# METHODS OF CHANGE IMPACT ANALYSIS FOR PRODUCT DEVELOPMENT: A SYSTEMATIC REVIEW OF THE LITERATURE

**Mordaschew, Viktoria;**
**Herrmann, Jan-Phillip;**
**Tackenberg, Sven**

OWL University of Applied Sciences and Arts

## ABSTRACT

During product development, the customer or internal stakeholders initiate changes concerning the components or functions of a cyber-physical system (CPS). The complexity of such a CPS causes difficulties in evaluating the effects of a component change. Accordingly, product developers need an assistance system to quantify the impact of a component change on hardware, software, system functions, and production processes. Therefore, this paper focuses on concepts to evaluate the effects of component, functional, and process changes and contributes to its clarification and further understanding of the importance and requirements for such an assistance system. The literature review assesses the identified methods regarding their objectives, application objects, level of automation, and relations characteristics. However, the literature review pointed out that the change prediction method from Clarkson et al. (2004) is well-established in the literature and able to quantify the impact of a change.

**Keywords**: Product Lifecycle Management (PLM), Change Impact, Complexity, Uncertainty

**Contact**:
Mordaschew, Viktoria
OWL University of Applied Sciences and Arts
Germany
viktoria.mordaschew@th-owl.de

**Cite this article:** Mordaschew, V., Herrmann, J.-P., Tackenberg, S. (2023) ‘Methods of Change Impact Analysis for Product Development: A Systematic Review of the Literature’, in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.266



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

# 1 INTRODUCTION

Cyber-physical systems (CPS) are products that consist of a significant number of electrical, mechanical, and software-technological components (Drossel et al., 2018). The complexity of such a CPS results from many parts and their relations with each other (Scheinpflug, 2017). These relations entail structural and functional dependencies between components. A CPS passes through different product life cycle phases. During product development, the customer or the internal stakeholders of the product development process initiate changes concerning the product's components or functions. Due to the many relations between hardware and software components and production processes, assessing the effects of a product or process change is difficult. Consequently, there is a risk that product and process developers will incorrectly evaluate the impact of changing requirements or replacing components on a product. Accordingly, the effect on the expected costs or the time required for a change can, at best, be estimated based on empirical knowledge with a high degree of uncertainty.

Matrices are suitable for formally describing effective relations between components or processes. These are also named Design-Structure-Matrices (DSM) for describing product and process characteristics (Steward 1981). A DSM is a square matrix with which complex systems can be described and analysed. Therefore, a DSM enables product developers to understand the complex relations between components of a CPS and between parts and processes. In literature, approaches exist that analyse or quantify the impact of a change using a DSM. However, an initial preliminary study revealed that only a few graphical assistance systems exist for quantitative analysis of product and process changes based on a DSM (Eckert et al., 2011). Thus, there is a need for more assistance systems for product and process developers to evaluate the impact of component and process changes.

This paper presents a systematic literature review (SLR) to identify suitable concepts for evaluating the effects of component, functional, and process changes. Of interest are approaches that enable product developers and process managers to analyse a product's structural and functional dependencies. Further, publications are introduced that determine the impact of a change between the domains of software, hardware, system function, and production processes. Relevant articles for change impact analysis are shortly described regarding objectives, application objects, level of automation, and relations characteristics. Finally, the identified methodological and technical approaches are evaluated in terms of adoption for an assistance system to be developed.

# 2 SYSTEMATIC LITERATURE REVIEW

## 2.1 Methodology

The literature review was conducted from 09.06.2022, until 05.07.2022, in German and English language. It includes all journal articles and conference papers in accepted or higher publication status. The databases Web of Science - Core Collection (WoS-CC) (webofscience.com) and Scopus (scopus.com) were used to review international publications. The database WISO (wiso-net.de) was utilized to identify publications from German institutions and researchers. The SLR aims to identify concepts that can be used to develop an assistance system to analyse the impact of a component change for other components and functions of the CPS. The approach must be based on a DSM to enable a mathematical description and simulation of effective relationships for a CPS. Thus, this paper considers publications that fulfil all the following criteria:

- The concept focuses primarily on a product, CPS, or software.
- The approach uses a DSM or extension.
- The article analyses, quantifies, or ranks the change impact a component has on other
- components.
- Product and process developers can use the concept in all phases of the product lifecycle.

The search strategy consists of two parts. The first part mainly focuses on the term DSM. The search strategy also uses the synonyms Dependency-Structure-Matrix and Dependency-Source-Matrix to obtain more search results (Browning, 2001). The second part of the search strategy consists of the term change and the word family of prediction or propagation. The search strategy is defined as



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

((Design AND structure AND matrix) OR (Dependency AND structure AND matrix) OR (Dependency AND source AND matrix)) AND (Change AND (Propagat* OR Predict*)). In WiSo the search strategy is slightly adjusted to adapt better to German articles. To check whether the search strategy is suitable, the search results obtained in WoS-CC are displayed graphically using VOSviewer, checking the co-occurrence of keywords. VOSviewer is a free program that can be used to perform bibliometric analyses and display their results in images. These images are created using the "visualization of similarities" method, also known as the VOS method (van Eck and Waltman, 2010). VOSviewer analyses whether the keyword is present in an article or not. The size of the circle is proportional to the frequency of usage of the keyword. In addition, keywords that occur together in articles are connected by lines. The more often these keywords occur together, the thicker the lines are. Figure 1 shows the co-occurrence of keywords in the articles of the WoS-CC search results. Only keywords that occur in at least twelve articles are shown. Since the keywords "change propagation", "impact", and "DSM" occurred quite often together, the search term gave confidence in its suitability.

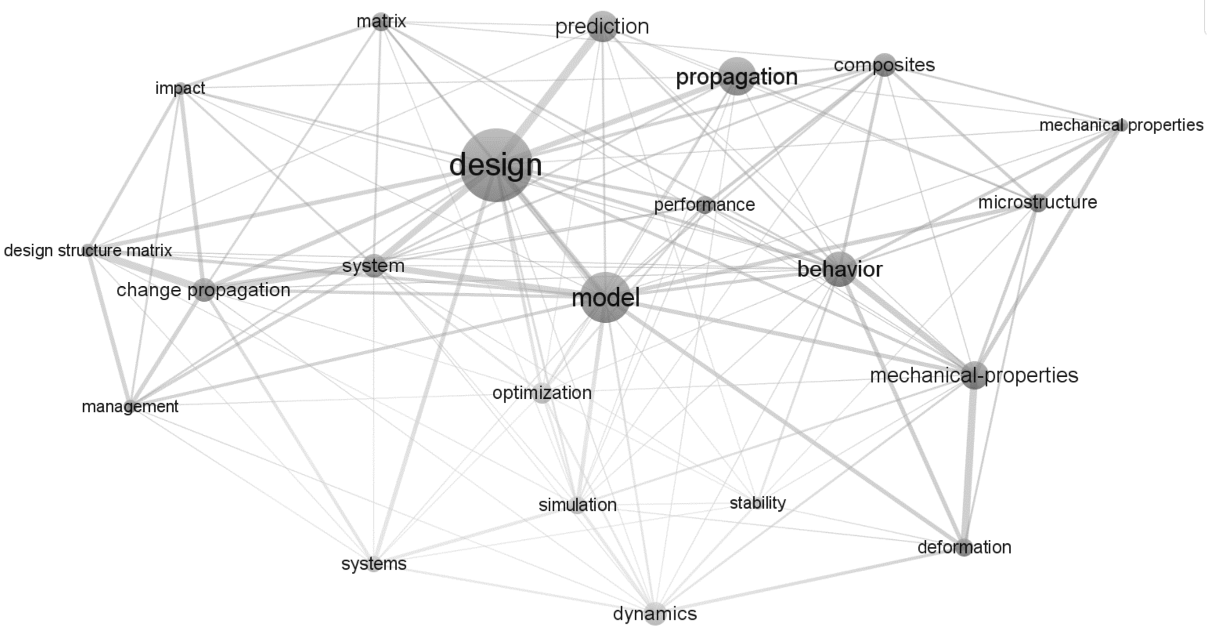


*Figure 1. Co-occurrence of keywords*

The search pass ascertained 563 (WoS-CC), 605 (Scopus), and 704 (WiSo) publications. Afterwards the title and abstract of each publication were screened. A publication had to contribute to all the criteria stated in the paragraph above. Subsequently, the full text of the selected publications was evaluated for eligibility. For WiSo, the search strategy was applied for full text. If a publication was identified in different databases using the search criteria, it was only included once for the literature review. The search in the three databases resulted in 19 articles, which are shown in Table 1. Figure 2 shows the search strategies used and the selection process in detail.

*Table 1. Identified articles*

| | | |
|---|---|---|
| [1] Chen and Whyte, 2021 | [8] Hamraz et al., 2012 | [14] Oduncuoglu and Thomson, 2011 |
| [2] Chen at al., 2010 | [9] Kim et al., 2014 | [15] Rajinia and Li, 2010 |
| [3] Cheng et al., 2022 | [10] Koh et al., 2015 | [16] Saoud et al., 2017 |
| [4] Eckert et al., 2011 | [11] Maier et al., 2014 | [17] Schlick et al., 2007 |
| [5] Flanagan et al., 2003 | [12] Morkos et al., 2013 | [18] Schneider et al., 2012 |
| [6] Fu et al., 2012 | [13] Nonsiri et al., 2013 | [19] Zhao et al., 2008 |
| [7] Hamraz et al., 2013 | | |



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

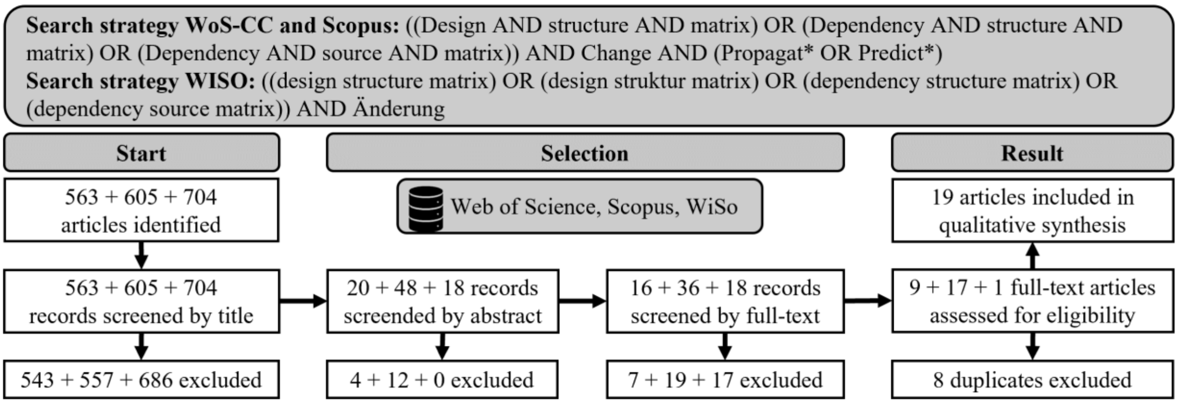


*Figure 2. Search terms and paper selection process*

The intra- and interrater reliability were calculated using Cohen's kappa (κ), to assess the repeatability and reproducibility of the SLR. For the intrarater reliability, the SLR was conducted two times. A second rater conducted the SLR for the databases WoS-CC and Scopus for the articles' title and abstract exclusion criteria to calculate the interrater reliability. κ and the confidence intervals were calculated for the inter- and intrarater reliability using the formulas from McHugh (2012). Figure 3 shows the data for calculating the intra- and interrater reliability. If an article was excluded from the SLR, it was assigned a value of "0". Included publications were assigned a value of "1". The confidence level for the κ calculation of the intra- and interrater reliability was estimated at 95%. This results in a value of 0.93 for the intrarater reliability with a confidence interval from 0.86 to 1. The κ value for the interrater reliability is 0.88, with a confidence interval from 0.81 to 0.95. Values between 0.8 and 0.9 state that the matching between the first and the second time or between rater one and rater two is high (McHugh, 2012). Consequently, κ indicates that the conducted SLR is reproducible and repeatable with the set-up inclusion criteria.

| | | **First time** | | Σ |
|---|---|---|---|---|
| | | 0 | 1 | |
| **Second time** | 0 | 1842 | 1 | 1843 |
| | 1 | 3 | 26 | 29 |
| Σ | | 1845 | 27 | **1872** |

| | | **Rater 1** | | Σ |
|---|---|---|---|---|
| | | 0 | 1 | |
| **Rater 2** | 0 | 1113 | 7 | 1120 |
| | 1 | 4 | 44 | 48 |
| Σ | | 1117 | 51 | **1168** |

*Figure 3. Data for calculating the intra- (left) and interrater reliability (right)*

## 2.2 Results

A CPS consists of hardware and software that provide the necessary functions. Therefore, the literature review must provide a method to describe structural and functional relations. Ideally, an approach also displays the relations between the CPS and the production processes. Further, the concept allows the description of direct and indirect dependencies occurring with a change. At best, the change impact is quantified so developers can estimate the likelihood of a change propagation to other components. Therefore, an assistance system must calculate the change impact with the lowest possible input. Thus, this paper provides a qualitative content analysis using a deductive category system based on the defined functions an assistance system must provide. Therefore, the following aspects are analysed and presented:

- Application Object (the object the method was used on).
- Objectives (the aim the method targets at).
- Type of dependencies (the dependencies the method displays and analyses).
- Elements displayed (the elements of which the dependencies are shown in the DSM).
- Level of automation (whether the analysis is executed per hand or per computer).
- Evaluation method (whether the approaches were evaluated and if so, which methods were used).



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

**Application Object**

The application object describes the category of object, for which the approach was developed. It does not represent the kind of elements used in the DSM. The application object must be a complex product that combines mechanical, electrical, and software-technological components, ideally a CPS. Most of the articles use a complex product as an application object. For example, Hamraz et al. (2013) introduce a method to predict the combined change risk with an algorithm that does not include cyclic change propagation paths. Therefore, the uncertainty in anticipating the changing risk is minimized for developers. Nevertheless, those approaches do not contain software. Only two articles deal explicitly with software. Fu et al. (2012) describe the change impact during software development while simulating the changed project duration and cost. The functional dependencies between the system architecture elements are shown in a DSM, and the combined risk is calculated using the change prediction method from Clarkson et al. (2004). The developer assigns every element of the system architecture to the minimal, likely, and maximal cost and schedule values. As a result, a simulation of the change propagation paths generates the estimated cost and schedule of the software project. However, this approach only includes software components and does not include the hardware of a CPS or its production processes.

Four of the identified articles deal with complex projects. Zhao et al. (2008) simulate a construction project's progress over time with a Monte-Carlo simulation. Therefore, they use process-DSMs, which calculate the rework probability. The Monte-Carlo simulation calculates all necessary changes and the demanded rework. Saoud et al. (2017) determine the impact of change on a building component. The necessary DSMs are constructed automatically from the Building Information Model and used for change prediction. As an output, this method provides a probability value for components affected by the initial change. However, none of these approaches enables the developers to display indirect changes or the impact of the initial change for a CPS and its production processes. Table 2 shows a summary of all application objects.

*Table 2. Application objects*

| Application objects | | | |
|---|---|---|---|
| Complex products | Complex projects | Requirements | Software |
| [1] [3] [4] [5] [7] [8] [9] [10] [11] [14] | [16] [17] [18] [19] | [2] [12] [13] | [6] [15] |

**Objectives**

The objectives considered in this paper relate to the quantitative assessment of a change impact. Developers should be empowered to estimate direct and indirect change effects on the components of a CPS. Eleven approaches fulfil this goal. Three articles present a method for ranking change options from most to less impact on the object. Therefore, developers can rank change decisions against each other. This results in the possibility of estimating the change impact. Rajinia and Li (2010) present a method that displays an object-oriented software program based on a static DSM and rates different solutions for a software change, which are scaled with priority numbers. Therefore, the instances of an object-oriented software program are identified and displayed as a matrix element. The dependencies between these elements are binary. The method calculates priority numbers for every possible change propagation path of the initial change. The less the priority number, the fewer instances of the software program are affected. Due to this method, developers can choose the change option, which affects the software program less. Koh et al. (2012) use an engineering change forecast to identify a change's influence on a system module and how strongly it affects other system modules. The calculated Engineering Change Forecast Index ranks which component change mainly influences the system. Therefore, developers should consider modularizing these influential modules to minimize their change impact. Another four articles evaluate the expected project duration and the estimated cost, which does not provide an understanding of change's effects on other CPS components. Maier et al. (2014) calculate the project duration with a Monte-Carlo simulation, including progressive iterations, rework and resource constraints. Changes are only executed when an employee is available for the task. The maturity level describes the state of progress of each component. If a component is changed, the maturity level rises. If rework is necessary on a component, its maturity level descends. The Monte-Carlo simulation ends when all components reach their highest possible maturity level.



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

Afterwards, the estimated project duration is displayed with their respective likelihood, enabling project managers to make assumptions about the project duration in the early phases. Schlick et al. (2007) use a Work Transformation Matrix to estimate the expected project duration. The dependencies between parallel-executed activities are displayed. The diagonal cells contain the work progress of tasks if independently processed. The off-diagonal cells describe whether those activities are unaffected, negatively, or positively correlated. A range of values between -1 and 1 represents the characteristics of the relation. The expected progress for each period is calculated and shown in a graph. The article of Oduncuoglu and Thomson (2011) is the only one identified that primarily classifies the change type. The dependencies between the design functions and the product components are displayed, and the change risk is calculated with the change prediction method from Clarkson et al. (2004). Afterwards, the risk of change for each design function is classified into the types "ripple", "blossom", and "avalanche". Thereby, this approach does not fulfil the aim expected from the identified approaches. Table 3 shows the objectives of the methods in the identified articles in detail.

*Table 3. Objectives of the articles*

| **Objectives** | | | |
|---|---|---|---|
| Predicting change propagation | Forecasting project cost or duration | Choosing a change alternative | Identifying the change type |
| [1] [2] [4] [5] [6] [7] [8] [9] [12] [13] [16] | [11] [17] [18] [19] | [3] [10] [15] | [14] |

**Type of dependencies and elements displayed in the DSM**

A developer must know the functional and the structural dependencies between components of a CPS to evaluate the impact of a change. A change in the CPS can affect hardware as well as software components. Additionally, a change can affect the functions of a CPS. Moreover, the change impact must be evaluated for the whole product lifecycle. Further, the processes and requirements must be evaluated. Therefore, an extension of the classic DSM, e.g. a Multiple-Domain-Matrix (MDM) (Maurer and Lindemann, 2007), is needed. For example, Flanagan et al. (2003) use a Domain-Mapping-Matrix (DMM) (Danilovic and Browning, 2007) to display functional dependencies between the system functions and the system components. A feature describes a function that is executed by a specific system component. The direct dependencies of features are displayed by creating a so-called feature-feature matrix Consequently, the indirect dependencies can be determined and displayed in change paths. This way, developers can trace the indirect dependencies with this matrix and identify the change propagation paths. Hamraz et al. (2012) use a function-behaviour-structure linkage model to trace functional and structural dependencies in a MDM. The method connects the functions and structures of a system through the system's behaviour. The change prediction method (Clarkson et al., 2004) calculates the combined risk for these three domains.

In general, seven articles use structural and functional dependencies in their approach. The article by Chen et al. (2010) is the only one of the identified articles to address non-functional requirements. The dependencies of non-functional and functional requirements in software projects are represented by a DMM, where functional requirements are represented by use cases. A ranking is used to show which non-functional requirements have the greatest influence on the system. Table 4 shows the types of dependencies identified in the approaches and Table 5 shows which elements the dependencies display.

*Table 4. Used types of dependencies*

| **Type of dependencies** | | |
|---|---|---|
| Functional | Structural | Functional and structural |
| [2] [5] [6] [12] [13] [14] [15] [17] [19] | [4] [7] [11] | [1] [3] [8] [9] [10] [16] [18] |


https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

*Table 5. Displayed elements in the DSM*

| Objects displayed in the DSM | Article | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** |
| Hardware - Hardware | x | | x | x | x | | x | x | x | x | x | | | x | | x | | x | |
| Software - Software | | | | | | | | | | | | | | | x | | | | |
| Requirement - Hardware | | | | | | | | | | x | | | | | | | | | |
| Requirement - Software | | | | | | x | | | | | | | | | | | | | |
| Requirement - Require. | | | | | | | | | | | | x | x | | | | | | |
| Requirement - Use case | | x | | | | | | | | | | | | | | | | | |
| Function - Hardware | | | | | x | | | x | | | | | | x | | | | | |
| Function - Function | | | | | x | | | x | | | | | | | | | | | |
| Behaviour - Hardware | | | | | | | | x | | | | | | | | | | | |
| Behaviour - Function | | | | | | | | x | | | | | | | | | | | |
| Behaviour - Behaviour | | | | | | | | x | | | | | | | | | | | |
| Process - Hardware | | | | | | | | | | | | | | | | | | x | |
| Process - Process | | | | | | | | | | | | | | | | | x | x | x |
| Parameter - Hardware | | | | | | | | | | | | | | | | | | x | |
| Parameter - Parameter | | | | | | | | | | | | | | | | | | x | |

### Automation level

Fully automated methods only need input to generate an output. An assistance system should automatically process the required calculations and graphics for a change impact analysis. A high degree of automation adds value for a developer. Otherwise, evaluating and tracing the dependencies, especially of large CPS, would require too much time for developers. Chen and Whyte (2021) use such a method, which uses IFC files of a digital twin to predict the impact of an engineering change. Therefore, they combine the change prediction method from Clarkson et al. (2004) with fuzzy linguistics and a load-capacity model. Components are clustered into system modules by an algorithm so that changes in one module affect other modules as less as possible. Cheng et al. (2021) use a partly automated method, quantifying the change impact between system modules. Therefore, influential modules are identified through an algorithm so developers can avoid changes in these modules. Schneider et al. (2012) present a non-automated approach that enables the prediction of expected project duration and costs. For this purpose, a DSM is created for each of the domains process, parameter and product. By linking parameters, components and processes, the expected durations required for component changes can be determined, since these changes depend on time-consuming processes. The expected project duration can be calculated by determining this required time, from which the expected project costs can then be calculated. Table 6 shows the level of automation for every article.

*Table 6. Automation level of the approaches*

| **Level of automation** | | |
|---|---|---|
| Fully automated | Partly automated | Not automated |
| [1] [4] [7] [11] [16] [19] | [3] [5] [6] [8] [14] | [2] [9] [10] [12] [13] [15] [17] [18] |

### Evaluation method

The overall goal is to identify concepts, which can be adapted into an assistance system. Therefore, the approaches should be evaluated with a case study and ideally qualitatively and quantitatively. All publications, except two, presented a case study. Koh et al. (2015) conducted as only ones a case study as well as a quantitative and qualitative evaluation. Maier et al. (2014) validated their approach qualitatively. Therefore, they used eleven examples of different products from the literature. Table 7 shows the used kind of evaluation.



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

*Table 7. Evaluation methods used for the approaches*

| Evaluation method | | | |
|---|---|---|---|
| None | Case study | Qualitative method | Quantitative method |
| [2] [18] | [1] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [19] | [10] [11] | [7] [10] [15] [17] |

**Change Prediction Method**

Nine of the 19 articles use the change prediction method from Clarkson et al. (2004). Eckert et al. (2011) developed a software tool that automatically calculates the change risk based on the input DSM. Furthermore, the software tool can show the propagation paths of a change or a diagram, which classifies the product components into propagation multipliers or absorbers. Such a tool enables developers to understand the dependencies existing in a product. In summary, many of the identified approaches use the change prediction method. Therefore, this method can contribute to correctly assessing the impact of changes. Consequently, it is reasonable to implement this method in an assistance system to estimate the impact of component and process changes. However, the change prediction method must be modified to evaluate the change impact for a CPS and its corresponding production processes. One of these modifications includes the representation of the functional dependencies between the software and hardware components of the CPS. Only then product developers are enabled to assess the change impact in a CPS properly.

**Identified limitations**

The following limitations and gaps in research were identified with the SLR:

- No impact analysis was identified that maps hardware and software together with production processes.
- In addition, no impact analysis was identified that explicitly separates hardware and software components and depicts their impact on each other.
- Furthermore, no approach depicts the production processes that may need to be adapted, but rather the processes and activities that need to be performed in the development or change projects.
- The articles do not address the additional development time required that is triggered by a change, but rather calculate the total duration needed for the whole development process.
- In many approaches, adding or removing components is not considered and not possible to display with the developed concepts, e.g. in the article from Saoud et al. (2017).

## 3 CONCLUSION AND FUTURE WORK

This paper provides an overview of existing approaches to trace, classify or quantify the change impact. In conclusion, some methods enable product developers to understand the impact of a change on a product or a software function. Further, some approaches allow estimating the project duration and cost. Nevertheless, none of the identified papers presents a fully automated method for describing the relations between software and hardware components of a CPS and its production processes concerning the influence of changes in requirements. Therefore, developers need more than the methods to fully assess a CPS's functional and structural interdependencies and production processes. Thus, there needs to be a comprehensive theoretical concept for the envisioned assistance system for evaluating the impact of a change. Due to this research gap, there is uncertainty in product development projects regarding the effects of changes. As a result, when a change occurs, the components and functions to be changed often need to be fully identified. Accordingly, the effort required for a change is assumed to be too low. Future work will therefore require the development of a method for evaluating the impact of changes on all relevant assets of a CPS. In addition to the parts, hardware and software functions and a CPS's underlying production processes must be considered. Thus, developers can comprehensively analyse dependencies between a product's components, functions, and production processes. Knowledge of the dependencies can be used to reduce the number of iterations and the resulting rework.



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

## ACKNOWLEDGMENTS

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the "The Future of Value Creation—Research on Production, Services and Work" program and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

## REFERENCES

Browning, T.R. (2001), "Applying the design structure matrix to system decomposition and integration problems: a review and new directions", *IEEE Transactions on Engineering Management*, Vol. 48 No. 3, pp. 292–306. http://doi.org/10.1109/17.946528.

Chen, L. and Whyte, J. (2021), "Understanding design change propagation in complex engineering systems using a digital twin and design structure matrix", *Engineering, Construction and Architectural Management, Vol. 29 No. 8, pp. 2950-2975*. https://doi.org/10.1108/ECAM-08-2020-0615.

Chen, Y., Cheng, P. and Yin, J. (2010), "Change propagation analysis of trustworthy requirements based on dependency relations", *IEEE International Conference on Information Management and Engineering*, Chengdu, China, 16-18 April 2010, IEEE, IEEE Xplore pp. 246-251 http://doi.org/10.1109/ICIME.2010.5477679.

Cheng, X., Guo, Z., M., X. and Yuan, T. (2021), "Identification of Influential Modules Considering Design Change Impacts Based on Parallel Breadth-First Search and Bat Algorithm", *Frontiers in bioengineering and biotechnology*, Vol. 9, p. 791566. http://doi.org/10.3389/fbioe.2021.791566.

Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting Change Propagation in Complex Design", *Journal of Mechanical Design*, Vol. 126 No. 5, pp. 788–797. http://doi.org/10.1115/1.1765117.

Danilovic, M.; Browning, T. R. (2007): "Managing complex product development projects with design structure matrices and domain mapping matrices" *International Journal of Project Management* 25 (3), pp. 300–314. https://doi.org/10.1016/j.ijproman.2006.11.003.

Drossel, W.-G., Ihlenfeldt, S., Langer, T. and Dumitrescu, R. (2018), "Cyber-Physische Systeme", In: Neugebauer, R. (Ed.), Digitalisierung, Springer Vieweg, Berlin, Heidelberg, pp. 197–222. http://doi.org/10.1007/978-3-662-55890-4_12.

Eckert, C.M., Keller, R. and Clarkson, P.J. (2011), "Change prediction in innovative products to avoid emergency innovation*", International Journal of Technology Management*, Vol. 55 No. 3/4, p. 226-237. http://doi.org/10.1504/IJTM.2011.041949.

Flanagan, T.L., Eckert, C.M., Eger, T., Smith, J. and Clarkson, P.J. (2003), "A FUNCTIONAL ANALYSIS OF CHANGE PROPAGATION", DS 31: Proceedings of ICED 03, *International Conference on Engineering Design*, Stockholm, 441-442 (exec.summ.), full paper no. DS31_1064FP.

Fu, Y., Li, M. and Chen, F. (2012), "Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix", *International Journal of Project Management*, Vol. 30 No. 3, pp. 363–373. http://doi.org/10.1016/j.ijproman.2011.08.004.

Hamraz, B., Caldwell, N.H.M. and Clarkson, P.J. (2012), "A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design", *Journal of Mechanical Design*, Vol. 134 No. 10, 100905. http://doi.org/10.1115/1.4007397.

Hamraz, B., Caldwell, N.H.M. and Clarkson, P.J. (2013), "A Matrix-Calculation-Based Algorithm for Numerical Change Propagation Analysis", *IEEE Transactions on Engineering Management*, Vol. 60 No. 1, pp. 186–198. http://doi.org/10.1109/TEM.2012.2203307.

Kim, S.Y., Moon, S.K., Oh, H.S., Park, T., Kyung, G. and Park, K.J. (2013), "Change propagation analysis for sustainability in product design", *IEEE International Conference on Industrial Engineering and Engineering Management*, Bangkok, Thailand, 10-13 December 2013, IEEE, IEEE Xplore, pp. 872-876. http://doi.org/10.1109/IEEM.2013.6962536.

Koh, E.C.Y., Förg, A., Kreimeyer, M. and Lienkamp, M. (2015), "Using engineering change forecast to prioritise component modularisation", *Research in Engineering Design*, Vol. 26 No. 4, pp. 337–353. http://doi.org/10.1007/s00163-015-0200-5.

Maier, J.F., Wynn, D.C., Biedermann, W., Lindemann, U. and Clarkson, P.J. (2014), "Simulating progressive iteration, rework and change propagation to prioritise design tasks", *Research in Engineering Design*, Vol. 25 No. 4, pp. 283–307. http://doi.org/10.1007/s00163-014-0174-8.

Maurer, M. and Lindemann, U. (2007), "Structural awareness in complex product design – the Multiple-Domain Matrix", *DSM 2007:* Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, Design Society, pp. 87–100.

McHugh, M.L. (2012), "Interrater reliability: the kappa statistic", *Biochemia medica*, Vol. 22 No. 3, pp. 276–282.



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press

Morkos, B., Shankar, P. and Summers, J.D. (2012), "Predicting requirement change propagation, using higher order design structure matrices: an industry case study", *Journal of Engineering Design*, Vol. 23 No. 12, pp. 905–926. http://doi.org/10.1080/09544828.2012.662273.

Nonsiri, S., Coatanea, E., Bakhouya, M. and Mokammel, F. (2013), "Model-based approach for change propagation analysis in requirements", *7th Annual IEEE Systems Conference*, 4/15/2013 - 4/18/2013, Orlando, FL, IEEE, pp. 497–503. http://doi.org/10.1109/SysCon.2013.6549928.

Oduncuoglu, A. and Thomson, V. (2011), "EVALUATING THE RISK OF CHANGE PROPAGATION", *DS 68-10*: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 10: Design Methods and Tools pt. 2, Copenhagen, Denmark, 15.-19.08.2011, pp. 114–125.

Rajinia, E. and Li, S. (2010), "Estimation of the Scope of Change Propagation in Object-Oriented Programming", *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Volume 3: 30th Computers and Information in Engineering Conference, Parts A and B. Montreal, Quebec, Canada. August 15–18, 2010. pp. 1203-1212 http://doi.org/10.1115/DETC2010-28773.

Saoud, L.A., Omran, J., Hassan, B., Vilutienė, T. and Kiaulakis, A. (2017), "A method to predict change propagation within building information model", *Journal of Civil Engineering and Management*, Vol. 23 No. 6, pp. 836–846. http://doi.org/10.3846/13923730.2017.1323006.

Scheinpflug, R. (2017), "Komplexität – Herausforderungen für Unternehmen und Management", In: Scheinpflug, R., Stolzenberg, K. (Ed.) Neue Komplexität in Personalarbeit und Führung, Springer Gabler, Wiesbaden, pp. 11–28. http://doi.org/10.1007/978-3-658-16135-4_2.

Schlick, C.M., Duckwitz, S., Hinrichsen, S. and Licht, T. (2007), "Komplexitätsbestimmung von Concurrent Engineering Projekten*", Industrie Management*, Vol. 23 No. 6, pp. 25-28.

Schneider, S., Schlick, C.M., Röwenstrunk, M. and Mütze-Niewöhner, S. (2012), "SIMULATION OF PRODUCT CHANGE EFFECTS BASED ON DESIGN STRUCTURE AND DOMAIN MAPPING MATRICES, *Proceedings of the 14th International DSM Conference*, Kyoto, Japan, 13.-14. September 2012, Hanser, München, pp. 423–435. http://doi.org/10.3139/9783446434127.033.

Steward, D.V. (1981), "The design structure system: A method for managing the design of complex systems", *IEEE Transactions on Engineering Management*, EM-28 No. 3, pp. 71–74. http://doi.org/10.1109/TEM.1981.6448589.

van Eck, N. J.; Waltman, L. (2010): Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics* 84 (2), pp. 523–538. https://dx.doi.org/10.1007/s11192-009-0146-3.

Zhao, Z.-Y., Lv, Q.-L. and You, W.-Y. (2008), "Applying Dependency Structure Matrix and Monte Carlo simulation to predict change in construction project", International Conference on Machine Learning and Cybernetics, Kunming, China, 12-15 July 2008, IEEE, IEEE Xplore, pp. 670-675 http://doi.org/10.1109/ICMLC.2008.4620489.



https://doi.org/10.1017/pds.2023.266 Published online by Cambridge University Press