable design: concepts, methods, and applications. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacturing*, 223(11), pp.1367–1387.

Hamraz, B., Caldwell, Nicholas H. M. & John Clarkson, P., 2012. A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design. *Journal of Mechanical Design*, 134(10), p.100905.

Hashemian, M., 2005. *Design for adaptability*. Saskatchewan, Canada.

Haskins, C. et al., 2010. SYSTEMS ENGINEERING HANDBOOK. , (January).

Jarratt, T. a. W. et al., 2011. Engineering change: an overview and perspective on the literature. *Research in Engineering Design*, 22(2), pp.103–124.

Jiao, J. & Tseng, M.M., 2000. Fundamentals of product family architecture. *Integrated Manufacturing Systems*, 11(7), pp.469–483.

Jonas, H, Gebhardt, N. & Krause, D, 2012. TOWARDS A STRATEGIC DEVELOPMENT OF MODULAR PRODUCT PROGRAMS. , pp.959–968.

Kissel, M., Schrieverhoff, P. & Lindemann, U., 2012. Design for Adaptability – Identifying Potential for Improvement on an Architecture Basis. In *Proceedings of the NordDesign 2012, Aalborg*.

Krishnan, V. & Ulrich, K.T., 2001. Product Development Decisions: A Review of the Literature. *Management Science*, 47(1), pp.1–21.

Laux, H., Gillenkirch, R.M. & Schenk-Mathes, H.Y., 2012. *Entscheidungstheorie* 8th ed., Berlin, Heidelberg: Springer Gabler.

Li, Y., Xue, Deyi & Gu, Peihua, 2008. Design for Product Adaptability. *Concurrent Engineering*, 16(3), pp.221–232.

Plaikner, S. et al., 2012. An Architecture Framework for Multi-Product Portfolio Management in the Commercial Vehicle Industry. In *Proceedings of the NordDesign Conference 2012, Aalborg*.

Ross, A.M., Rhodes, D.H. & Hastings, D.E., 2008. Defining Changeability: Reconciling Flexibility , Adaptability , Scalability , Modifiability , and Robustness for Maintaining System Lifecycle Value. *Systems Engineering*, 11(3), pp.246–262.

Scherpereel, C.M., 2006. Decision orders: a decision taxonomy. *Management Decision*, 44(1), pp.123–136.

Schrieverhoff, P. et al., 2012. EVALUATION OF ARCHITECTURE OPTIONS IN SYSTEMS ENGINEERING. In *Proceedings of the Design Conference 2012*. Dubrovnik, Croatia.

Steffens, W., Martinsuo, M. & Artto, K., 2007. Change decisions in product development projects. *International Journal of Project Management*, 25(7), pp.702–713.

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), pp.419–440.

De Weck, O., 2011. Life-Cycle Properties of Engineering Systems: The Ilities. *Engineering*, pp.65–96.