

METHOD FOR ANALYSING REQUIREMENT CHANGE PROPAGATION BASED ON A MODIFIED PAGERANK ALGORITHM

Gräßler, Iris; Thiele, Henrik; Oleff, Christian; Scholle, Philipp; Schulze, Veronika

Paderborn University

ABSTRACT

Complexity of products and systems is increasing through digitalization, interdisciplinarity as well as high technology maturity and new business models. In consequence, new product development (NPD) projects need to manage and satisfy a large number of requirements from a broad range of stakeholders. Yet, NPD projects are often delayed due to requirement changes. In this paper, a new method for analyzing requirement change propagation is presented. The method is based on the assessment of requirement interrelations structured in a requirements structure matrix by a modified page-rank algorithm. By the method, a high number of strongly interrelated requirements can be analyzed in an efficient manner. Additionally, higher-level interrelations as well as the relative weights of requirements are also incorporated in the analysis. Hereby, an efficient holistic approach towards the analysis of requirement change propagation is proposed.

Keywords: Requirements, Algorithm, Clustering, Uncertainty, Complexity

Contact:

Thiele, Henrik
University of Paderborn
Heinz Nixdorf Institut
Germany
henrik.thiele@hni.upb.de

Cite this article: Gräßler, I., Thiele, H., Oleff, C., Scholle, P., Schulze, V. (2019) 'Method for Analysing Requirement Change Propagation based on a Modified Pagerank Algorithm', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.375

1 INTRODUCTION

Complexity of products and systems is increasing through digitalization, interdisciplinarity as well as high technology maturity and new business models. In consequence, new product development (NPD) projects need to manage and satisfy a large number of requirements from a broad range of stakeholders.

Requirements management therefore is a key success factor for NPD projects (Brauns, 2016). Volatile and incomplete requirements significantly affect project success and cause delays in nearly 24% of all (software-) projects (Pohl, 2010). Delays are often caused by requirement changes (Song *et al.*, 2017). One reason is that requirements are often treated as independent entities, although they are highly interdependent by physical fundamentals or working principles (Scholle *et al.*, 2015). These interdependencies can cause propagation of requirement changes. If propagating requirement changes are not considered, inconsistencies in the requirement set will occur (Pohl and Rupp, 2015).

Most notably such requirement changes are occurring at decisions points when requirements are reassessed (Albers *et al.*, 2018). Usually, errors are disclosed and accordingly proper actions are initiated to correct such deficits (Albers *et al.*, 2018). Besides those intended requirement changes, development projects also face unintended changes. For example, external circumstances can force the project team to adapt the set of requirements (e.g. new standards, changes in competition or technology-breakthrough) (Gräßler *et al.*, 2018a). To address this issue, requirement changes and change propagation within the requirements set (i.e. a sequence of changes to related requirements) should be part of decision-making processes. The increase of this problem correlates with the complexity of systems. Whilst consistency of the requirements set can be ensured by engineer's experience for few requirements, it is not feasible for highly complex, interdisciplinary systems with a large number of requirements.

To assess the propagation of requirement changes in complex systems, a large number of highly interrelated requirements has to be analysed in an efficient way. The research objective of this paper is to answer the following research questions:

- How can relationships within a large number of requirements be analysed efficiently?
- How can the calculation effort for the incorporation of higher-level interdependencies between requirements be reduced?

To answer these research questions, an initial literature study on matrix-based approaches towards change propagation in requirements engineering was conducted. Results are summarized in section 2. In section 3, a novel approach towards requirements structuring based on a modified PageRank algorithm called 'Requirement Rank' is presented. The approach is validated in a two-fold approach: At first, a qualitative comparison to existing approaches is made. This is followed by an empirical validation in a students' development project.

The approach presented yields the potential to improve NPD significantly by reducing iterations induced by requirement-changes throughout an efficient, broad assessment of potential requirement change propagation. In comparison to existing approaches, the proposed Requirement Rank is faster in computation, even though higher level influences can be incorporated into the holistic approach.

2 STATE OF THE ART

Requirements Engineering is a key task in product development (Eben, K. G. M. and Lindemann, 2010). Although most research is done in software engineering (Pohl and Rupp, 2015), several requirements engineering approaches were developed and transferred to product engineering (Jochem and Landgraf, 2011).

2.1 Requirements engineering

Major activities of Requirements Engineering are to elicit, document, check, align and manage requirements (Rupp, 2014; Pohl and Rupp, 2015). Requirements constitute the needs and information from all system stakeholders and thereby set the basis for verification (product is built right) and validation effort (right product is built) (SE Handbook Working Group, 2014). In case a system is considered as verified and validated, it fulfils all requirements and satisfies the stakeholder needs (Gräßler and Hentze, 2017).

To ensure that development activities target the actual needs, it is important to update the requirement set alongside a deeper understanding of the stakeholder needs, environmental restrictions as well as the system itself. Such understanding is increasing with system maturity and project progress. Requirement changes are mostly originating from dynamics of stakeholder interests, further restrictions or inaccuracy/fuzziness during requirement elicitation (Pahl *et al.*, 2007). The impact of requirement changes on project duration, as mentioned in section 1, can be traced back to the change propagation within the requirements network for most subsystems and components (Pohl and Rupp, 2015).

Recent research shows that the simplification to treat requirements as independent from one another is insufficient and emphasizes the importance of considering dependencies for product development projects (Brauns, 2016; Pohl, 2010; Song *et al.*, 2017). In literature several types of requirement dependencies are described (Pohl, 1996; Dahlstedt and Persson, 2005; Rupp, 2014; Gräßler and Hentze, 2017; Robinson *et al.*, 2003). A very simple differentiation is whether the dependency is unidirectional or bidirectional (Clarkson *et al.*, 2004). Various dependency-models can be found in literature. First, they vary according to the level of detail. Second some include dependencies between requirements and non-requirement artefacts (e.g. documents) (Pohl, 1996), others focus solely on requirement-requirement dependencies (Gräßler and Hentze, 2017). Taking a wide perspective can be very useful for traceability (Rupp, 2009), but leads to significantly more interconnections. In order to identify and assess requirement changes within the requirement set, it is sufficient to take a narrow perspective and just consider requirement-requirement dependencies.

The identification and assessment of requirement changes should be performed in a systematic way. Part of this is to understand and predict the change propagation. Such knowledge is valuable to:

- assess the impact of a requirement change on the requirement-network
- decide on change implementation and consequences in the requirement set
- assess the probability of a propagating requirement change, induced by changes of other requirements

Approaches towards impact analysis and change assessment not intended to address requirements directly were presented by Weber and Deubel (Weber and Deubel, 2003) or Krusche (Krusche, 2000). The dynamics of customer requirements and its management is addressed by other approaches (Gutpa *et al.*, 2018). All approaches mentioned yield the potential to support requirements engineering.

2.2 Matrix-based approaches towards requirements interdependency and change propagation

Based on the Design Structure Matrix (DSM) proposed by Eppinger and Browning (Eppinger and Browning, 2012; Browning, 2001), other approaches addressing change propagation between either components or their requirements have been presented by various authors (see (Clarkson *et al.*, 2004; Eben and Lindemann, 2010; Gräßler *et al.*, 2018b)).

Taking into account the DSM representing the relationship of subsystems or components in a square matrix, the impact and propagation of changes of system elements in a system is addressed by the Change Prediction Method (CPM) proposed by Clarkson *et al.* (Clarkson *et al.*, 2004). Probabilities are incorporated to assess the risk of change. The risk is calculated by multiplying a matrix representing the likelihood of change and a matrix representing the impact of changes. Higher-order dependencies can be analysed with CPM as well. The result is a change propagation tree combining both direct and indirect dependencies. Both impact and likelihood matrix are set-up manually. The effort of calculation is relatively high due to the dense nature of the matrices (Clarkson *et al.*, 2004).

An approach for assessing the interdependencies of requirements in a design structure matrix is presented by Eben and Lindemann (2010). Here, the requirements are considered as the nodes of a directed graph. Edges represent interdependencies. To filter critical requirements, graph-theoretical indicators such as active and passive sum as well as criticality are taken into consideration (Eben and Lindemann, 2010). Change propagation is assessed by analysing reachable nodes of a single requirement (Eben and Lindemann, 2010).

Another matrix-based approach towards change propagation was presented by Gräßler *et al.* (Gräßler *et al.*, 2018a; Gräßler *et al.*, 2018b). Here, requirement interrelations are assessed in the Requirements Structure Matrix, a derivative of the DSM. To reduce efforts compared to the approach of Eben and Lindemann, the RSM can be derived semi-automatically by applying a rule set (Gräßler *et al.*, 2018b). The identification of critical requirements based on interdependencies is combined with a risk indication per requirement. This risk assessment is based on the criteria ‘uncertainty’, ‘dynamics’ and ‘relevance’ of a requirement (or its source) (Gräßler *et al.*, 2018a). Yet, higher-level interdependencies are not considered by the approach.

The differences between the approaches presented are outlined in table 1 below.

Table 1: Comparison of matrix-based approaches

Criterion	Approach		
	Clarkson et al. (2004)	Eben and Lindemann (2010)	Gräßler et al.(2018a)
Integration of Change Propagation	Yes	Yes	Limited (just direct influences)
Direct implementation of change risk	Yes	No	Yes
Usability for a large number of requirements	Limited (calculation effort)	Limited (calculation effort)	Yes

Approaches considering higher-level interdependencies are less applicable to complex problems. They require too high calculation efforts for indirect interdependencies in the dense matrices depicting the interrelations. If only direct influences are considered, the integration of change propagation is limited since higher-level interdependencies are not considered.

Overall, there is no sufficient support in product development for requirement change analysis, which takes into account higher-level requirement interdependencies and change propagation. Furthermore, existing approaches are not scalable to extensive requirement sets, which will become more important in future development of complex and interdisciplinary systems.

3 METHOD FOR ANALYSING REQUIREMENT CHANGE PROPAGATION BASED ON A MODIFIED PAGERANK ALGORITHM

In this section, a novel approach towards requirement structuring based on the PageRank algorithm is presented. Originating from computer science, the PageRank algorithm was developed by Brin and Page as Google's approach towards analysing the structure of the World Wide Web for optimization of search engine results (Brin and Page, 1998). The PageRank of a webpage is an indicator for a random user using the interrelations (here: hyperlinks) between two webpages to navigate to another page. A user of the world wide web is modelled by equation (1). The chance of choosing a random page is $(1-d)$. In the context of requirements engineering, the PageRank represents the potential change propagation in a given requirement network.

For the application of the PageRank algorithm, a given matrix has to be analysed as a graph. In this paper, the requirements structure matrix (RSM) is taken as the input for the PageRank analysis. Every node represents a requirement and every edge between two nodes represents an interdependency between two requirements (Gräßler et al., 2018b). Based on the graph the sets E_i^{in} and E_i^{out} are defined as the set of inbound and outbound edges of a requirement r_i . The sets therefore represent the requirements by which r_i is influenced by and r_i influences, respectively. A specific interdependency between requirements r_i and r_j is therefore set as v_{ij} . For the total number of n requirements and a damping factor d , the Requirement Passive Rank $PR(r_i)$ for the requirement r_i is computed as:

$$PR(r_i) = \frac{1-d}{n} + d \sum_{r_j \in E_i^{in}} \left(\sum_{r_k \in E_j^{out}} \frac{PR(r_k)}{v_{kj}} \right) \quad (1)$$

Subsequently a Requirement Active Rank $AR(r_i)$ will be defined. The influence of a requirement onto others is characterized by $AR(r_i)$, the influencing of a requirement by $PR(r_i)$. This algorithm for $PR(r_i)$ differs in the second addend from the original formula of Brin and Page. In the original, the sum of all pages pointing to r_i is taken and the quotient of $PR(r_j)$ divided by the number of outgoing links of r_j is calculated. In order to integrate the strength of interrelation of the specific entries of the RSM into the calculation, the second sum here for the edges starting from r_j is added. The quotient is calculated from the Page Rank $PR(r_j)$ and the interdependency v_{kj} between r_k and r_j . This improvement to the original method allows the consideration of the strength of the observed

interrelations between requirements. This is not to be confused with the weighted Page Rank. (Xing and Ghorbani, 2004)

The method defined by equation (1) is an iterative method. Initial values have to be chosen. In this case, all nodes are assigned the same weight, i.e. $PR(r_1) = PR(r_2) = \dots = PR(r_n) = \frac{1}{n}$. In every step of the iteration, the weight of node r_i is distributed between all neighbouring nodes r_j with an edge between r_i and r_j . Value v_{ij} in the RSM indicates the amount assigned to an edge. For every $d < 1$, convergence within the values $PR(r_i)$ is achieved. This means that each additional step changes the sum of all $PR(r_i)$ less than a predefined value ϵ , close to 0. Moreover, for a $d < 1$ a unique solution is obtained. (Bryan and Leise, 2006)

Through the damping factor $d \in [0,1]$ a certain amount of the weight is distributed to all other nodes. It characterizes the likelihood a requirement is changed by another requirement, where no direct interrelation is denoted in the RSM. The factor d therefore represents the extend of assumed uncertainty in the elicitation of requirements interdependencies. For $d=0$ the Requirement Passive Rank is equal for all requirements and the interpretation is that all requirements influence one another randomly while no interdependency is known. For $d=1$ all possible interrelations are known and the likelihood of a changes to requirements, not represented by the RSM or not linked to other requirements, is set to 0. Since the method is not stable for $d=1$, the factor can be set as $d = 0.9999 \approx 1$ for a calculation under said pretences. As proposed by Brin and Page, in practice a value of $d = 0.85$ can be assumed, as it proved to be a robust choice for d . Intelligent strategies in choosing d though the values of the RSM can be applied. (Fu et al., 2006)

For a project with 30 requirements where a specific r_i has three inbound interrelations, the requirement r_i is influenced by any neighbouring requirement with a chance of $\frac{0.85}{3} \approx 0.2833$ and by any other requirement with a chance of $\frac{0.15}{27} \approx 0.0055$.

Following the original work and applying it to the given context, the value $PR(r_i)$ can be interpreted as the likelihood a requirement r_i is changed within the network defined by the RSM. The overall sum of all these likelihoods is $\sum_i r_i = 1$.

Since the overall sum of all Requirement Passive Ranks is constant through every step, and distributed through the whole examined system, the method yields a holistic view of all requirements. Therefore, the user is able to interpret each calculated value within the context of the whole system.

Nodes with no outbound edge, are called dangling nodes. If a RSM includes such a requirement, a modification to equation (1) is needed as to ensure convergence of the method. (Bryan and Leise, 2006)

An exemplary result can be examined in Figure 1, where the diameter represents the Requirements Passive Rank. Requirement r_2 , as the node with most inbound edges, is ranked highest. Requirements r_1 , r_5 and r_6 are all weighted similarly less than requirement r_2 , while having a different number of inbound edges. The effect traces back to the distribution of the weight of one's neighbour, as seen in the r_5 to r_6 : comparison r_5 has more inbound edges, but r_6 is neighbour to the relatively high weighted r_2 .

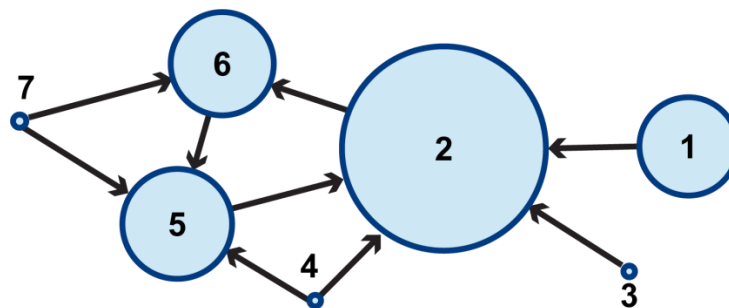


Figure 1: Exemplary Requirement Graph. The size of the nodes represents the Requirement Passive Rank $PR(r_i)$.

Besides the effect of changing requirements on a specific requirement, the effect of a specific requirement on other requirements has to be considered in prioritization of requirements. For said

effect, the Requirement Active Rank $AR(r_i)$ of a requirement is calculated by interchanging the sets E^{in} and E^{out} in the algorithm above. The active value $AR(r_i)$ therefore calculates the likelihood a requirement r_i induces a change impulse within the system. The Requirement Active Rank algorithm yields for the RSM, given above, the following values for the Requirement Active Rank $AR(r_{1..7}) = (0.121; 0.22; 0.121; 0.155; 0.121; 0.108; 0.154)$.

Table 2: Exemplary Requirement Structure Matrix (RSM)

	r_1	r_2	r_3	r_4	r_5	r_6	r_7
r_1	0	1	0	0	0	0	0
r_2	1	0	0	0	0	1	0
r_3	0	1	0	0	0	0	0
r_4	0	1	0	0	1	0	0
r_5	0	1	0	0	0	0	0
r_6	0	0	0	0	1	0	0
r_7	0	0	0	0	1	1	0
$PR(r_i)$	0,18	0,7	0,02	0,02	0,20	0,19	0,02

The values of $AR(r_i)$ can be plotted against those of $PR(r_i)$ to evaluate the joint value of a requirement in comparison to all other requirements. The values of $AR(r_i)$ and $PR(r_i)$ are scaled for the highest value of each to be 1. The average Rank is plotted to the portfolio accordingly.

The portfolio plot allows a joint analysis of both the $AR(r_i)$ and $PR(r_i)$. Following

1. a Systems Grid Logic, requirements can be sorted into groups by comparing $AR(r_i)$ and $PR(r_i)$ to the average value ϕAR and ϕPR . This yields four groups high/low instigation - high/low receivement (blue lines in Figure 2) (see (Gräßler *et al.*, 2017)),
2. the Isometric Logic of Clarkson, the requirements can be grouped by their distance to the isometric AR and PR values (dotted lines in Figure 2) (see (Clarkson *et al.*, 2004)).

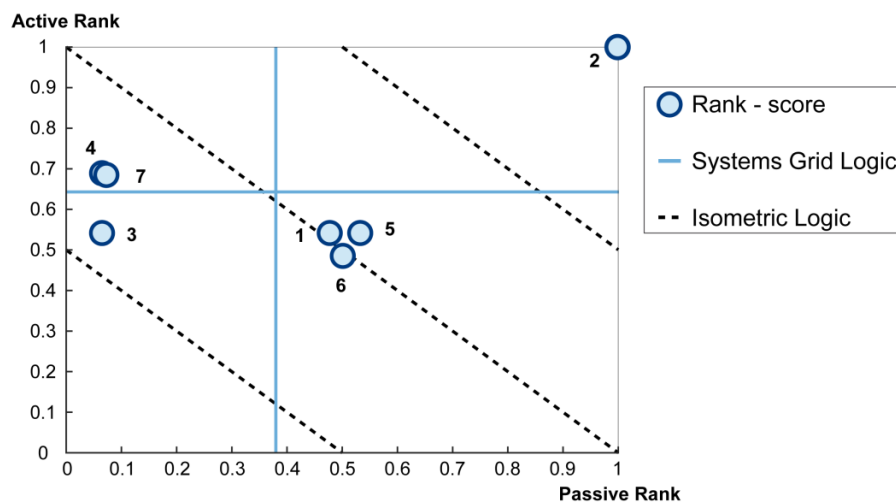


Figure 2: Exemplary Portfolio Plot

For each of the groups, suggestions for actions can be derived. In both cases, the top right group is assigned the highest risk and the bottom left the lowest risk. The calculated joint value $AR(r_i), PR(r_i)$ is called Requirement Rank.

The Requirement Rank allows a holistic consideration of the effect changing requirements have on other interrelated requirements. In contrast to other approaches, any resulting value can be compared, independent from a specific path length. The relative importance of requirements is aggregated in the Requirement Rank as well, as it is calculated on the basis of the weight of each node and the interrelated nodes. Hence, the selection of the requirements with the highest risk is absolute and not in relation to only the nearest neighbour, thus fulfilling the requirements to a holistic method of prioritizing requirements.

4 VALIDATION

To validate the results, a qualitative discussion of all methods presented is performed, the Requirement Rank is applied to a case study for a wheel carrier with 26 requirements (Gräßler *et al.*, 2018b) and for a qualitative analysis and to a set of 68 RSMs for a quantitative analysis.

4.1 Qualitative comparison

To assess the validity of the results, the proposed method is compared qualitatively to the CPM method from both, a theoretical and a case study perspective.

4.1.1 Qualitative comparison to the Change Prediction Method (CPM)

The aim of the DSM based methods shown in section 2 is to give a prediction of change propagation. To substantiate the application of the Requirement Rank algorithm, the presented approach is compared to Change Prediction Method (CPM). Both methods are compared in regards to the three characteristics mentioned in section 2.

- **Integration of Change Propagation**
The basic approach of probabilistic considerations is the same for the Requirement Rank as for the CPM. In the Requirement Rank method, the random surfer principle is used for interpretation. The principle can be summarized for the adjacency matrix and the transposed adjacency matrix as follows: The likelihood a requirement will affect the system and how much it will be affected. Through the random surfer principle, the change in one requirement propagates through the whole system of requirements with decreasing strength. The overall result of the CPM are the likelihood and risk of a requirement to change. As the values are calculated through consideration of all chains of change propagation up to a predefined length, change propagation is implemented in the CPM method locally.
- **Direct Implementation of Change Risk**
The values $PR(r_i)$ are the chance a requirement is changed by any other requirement with the sum over all chance being 1. Therefore, a direct risk of the chance a requirement to change is obtained. The value $AR(r_i)$ furthermore yields the chance, i.e. risk, of a requirement to change any requirement other of the system. As the CPM yields the likelihood and risk of a requirement to change. As for the method, proposed in section 3, the change risk is obtained directly.
- **Computation time**
Computation time of the PageRank is linear dependent on both the number of edges and vertices, as linear systems of equations have to be solved (Bryan and Leise, 2006). CPMs is exponentially dependent on the number of nodes due to the breadth-first search of the graph. For a depth length of bigger the six, computation times out (Clarkson *et al.*, 2004).

4.1.2 Case study - wheel carrier

Part of the Formula Student's race car of Paderborn University, the wheel carrier is an additively manufactured component. The initial requirement list contained 25 elements. A detailed description of the wheel carrier can be found in (Gräßler *et al.*, 2018b).

When applying CPM to the case study, the computation timed out. Results obtained by the Requirement Rank were computed efficiently and fast. Since higher-level interdependencies are automatically considered, the results are more differentiated compared to conventional approaches based on active and passive sum. With the conventional approaches, the three requirements r_{10} , (specification of interfaces), r_{17} (specified loads) and r_{18} (maximum displacement) as well as the two requirements r_{11} (position of brake caliper) and r_{25} (temperature resistance) are classified by an identical AR and PR for each group. By applying the requirement rank, differentiated results were obtained: Requirement r_{10} is characterized by a higher AR compared to r_{17} and r_{18} (whose PR is higher) and may therefore be prioritized higher due to its driving character in the network. When comparing r_{11} and r_{25} , both AR and PR are higher for the first requirement. Results obtained by Requirement Rank are more differentiated in comparison compared to conventional approaches since higher level influences were considered.

The advantages of the Requirement Rank, faster computation time and a differentiated analysis were validated by the case study of a wheel carrier.

4.2 Quantitative study

For a quantitative study of the results, a set of 68 Requirement Structure Matrices with up to 25 requirements each were examined. The RSMs were obtained by students, eliciting requirements for various products. The objective of this study is to compare results of the Requirement Rank with conventional, graph theoretical, matrix based approaches such as active and passive sum proposed by Eben and Lindemann (2010) and Gräßler et al. (2018a), introduced in section 2. At this point it is added that the Eben and Lindemann method can also be applied to the RSM instead of the DSM. Only a more specific matrix with regard to the requirements, which can be taken from the DSM.

Quantitatively, the comparison can be made over the groupings of the requirements with two different logics. The overlapping of the requirements in the groups shows comparable quality of the methods.

For each requirement r_i within the RSM, the values for the Requirement Active $AR(r_i)$ and the Passive Rank $PR(r_i)$ were calculated as well as the active sum $AS(r_i)$ and passive sum $PS(r_i)$. To clarify and repeat the terms, the Requirement Rank is the calculated joint value $AR(r_i), PR(r_i)$. Following the groupings of section 3, i.e. by the System Grid logic and the Isometric logic, all requirements were sorted into groups by their active and passive sum and their Requirement Rank respectively.

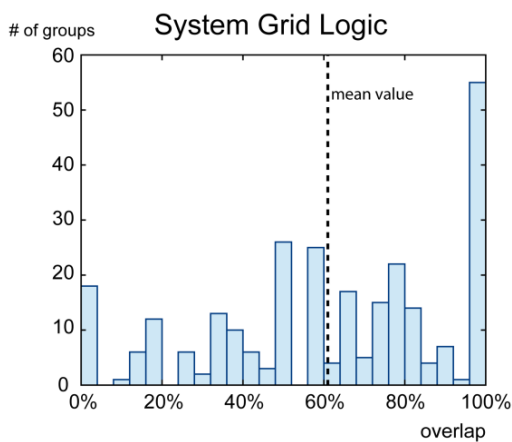


Figure 3: System Grid overlap

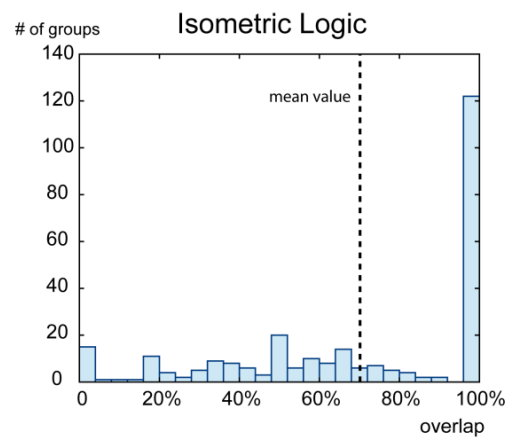


Figure 4: Isometric overlap

For each project, the overlap between the requirements in each group is calculated. As an example, if $[r_1, r_2, r_3]$ and $[r_1, r_2, r_4]$ are sorted into one group by the Requirement Rank and active/passive sum method respectively, an overlap of 66,6% is calculated. The overlap in percent over all groups and projects is displayed in Figure 2 and Figure 3. For a grouping by the System Grid logic, an average of 61% of the requirements are sorted into the same category and for the isometric logic, the average overlap is 70%. It can be observed that for both methods, the comparatively highest amount of groups contains the same requirements.

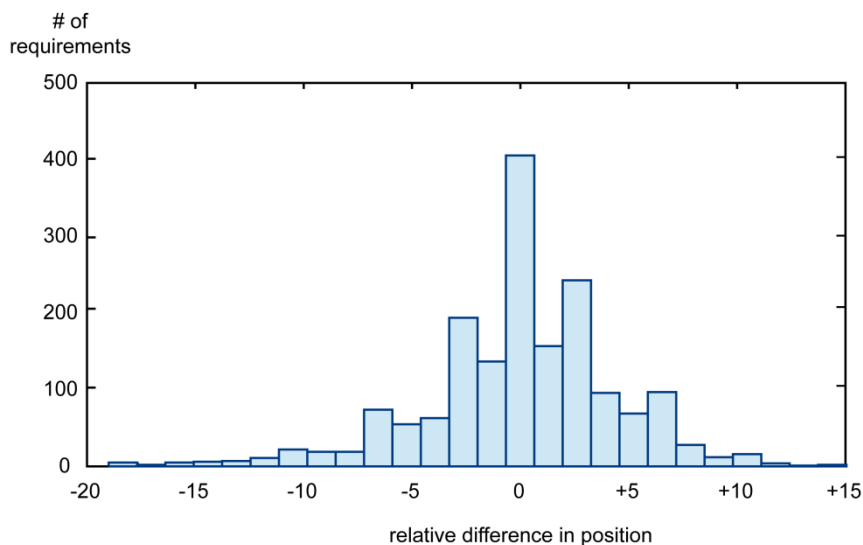


Figure 5: Relative difference in position in ordering by criticality between sum and Rank

The proposed approach is based on of an active and passive score for inbound and outbound relations of a requirement. Hence, the high amount of values, sorted into the same category is logical. For both the sum and score, the criticality (according to Eben and Lindemann, 2010) is calculated and a ranking is obtained. The difference in position within the ranking for each requirement over all RSMs is displayed in Figure 4. The mean difference value is 0 with a standard derivation of 4.41 and the mean difference of the absolute values is 3.06 with a maximum difference of 19. The results obtained by the Requirement Rank are similar to the results obtained with the approach of Eben and Lindemann (2010).

4.3 Summary of the validation

The proposed Requirement Rank method is shown to have qualitative advantages in comparison to existing approaches. With a wider range of applications, the method can be used in a wider set of practical contexts. Degrees of freedom are higher and computation time is lower than of those of the other methods. Additionally, the quality of results is higher as well, while being based on the same basic approach. Especially for the development of products with high complexity, applicability is proven. Thus, the method is usable for a type of development projects with a high need for requirement prioritization.

These results are manifested in the quantitative analysis as well, since a similarity to conventional approaches was proven, but not equal results over a set of different RSMs. Therefore, no contradiction between the proposed and the established and well-established methods is found, while improvements to the premise of the analysis were made.

5 CONCLUSION AND OUTLOOK

Due to requirement interdependencies, the change of one requirement can influence other requirements. An assessment of requirements regarding their likelihood of changes enables advanced decision making in NPD. A method for prioritizing requirements by calculating a Requirements Rank is presented in the paper. It was derived by modification of the Google PageRank algorithm originally proposed by Brin and Page (1998). Besides inbound and outbound dependencies, weights of all interrelated requirements are considered by the method. This yields a holistic view of the priority within a requirements network. A parameter to formalize uncertainty in requirements elicitation was implemented in form of a damping factor. The proposed method was validated qualitatively and quantitatively by a comparison to existing approaches in a case study with 68 requirements structure matrices (RSM). Compared to existing approaches, the method yielded a broader range of applicability, a more holistic approach and a faster computation time.

Since the proposed method is fast in computation, an efficient application to highly complex systems with large RSMs is possible. It is expected that the method performs well on a large number of requirements since the PageRank was developed for web graphs bigger than a million nodes. A validation for complex systems and large, dense RSMs will be part of future work. Additionally, further studies into the optimal choice of the damping factor are proposed. Methods of choice for d in the context of robustness exist (Fu *et al.*, 2006), but an analysis regarding use cases in requirements engineering demands further investigation. Lastly, none of the mentioned methods takes into account the type of interrelation between the requirements. Since the type of interrelation has a high implication onto the type of suggested actions, further studies into the integration of the type of dependency are planned.

REFERENCES

- Albers, A., Heimicke, J., Hirschter, T., Richter, T., Reiß, N., Maier, A. and Bursac, N. (2018), "Managing Systems of Objectives in the agile Development of Mechatronic Systems by ASD – Agile Systems Design", Linköping, Sweden, 14th - 17th August.
- Brauns, C. (7 October 2016), Requirements Engineering und Management in der wehrtechnischen Beschaffung. Empirische Bestandsaufnahme, Konzeptentwicklung, Evaluation, *Dissertation*, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Hamburg.
- Brin, S. and Page, L. (1998), "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, Vol. 30 No. 1-7, pp. 107–117.
- 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.

- Bryan, K. and Leise, T. (2006), "The \$25,000,000,000 Eigenvector. The Linear Algebra behind Google", *SIAM Review*, Vol. 48 No. 3, pp. 569–581.
- 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.
- Dahlstedt, Å.G. and Persson, A. (2005), "Requirements Interdependencies: State of the Art and Future Challenges", In: A. Aurum and C. Wohlin, (Ed.), *Engineering and Managing Software Requirements*, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, pp. 95–116.
- Eben, K.G.M. and Lindemann, U. (2010), *Structural Analysis of Requirements. Interpretation of structural criteria*, Cambridge.
- Eppinger, S.D. and Browning, T.R. (2012), *Design structure matrix methods and applications*, Engineering systems.
- Fu, H.-H., Lin, D.K.J. and Tsai, H.-T. (2006), "Damping factor in Google page ranking", *Applied Stochastic Models in Business and Industry*, Vol. 22 No. 5-6, pp. 431–444.
- Gräßler, I. and Hentze, J. (2017), "Structuring and Describing Requirements in a Flexible Mesh for Development of Smart Interdisciplinary Systems", In: A. Araujo and C.A. Mota Soares, (Ed.), *Smart Structures and Materials*, Springer International Publishing, Basel, pp. 1622–1631.
- Gräßler, I., Oleff, C. and Scholle, P. (2018a), "Methode zur Bewertung von Anforderungsänderungen additiv gefertigter Produkte", In: D. Krause, K. Paetzold and S. Wartzack, (Ed.), *Design for X: Beiträge zum 29. DfX-Symposium, Tutzing, 25./26. September 2018*, TuTech Innovation, Hamburg, pp. 333–344.
- Gräßler, I., Scholle, P., Hentze, J. and Oleff, C. (2018b), "Semi-Automated Assessment of Requirement Interrelations", *paper presented at DESIGN Conference*, Dubrovnik, 21.-24.05.2018, Available at: <https://doi.org/10.21278/idc.2018.0298>
- Gräßler, I., Scholle, P. and Pottebaum, J. (2017), "Integrated process and data model for applying scenario-technique in requirements engineering", In: Design Society, (Ed.), *ICED17: 21st International Conference on Engineering Design Vancouver, 21.-25.08.2017*, Design Society, pp. 261–270.
- Gupta, R.K., Belkadi, F., Buergy, C., Bitte, F., Da Cunha, C., Buergin, J., Lanza, G. and Bernard, A. (2018), "Gathering, evaluating and managing customer feedback during aircraft production", *Computers & Industrial Engineering*, Vol. 115, pp. 559–572. <http://dx.doi.org/10.1016/j.cie.2017.12.012>.
- Jochem, R. and Landgraf, K. (Eds.) (2011), *Anforderungsmanagement in der Produktentwicklung: Komplexität reduzieren, Prozesse optimieren, Qualität sichern*, 1, Auflage, Symposium Publishing GmbH, Düsseldorf.
- Krusche, T. (2000), Strukturierung von Anforderungen für eine effiziente und effektive Produktentwicklung, Zugl.: Braunschweig, Techn. Univ., Diss., 2000, Bericht / Institut für Konstruktionslehre, Maschinen- und Feinwerkelemente. Technische Universität Braunschweig, Vol. 60, 1. Aufl., Mainz, Aachen.
- Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung ; Methoden und Anwendung*, 7, Aufl., Springer, Berlin, Heidelberg.
- Pohl, K. (1996), *Process-centered requirements engineering*, *Advanced software development series*, Vol. 5, Research Studies Press, Taunton.
- Pohl, K. (2010), *Requirements engineering: Fundamentals, principles, and techniques*, Springer, New York.
- Pohl, K. and Rupp, C. (2015), *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering foundation level nach IREB-Standard*, 4, überarbeitete Auflage, dpunkt, Heidelberg.
- Robinson, W.N., Pawlowski, S.D. and Volkov, V. (2003), "Requirements interaction management", *ACM Computing Surveys*, Vol. 35 No. 2, pp. 132–190.
- Rupp, C. (2009), *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*, 5., aktualisierte und erw. Aufl., Hanser, München.
- Rupp, C. (2014), *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*, 6., aktualisierte und erweiterte Auflage, Hanser, München.
- SE Handbook Working Group (INCOSE) (2014), *Systems Engineering Handbook: A Guide for System Life Cycle Processes And Activities*, San Diego.
- Song, Y.-W., Chahin, A., Scholle, P., Bender, B., Gräßler, I. and Paetzold, K. (2017), "Optimierung des Produktentwicklungsprozesses mittels Risikoanalyse vernetzter Anforderungen", In: D. Krause, K. Paetzold and S. Wartzack, (Ed.), *Design for X: Beiträge zum 28. DfX-Symposium, Oktober 2017*, TuTech Innovation, Hamburg, pp. 339–351.
- Weber, C. and Deubel, T. (2003), *New theory-based concepts for PDM and PLM*, Universität des Saarlandes.
- Xing, W. and Ghorbani, A. (2004), "Weighted PageRank algorithm", *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004, Fredericton, NB, Canada, 21.05.2004 - 21.05.2004*, IEEE, pp. 305–314.