AN INVESTIGATION OF DESIGN PROCESS CHANGES

Shapiro, Daniel; Sommer, Anita Friis; Clarkson, Peter John
University of Cambridge, United Kingdom

Abstract

Research on changes in design has focused on engineering changes, i.e. changes in the product domain, which is the manifestation of the design process. This article takes a different perspective and investigates changes in the process domain, which is characterised by the coordinated execution of design activities with complex interdependencies.
Design process changes (DPCs) comprise various types of perturbations, e.g., delays in activities, changes in customer requirements on design parameters or the addition of new activities to the process plan. This article derives a definition for DPCs, investigates their characteristics, including reasons, types and consequences, and examines existing methods to model and manage DPCs. The research is based on a systematic literature review, which is supported through an on-going industrial case study. Given both the frequent occurrence and potentially severe impacts of DPCs on process performance a set of substantial research gaps are identified and promising directions for future research on DPCs are derived.

Keywords: Design process, Uncertainty, Process change, Process modelling

Contact:
Daniel Shapiro
University of Cambridge
Engineering
United Kingdom
ds678@cam.ac.uk

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

Product development (PD) is a key function in industrial organisations and crucial for their commercial success in today's globalised markets. Fierce competition has put pressure on companies to develop cheaper products of higher quality in less time and to fulfil specific and rapidly changing customer needs, consequently increasing the product variety drastically (Holman et al. 2003). Additionally, decreasing technology life cycles and an increasing technological diversity in products have amplified the pace and complexity of PD. This has drawn much attention to the management of design processes, which encompass a spectrum of activities at the core of PD and aim at creating recipes for the production of products.

Both the dynamic and complex environment of PD as well as the inherently uncertain nature of innovative design processes lead to an industrial reality, in which engineering changes (EC), loosely defined as changes in released engineering documentation stemming from modifications of the product, are the rule and not the exception (Clark and Fujimoto 1991). Consequently, since the late 90's the interest in engineering change management (ECM) has risen strongly and manifold methods and tools for ECM have been developed (Hamraz et al. 2013a).

However, not only ECs, i.e. changes in the product domain, are likely to occur but also changes in the process domain, e.g., delays in activities, unplanned iterations or the addition of new activities to the process plan. In fact, whenever an EC arises during a design process, the process has to be amended because inputs of activities change (Wynn et al. 2014). Such design process changes (DPCs) can propagate resulting in hardly predictable consequences on major process performance metrics, particularly process duration and development cost (Lukas et al. 2007), and can ultimately affect customer satisfaction. In a study of 448 technological projects, Dvir and Lechler (2004) even found that the only distinguishing factor between successful and failed technological projects, independent of their innovativeness, is the amount of goal and plan changes during project execution. As DPCs are inevitable, Fricke et al. (2000) argued that processes should be designed with changeability in mind to enable an efficient change implementation. Also Karniel and Reich (2013, p.208) acknowledged the relevance of DPCs and observed that "the typical practice has been reactively following changes… rather than proactively planning through analysis of potential changes."

Although changes in the process domain are ubiquitous in engineering design and recognised to have severe impacts, to the authors' knowledge existing publications only cover singular aspects of DPCs – often implicitly. Therefore, this article intends to comprehensively address the issue of DPCs by deriving a definition, investigating the characteristics and discussing existing support methods for DPCs based on a systematic literature review and examples from an on-going industrial study.

The remainder of the paper is organised as follows. Section 2 explains the chosen research methodology. Section 3 derives a definition of DPCs, provides an overview on reasons for and consequences of DPCs and suggests a DPC classification. Section 4 examines methods to model and manage DPCs. Section 5 adds to the theory from the previous sections through practical DPC examples from an on-going industrial study. Section 6 discusses the major findings and derives opportunities for future research. Section 7 summarises and concludes.

2 RESEARCH METHODOLOGY

The methodology, applied in this research to investigate DPCs, followed three steps.

First, a systematic review of existing literature on DPCs was conducted. As neither a definition nor a common terminology exists for DPCs in the literature (see Section 3), the review was started with a search for the term "change" – an intentionally very broad search – in the title, abstract and key words of articles published since 2000 in nine relevant journals[1]. Articles, which were appraised to investigate DPCs based on their title and abstract, were shortlisted. Furthermore, the very comprehensive and recent literature review on ECM of Hamraz et al. (2013a) was examined for

---

[1] Research in Engineering Design, Journal of Engineering Design, Design Studies, IEEE Transactions on Engineering Management, Product Innovation Management, Computers in Industry, Systems Engineering, Artificial Intelligence for Engineering Design, Analysis and Manufacturing and International Journal of Design Engineering. These journals predominantly focus on the area of engineering design. Additional relevant publications might be found in literature focusing on project management.

relevant references on DPCs. Additionally, the identified literature was complemented through cross-referencing of key publications. Following this approach, more than 700 publications were reviewed, of which 47 were identified to be relevant, covering different aspects of DPC management (please contact the first author for a list of these publications). Certain areas of literature, which are outside of the research scope, were deliberately excluded from this review. Only literature covering the design process stages from conceptual to detailed design was reviewed. Thus, publications on changes during production and also literature on process optimisation, which usually takes place before the actual process starts, were omitted. Likewise, publications on changes in construction projects were excluded as these typically describe changes of already built artefacts and thus, are comparable to changes during production. Lastly, literature on the management of changes in workflows was omitted, as it usually emphasises the information-systems required to configure and control processes with many identical repetitive instances (e.g., customer processing at the counter).

Second, a framework for classification of different DPC types was derived based on a triangulation of findings from the identified publications and prominent process modelling literature. On the top level, this DPC classification is inspired by Browning et al. (2006), who proposed a generalised framework for modelling design processes based on two fundamental elements: activities (including their attributes) and deliverables. Consequently three major DPC types are distinguished: changes in activities, changes in deliverables and changes in activity-deliverable relationships. These major DPC types comprise further sub-types, which are completely inspired by the reviewed literature (see Section 3.2). The authors have continuously challenged this classification to ensure that the DPC types are relevant and applicable to the existing literature. Moreover to confirm the fundamental validity of this classification, informal discussions were held with experienced design researchers from the authors' research institution and from the Design Society's MMEP Special Interest Group.

Third, this research is part of an on-going case study at a leading aerospace company with the aim of modelling and investigating the consequences of DPCs. The organisation suits particularly well into this research context due to its complex and uncertain design processes, which are at the core of its business. At the time of writing, 15 meetings with designers and design managers from different departments have been conducted in preparation of a formal and detailed investigation of DPCs at this company. Five meetings focussed on scoping the research project, another five meetings elaborated on the company's current design practice and challenges, and the last five meetings aimed at creating a detailed understanding of the conceptual and embodiment design process of a specific sub-system, which was identified as suitable for further in-depth investigations. The last meetings took the form of semi-structured interviews, which were recorded and transcribed. Based on this preliminary study, practical examples for DPCs and their characteristics could be extracted to support the theoretical findings from the literature review. Both the literature-based findings and the developed classification will be further evaluated and refined within this on-going research project.

Hereafter, the discussion of DPC characteristics (see Section 3), existing support methods (see Section 4) and practical examples (see Section 5) is structured according to the identified DPC types.

3 CHARACTERISTICS OF DESIGN PROCESS CHANGES

This section discusses major DPC characteristics, investigating their reasons, types, and consequences. However, to start this discussion first an unambiguous understanding for the term DPC is necessary.

Changes in the product domain, i.e. ECs, are defined as “changes and/or modifications to released structure…, behaviour…, function…, or the relationships between functions and behaviour (design principles), or behaviour and structure (physical laws) of a technical artefact” (Hamraz et al. 2013b, p. 475). Conversely, no formal definition was found on changes in the process domain. The authors thus, suggest a definition of DPCs in analogy to the EC definition above, which is inspired by the DPC classification discussed in Section 3.2: *DPCs are changes and/or modifications, during process execution, to planned design activities (involved resources, tools, etc.), their resultant deliverables (drawings, documents, prototypes and generally descriptions of the technical artefact) or the relationships between design activities and deliverables (process structure).*

The formulation “during process execution” must be emphasised as this differentiates a DPC from process optimisation (e.g., Eppinger et al. 1994), which usually takes place before the first activity within the process has started. It is also noteworthy that intentional iterations (Browning 1999) are not regarded as DPCs if the possibility of their occurrence is accounted for in the original process plan.

This definition is not discussed further, as changes in activities, deliverables and the relationships of activities and deliverables will be comprehensively examined in the following sections.

3.1 Reasons for Design Process Changes

Eckert et al. (2004) applied the following categorisation to reasons for ECs: Emergent ECs come from unsatisfying properties of the product, e.g., from design errors. Initiated ECs also target the product's properties but come from outside of the product, e.g., from customers. Additionally, both emergent and initiated ECs can propagate leading to further ECs. An analogous differentiation could be applied to reasons for DPCs, i.e. product-related reasons (including emergent, initiated and propagated ECs), non-product-related reasons and DPC propagation.

The scarce DPC literature mostly discusses changes that originate from ECs (e.g., Chua and Hossain 2012; Li and Moon 2012; Wynn et al. 2014) and thus, have product-related reasons. When ECs occur, deliverables, which are inputs (and outputs) of design activities, need to be altered and lead to rework of the respective activities. Besides, according to Karniel and Reich (2013) the inevitable evolution of product knowledge during the design process, another product-related reason, is likely to result in structural DPCs like adding or removing design activities to/from the original process plan.

However, as the design process involves the cooperation of people across multiple organisations there are countless other reasons which do not necessarily come from the product. These DPCs can, e.g., come from process improvements suggested by project-team members or by new managers who want to leave their mark on the project (Dvir and Lechler 2004), potentially affecting coordination quality and thus, activity and process duration (Browning 1999). Another major reason for DPCs is a shortage in manpower during process execution (Dvir and Lechler 2004). This finding can be generalised to resource availability, including information, facilities and funding (Browning 1999), which was also emphasised as a major cause by the consulted researchers. Chang (2002) also identified designers'/consultants' lack of competence and too optimistic plans as important reasons for cost and schedule increases in design projects. Too optimistic plans were produced consciously due to political reasons or unconsciously due to underestimation of scope and complexity.

Lastly, the propagation of DPCs can affect activities and deliverables throughout the design process. Looking at, e.g., ECs, an activity requiring rework because of a change in its input deliverable might result in a changed output deliverable. This output serves as an input to subsequent activities. Consequently, a change propagation can occur as subsequent activities and deliverables might also require rework (e.g., Ouertani 2008). Such a propagating behaviour of DPCs is very hard to predict and can be amplified by managerial actions as demonstrated by Williams et al. (1995) based on a system dynamics model of a special-vehicle design process. Because of the increasing number of ECs and delays in change approval, project management tried to avoid overall delay through increasing process concurrency, which led to even more rework and iterations and a significant increase in overall cost.

Although no empirical research exists on the frequency of DPCs, it can be inferred that DPCs occur at least as often as ECs, as every EC leads inevitably to a DPC. Additionally, there are other possible DPC reasons as explained above. The occurrence and relevance of ECs, however, is well documented. Acar et al. (1998), e.g., surveyed UK companies and discovered that over 50% of the companies designing and manufacturing products considered ECs a major issue.

Also, despite this diversity of DPC reasons the above studies confirm that there is a limited set of direct effects on the design process, i.e. changes in activities, deliverables and structural changes. These direct effects can be regarded as major DPC types and will be elaborated in the next section.

3.2 Types of Design Process Changes

Three fundamental DPC types were derived from the systematic literature review, i.e.

1. Changes in activities (Browning et al. 2006; Chalupnik et al. 2007)
2. Changes in deliverables (Browning et al. 2006; Wynn et al. 2014)
3. Structural changes, i.e. activity-deliverable relationships (Karniel and Reich 2013)

These are named after the directly affected process element and cover further sub-types (see Figure 1).

Changes in activities refer to changes in their attributes, including start times, durations, their iterative behaviour and resource requirements.

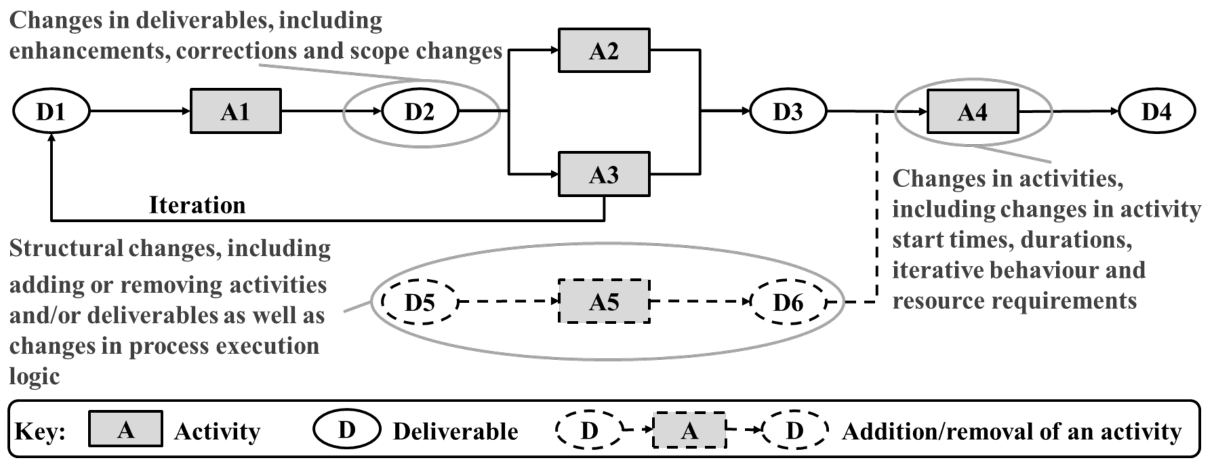

Figure 1. Major DPC types shown in an idealised design process model

Changes in deliverables comprise any changes in already produced activity inputs or outputs, including enhancements, corrections and scope changes. They are similar to ECs, which denote changes of product descriptions, i.e. geometric parameters, CAD drawings, bills of material or simulation data (Ouertani 2008), which can also be regarded as information inputs and outputs of activities. However there are two noteworthy differences between changes in deliverables and ECs. First, contrary to DPCs, ECs can also occur after the completion of the initial design process. Second, whereas ECs are defined as changes of already released product descriptions, changes in deliverables only presume that a product description has been already created.

Lastly, structural changes describe all DPCs that affect the process scheme during execution. These comprise adding or removing activities or deliverables as well as changing the process-execution logic, i.e. activity order or overlapping. Structural changes immediately affect at least one activity and one deliverable in the process. If, e.g., a new activity is added to the process, it might require existent deliverables as inputs and it will necessarily produce a new or change an existing output.

An existent classification of DPCs was only found for structural changes in Karniel and Reich 2011. This classification differentiates between seemingly similar DPCs (e.g., adding and removing forward deliverables) as different correctness verification issues can apply. This is meaningful given Karniel's and Reich's goal of automating the generation of changed processes. However, the automation aspect is not the focus of this research and thus the classification is simplified as presented above.

Although the DPC types described above denote the immediately affected process element, they can propagate to other process elements as discussed in Section 3.1. Also, multiple DPCs can affect a single design process concurrently or at different points in time.

Many other meaningful DPC classifications are conceivable. In fact, the consulted design researchers commented that a classification by reason would have been intuitive as well. However, as argued in the previous section, there are manifold reasons for DPCs with ultimately similar effects on design processes. Thus, categorising DPCs according to their immediate effect seems more practical since this leads to a simpler classification. This classification is also consistent with most activity network-based design process models, which view processes as discrete activities interconnected through deliverables. These models are commonly used to visualise, plan, execute and improve design processes (Browning and Ramasesh 2007). Therefore, the presented classification was deliberately selected to enable a particularly practicable discussion and handling of DPCs.

3.3 Consequences of Design Process Changes

Each DPC type discussed in Section 3.2 could affect every major design-process-performance metric, i.e. process duration, cost and potentially even product quality. However, it is noteworthy that only very few publications mention effects on product quality.

Regarding changes in activities, e.g., Lukas et al. (2007) showed that a decrease in an activity’s iteration-likelihood leads to a reduction in process duration, while Cronemyr et al. (2001) demonstrated how decreasing the rework-durations of activities reduces the process effort.

Examining changes in deliverables, e.g., Li and Moon (2012) concluded that increasing the arrival rate of ECs leads to an exponential growth of process duration and effort. Furthermore, the authors even

found a nearly proportional relationship of EC arrival rate and product quality, assuming that initiated ECs lead to an increase in the design-solution scope, which they associate with product quality.
Investigating structural changes, Karniel and Reich (2013) showed that adding an activity during run-time can lead to a process duration decrease, if successive changes in activity sequence are considered.
Hitherto, only DPC consequences on the holistic process level were discussed. Particularly, the literature on changes in deliverables also considers consequences on a lower level, as it describes change propagation effects from activity inputs to outputs (e.g., Chua and Hossain 2012). However, most publications only capture change propagation to improve performance predictions on the holistic process level. More granular effects, e.g., a change in a specific activity's duration following a change in another activity's attributes, are usually not analysed. Further details on DPC propagation, which is consequence of and reason for DPCs at the same time, are provided in Sections 3.1 and 4.
In summary, investigating DPCs appears highly relevant as they affect major process performance metrics and therefore the performance of the emerging product. Yet, this investigation is challenging due to non-trivial change propagation effects.

4 METHODS TO MODEL AND MANAGE DESIGN PROCESS CHANGES

This section presents a discussion of design process change management (DPCM) methods, as these dominate the existing literature on DPCs and are of particular interest to practitioners.
Change-propagation methods (e.g., Clarkson et al. 2004), which are ECM-methods that help designers assess the impacts and risks of ECs, have inspired many of the DPCM methods, reviewed in this section (e.g., Chua and Hossain 2012). Change-propagation methods focus on an *earlier* detection, *effective* prioritisation and *efficient* implementation of ECs (Hamraz et al. 2013a). Similarly, the goals of DPCM methods are accurately described by these attributes.
DPCM methods can assist in detecting propagating DPCs earlier, which might prove even more effective for some DPC types than for ECs, as the former are less dependent on potential late changes in customer requirements (obviously this holds true only for DPCs which are not induced by ECs). DPCM methods can also help assessing costs and benefits of DPC options thus, providing support for an effective prioritisation. Moreover, DPCM methods can help design managers react efficiently to an upcoming DPC by identifying the implementation path with minimal resource consumption. Besides, an efficient DPC implementation is also supported if the process is designed for changeability up-front (Fricke et al. 2000). DPCM methods can be utilised here by providing the required understanding of micro- and macro-effects of critical DPCs for restructuring the design process so that negative DPC effects are reduced and process changeability is increased.
The following sub-sections introduce relevant DPCM methods structured according to the major DPC types that they address, i.e., changes in activities, changes in deliverables or structural changes.

4.1 Methods addressing changes in activities

Five investigation methods for changes in activities were identified through the literature survey of which three particularly comprehensive methods are introduced hereafter. These methods were published in Chalupnik et al. 2007, Gärtner et al. 2008 and Khoo et al. 2003.
Many design processes experience delays in the durations of one or more activities compared to the original plan. Chalupnik et al. (2007) identified four basic possibilities how activity delays can affect design processes: (1) Absorption: activities, which are not on the critical path, can absorb delays until they become part of the critical path themselves. (2) Propagation: activities on the critical path are likely to propagate delays and can lead to resource inefficiency. (3) Accumulation: activities can accumulate delays if they iterate. (4) Negative response: delays in one activity combined with certain resource constraints can even decrease the process duration. Chalupnik et al. (2007) investigated these effects using a simulation-based method that grounds on an activity network-based design process model (Browning and Ramasesh 2007). Applying the method to a generic mechanical design process, the authors identified the activity with the largest impact on process duration if delayed and showed that the criticality-order of activities depends on iteration-likelihoods and resource constraints.
Lukas et al. (2007) and Gärtner et al. (2008) presented a matrix-based method which integrates both the product and process domain. The method accounts for activity dependencies, iteration-likelihoods, learning effects, component dependencies, dependencies between components and activities, EC timing and change extents of components. Similarly to Chalupnik et al. (2007) the method applies

Monte Carlo simulations to analyse propagating DPC effects. Lukas et al. (2007) demonstrated the applicability of the method based on a software development project, where they exemplarily examined the effects of activity duration and iteration-likelihood changes on process duration and cost. Gärtner et al. (2008) also applied this method to analyse the effects of ECs.

Khoo et al. (2003) proposed an approach, which generates a design process plan using a heuristic algorithm based on fuzzy numbers to prioritise activities under limited resources and tracks and reacts to unexpected events like activity delays, changes in resource-constraints or unexpected rework. Generated process plans include iterations from interdependent activities and account for learning effects as activities are preferably assigned to experienced designers with higher processing rates. Unexpected events are identified through deviations from the planned activity progress and trigger replanning of resourcing strategies, which can result in changed activity orders and ultimately reduce negative impacts on process duration. This method stands out as it does not rely on simulations.

In summary, the presented methods all address the consequences of changes in activity attributes, i.e. durations, resource-constraints or iteration-likelihoods, on process duration and/or cost. While the methods of Chalupnik et al. (2007) and Gärtner et al. (2008) quantify such consequences, the method of Khoo et al. (2003) additionally offers capabilities for process replanning.

4.2 Methods addressing changes in deliverables

More than half of the identified DPC literature deals with methods to assess the impact of changes in deliverables, particularly ECs, on the design process. Overall 20 methods, described by 29 publications were found. This relevance of changes in deliverables was also reflected in the comments of the consulted researchers. Three particularly comprehensive methods, published in Ouertani 2008, Chua and Hossain 2012 and Wynn et al. 2014, are introduced hereafter.

Ouertani (2008) presented the DEPNET approach that is used to identify deliverables and respective activities which require rework after the occurrence of ECs caused by resolutions of design conflicts. The DEPNET database keeps track of the on-going process and is utilised to generate a dependency network of produced deliverables. The degree of all dependencies in this network is quantified through a composed measure of three uncertainty characteristics: (1) completeness, describing how vital an input is for the creation of an output; (2) variability, expressing the change-likelihood of an input, and (3) sensitivity, describing the change extent of an output after a change in an input. In case a deliverable is changed designers can filter the dependency network and consider reworking other deliverables, which exceed a pre-defined dependency degree threshold. Moreover, various overlapping strategies for activities requiring rework to reduce rework duration and effort are discussed.

Chua and Hossain (2012) presented a simulation-based model of EC propagation, which considers likelihood mappings between change extents of inputs and outputs of activities, to assess the EC impact on process duration and effort. This propagation model is combined with a scheduling model, which accounts for the initially affected activity, change timing and rework durations, and which is able to simulate adapted process schedules. As the original schedule already includes rework coming from activity overlapping the additional rework stemming from ECs might be dampened depending on the change timing. Hence, design managers can decide on the best change implementation timing to reduce the impact on overall schedule by comparing new and original schedules.

Wynn et al. (2014) proposed the CPiW framework, which models rework sequence, resource requirements and duration of a change process, triggered by one or more ECs and their propagation effects. CPiW applies Monte-Carlo simulations and accounts for uncertain activity durations, concurrency, iterations as well as change-propagation likelihoods between activities and their outputs. Additionally, separate sensitivities to changes in inputs are considered for activity outputs and durations. Due to potentially multiple points of change initiation, complex process structures and resource constraints some inputs can accrue multiple changes before their respective activity is executed. Such changes are offset against each other to account for a possible overlap of change scopes.

In summary, the presented methods all address the consequences of changes in deliverables and their propagation through the design process. While the method of Ouertani (2008) supports designers in identifying activities and deliverables that require rework, the methods of Chua and Hossain (2012) and Wynn et al. (2014) also quantify the additional duration and cost incurred through this rework.

4.3 Methods addressing structural changes

Research on structural DPCs was mainly conducted by Karniel and Reich (2011, 2013) with the aim to automatically adapt processes after change occurrence so that they are logically correct and executable without requiring further verification. Their work grounds on transforming Design Structure Matrices (DSM) through adding logic activities into DSM nets, which are equivalent to a specific Petri net type for workflows: WRI-WF nets. This transformation is required because the direct translation of iterative DSMs into Petri net process plans might generate logically incorrect plans, e.g., containing deadlocks. Additionally, to ensure unique translations of DSMs into process plans business rules are applied, which determine, e.g., whether a downstream activity needs to wait until all iterations of the upstream activity are finished. While DSMs are utilised for static planning, DSM nets can be used for run-time process execution. If a structural change occurs an adapted process is automatically generated through a hierarchical expansion of the DSM net, accounting for the current process state and additional business rules for change. The automatically adapted process plan can then be simulated to assess the consequences of the structural change on process duration.

5 DESIGN PROCESS CHANGE EXAMPLES FROM INDUSTRIAL PRACTICE

The previous sections derived a definition for DPCs, investigated their characteristics and examined existing support methods based on a systematic literature review. This section supports these theoretical findings through presenting practical examples of DPCs drawn from an on-going case study at a leading aerospace company. Reasons, types and consequences of DPCs are discussed. As one designer noted the design teams permanently "live in a state of uncertainty" and thus, DPCs are very common. Hence, this section is limited to one characteristic example per major DPC type, i.e. changes in activities, deliverables and structural changes (see Section 3.2).

A typical example for changes in activities was found in the preliminary (i.e. conceptual and embodiment) design process of a sub-system, which will be further examined in this research project. Multiple designers involved in this process reported activity delays that are caused by the occupation of engineers through parallel projects and responsibilities - a reason, which is clearly not product-related. To avoid a subsequent increase in the overall process duration the design team would speed up the downstream process potentially leaving out some optional, explorative design iterations and thus, ultimately trading off duration against confidence into the design's quality. The role of a multi-project-environment as a trigger for DPCs was also stretched by the consulted researchers.

Moreover, various meetings brought up examples of changing requirements on design parameters during the preliminary design. Such changes could occur multiple times during a design process and would be either triggered directly by the customer, or by internal departments. Such DPCs are characteristic examples for changes in deliverables, which stem from the product domain. Depending on the affected requirement and the time point of its change, different extents of rework with corresponding effects on process duration and cost were observed.

Lastly, an experienced design manager also provided an example for a structural change, in which an evaluation activity that was not scheduled initially is added to a design process in order to reduce the risk associated with a certain design parameter. Although this change is not directly caused through unsatisfying properties of the product it is rather product-related. Adding an additional evaluation activity directly increases process cost and potentially also duration but enhances the confidence into the design's quality. As such an evaluation activity might also trigger rework in case of unsatisfactory test results it can additionally have an indirect effect on process cost and duration.

Initial discussions revealed that the organisation uses common project management software and activity network-based process maps but no specific methods or tools to manage DPCs.

6 DISCUSSION AND OPPORTUNITIES FOR FUTURE RESEARCH

The investigations on DPC characteristics and existing support methods led to interesting findings.

Firstly, from the manifold reasons that can trigger DPCs, including the very common phenomena of EC, it can be inferred that DPCs are omnipresent in engineering design processes. Secondly, the synthesis of studied DPC effects across relevant publications and industrial examples made clear that DPCs can have severe impacts on design process performance, particularly on process duration and cost and potentially even on product quality. From a risk management perspective where the risk associated with an incident is viewed as the product of its likelihood of occurrence and its impact (UK Ministry of Defense 1996), it is therefore argued that DPCs carry high risks for design processes. Still, relevant publications in engineering design usually discuss DPCs only implicitly.

Moreover, compared to the rich literature on ECM (Hamraz et al. 2013a) there is a lack of comprehensive methods to support management of DPCs. In fact, only 26 methods described by 42 publications were identified, among which 17 solely examine effects of changes in deliverables. Among the other nine methods there is none which explicitly covers investigations of all three major DPC types (see Section 3.2). Also, the reviewed DPCM methods offer highly varying features: Some methods, e.g., can help identifying activities affected by a DPC (Ahmad et al. 2013), while other methods additionally suggest an activity sequence for DPC implementation (Ouertani 2008). Or, some methods explicitly consider DPC-implementation timing (Chua and Hossain 2012), while others simply assume that DPCs affect already completed activities (Wynn et al. 2014).

One common feature among most DPCM methods (although treated in different ways) is the representation of change propagation. In fact, complex propagation paths across all of the major DPC types are conceivable. For example, a change in the effort invested in a certain design activity could lead to a change in the confidence in its own output deliverable as well as the output deliverables of directly and indirectly dependent activities. To revert to the planned confidence level activities might need to be reworked or test activities might be added or removed from the original process plan. Predicting the effects of such propagation paths on process performance is not trivial and requires the definition of new metrics and analysis methods.

Based on these insights, the following two directions for future research seem particularly valuable:

- There is a need for empirical research of DPCs across different industries. Such studies should examine the occurrence of DPCs, their reasons and consequences including managerial reactions. Particular emphasis should be attributed to DPC propagation paths through the process.
- There is a need for the development of comprehensive DPCM methods, which support design managers and their teams in investigating the effects of and possible reactions to different DPC types in order to manage design process performance during run-time. Such support methods should be based on systematically derived requirements and take into account the propagating behaviour and uncertainty associated with complex design processes.

7 CONCLUSION

Research on changes in design has focused on engineering changes, i.e. changes in the product domain, which is the manifestation of the design process. This article takes a different perspective and investigates changes in the process domain, which is characterised by the coordinated execution of design activities with complex interdependencies. Such design process changes (DPCs) comprise various types of perturbations that can affect the performance of engineering design processes during run-time, e.g., delays in activities, changes in design parameters or the addition of new activities.

The contributions of this article are threefold: First, it suggests a concise definition for DPCs. Second, it derives a classification of DPCs into three major types. Third, it provides a comprehensive overview on DPC characteristics, i.e. reasons and consequences, and on existing support methods.

The investigation leads to the following key insights:

- DPCs can be caused due to a wide range of reasons but can be fundamentally classified into changes in activities, changes in deliverables and structural changes. Due to complex change propagation mechanisms DPCs have hardly predictable effects on major process performance metrics, i.e. process duration, cost and even product quality. Explicit empirical research on DPC characteristics, including reasons, propagation paths and consequences, is clearly needed.
- Overall, only 26 existing support methods were found, none of which address the three major DPC types comprehensively. All methods comprise different features and offer varying degrees of support. Therefore, there is a need for the systematic development of comprehensive support methods, which help design managers understand and react to the consequences of DPCs.

Fellow researchers can directly benefit from this investigation, as it clearly indicates important research gaps and lays a foundation to systematically address these by providing a useful terminology on DPCs. In addition, practitioners can benefit from this research by understanding reasons and consequences of DPCs and by receiving an overview of the state-of-the-art in support methods.

REFERENCES

Acar, B.S., Benedetto-Neto, H., Wright, I.C. (1998) 'Design change: Problem or opportunity', in Engineering Design Conference '98, Brunel University, UK, 445–454.

Ahmad, N., Wynn, D.C., Clarkson, P.J. (2013) 'Change impact on a product and its redesign process: a tool for knowledge capture and reuse', Research in Engineering Design, 24(3), 219–244.

Browning, T.R. (1999) 'Sources of schedule risk in complex system development', Systems Engineering, 2(3), 129–142.

Browning, T.R., Fricke, E., Negele, H. (2006) 'Key concepts in modeling product development processes', Systems Engineering, 9(2), 104–128.

Browning, T.R., Ramasesh, R.V. (2007) 'A survey of activity network-based process models for managing product development projects', Production and Operations Management, 16(2), 217–240.

Chalupnik, M.J., Wynn, D.C., Eckert, C.M., Clarkson, P.J. (2007) 'Understanding design process robustness: A modelling approach', in 16th International Conference on Engineering Design, ICED 2007, Paris, France, 455–456.

Chang, A.S.T. (2002) 'Reasons for cost and schedule increase for engineering design projects', Journal of Management in Engineering, 18(1), 29–36.

Chua, D.K.H., Hossain, M.A. (2012) 'Predicting change propagation and impact on design schedule due to external changes', IEEE Transactions on Engineering Management, 59(3), 483–493.

Clark, K.B., Fujimoto, T. (1991) Product development performance: Strategy, organization, and management in the world auto industry, Harvard Business School Press: Boston, MA, USA.

Clarkson, P.J., Simons, C., Eckert, C. (2004) 'Predicting change propagation in complex design', Journal of Mechanical Design, 126(5), 788.

Cronemyr, P., Öhrwall Rönnbäck, A., Eppinger, S.D. (2001) 'A decision support tool for predicting the impact of development process improvements', Journal of Engineering Design, 12(3), 177–199.

Dvir, D., Lechler, T. (2004) 'Plans are nothing, changing plans is everything: The impact of changes on project success', Research Policy, 33(1), 1–15.

Eckert, C., Clarkson, P.J., Zanker, W. (2004) 'Change and customisation in complex engineering domains', Research in Engineering Design, 15(1), 1–21.

Eppinger, S.D., Whitney, D.E., Smith, R.P., Gebala, D.A. (1994) 'A model-based method for organizing tasks in product development', Research in Engineering Design, 6(1), 1–13.

Fricke, E., Gebhard, B., Negele, H., Igenbergs, E. (2000) 'Coping with changes: Causes, findings, and strategies', Systems Engineering, 3(4), 169–179.

Gärtner, T., Rohleder, N., Schlick, C.M. (2008) 'Simulation of product change effects on the duration of development processes based on the DSM', in 10th International Design Structure Matrix Conference, DSM'08, Stockholm, Sweden, 199–208.

Hamraz, B., Caldwell, N.H.M., Clarkson, P.J. (2013a) 'A holistic categorisation framework for literature on engineering change management', Systems Engineering, 16(4), 473–505.

Hamraz, B., Caldwell, N.H.M., Wynn, D.C., Clarkson, P.J. (2013b) 'Requirements-based development of an improved engineering change management method', Journal of Engineering Design, 24(11), 765–793.

Holman, R., Kaas, H.-W., Keeling, D. (2003) 'The future of product development', McKinsey Quarterly, (3), 28–39.

Karniel, A., Reich, Y. (2011) Managing the dynamics of new product development processes: A new product lifecycle management paradigm, Springer: London, UK.

Karniel, A., Reich, Y. (2013) 'Multi-level modelling and simulation of new product development processes', Journal of Engineering Design, 24(3), 185–210.

Khoo, L.P., Chen, C.-H., Jiao, L. (2003) 'A dynamic fuzzy decision support scheme for concurrent design planning', Concurrent Engineering Research and Applications, 11(4), 279–288.

Li, W., Moon, Y.B. (2012) 'Modeling and managing engineering changes in a complex product development process', The International Journal of Advanced Manufacturing Technology, 63(9-12), 863–874.

Lukas, M., Gärtner, T., Rohleder, N., Schlick, C.M. (2007) 'A simulation model to predict impacts of alterations in development processes', in 9th International Design Structure Matrix Conference, DSM'07, Munich, Germany, 127–136.

Ouertani, M.Z. (2008) 'Supporting conflict management in collaborative design: An approach to assess engineering change impacts', Computers in Industry, 59(9), 882–893.

UK Ministry of Defense (1996) Defense Standard 00-56 (PART 1)/Issue 2: Safety Management Requirements for Defense Systems, Glasgow, UK.

Williams, T., Eden, C., Ackermann, F., Tait, A. (1995) 'The effects of design changes and delays on project costs', The Journal of the Operational Research Society, 46(7), 809–818.

Wynn, D.C., Caldwell, N.H.M., Clarkson, P.J. (2014) 'Predicting change propagation in complex design workflows', Journal of Mechanical Design, 136(8), 081009–1 – 081009–13.