eFMEA — RAISING EFFICIENCY OF FMEA BY MATRIX-BASED FUNCTION AND FAILURE NETWORKS

Maik Maurer

Technische Universität München, Product Development, Boltzmannstr. 15, 85748 Garching, Germany.
Email: maik.maurer@pe.mw.tum.de

Heiner Kesper

TESEON GmbH, Parkring 4, 85748 Garching, Germany.
Email: heiner.kesper@teseon.com

Failure Mode and Effect Analyses (FMEAs) have been carried out and developed further for the last 60 years. The benefit of FMEAs is so significant that its application has become a standard in modern product development. It is even compulsory in some of the major industries. Underlying methods and available software tools have been steadily improved and adapted to a wide variety of applications. But this support still does not assure completeness nor does it provide adequate efficiency. The approach presented in this paper enhances the procedure and software support of conventional FMEAs. The "enhanced Failure Mode and Effects Analysis" (eFMEA) provides improved ergonomics and increased efficiency while guaranteeing the completeness of all required examinations at any time.

Keywords: FMEA, Multiple-Domain Matrix, product quality, product safety.

1. INITIAL SITUATION

Failure Mode and Effects Analysis (FMEA) is a well-established method in product development. In some parts of the industry, e.g. automotive suppliers, the application of FMEA is even compulsory. The method comprises the systematic linking of components, functions and failures in a comprehensible and target-oriented way. The hierarchical composition of the FMEA components is easy to apply. Software support is available and a quasi-standard is established.

2. PROBLEM

By application of the FMEA failure chains shall be derived, rated and optimization measures shall be implemented. Hereby, the amount of possible failure chains can be very high.

Figure 1 shows the steps of FMEA compilation following the Automotive Industry Action Group reference manual [1] as it is implemented in the software-based approach. Values extracted from an industrial project are shown at the specific steps. It can be seen that 139 functions are assigned to 11 components. This results in 2750 potential dependencies between functions. 251 failures have been assigned to the functions. This results in more than 11.000 possible dependencies between failures.

Figure 2 clarifies the basic modeling of elements in the FMEA method: The acquired components form a hierarchical structure. Functions get linked to the components and failures get linked to

Research into Design — Supporting Sustainable Product Development. *Edited by* Amaresh Chakrabarti

ISBN: 978-981-08-7721-7

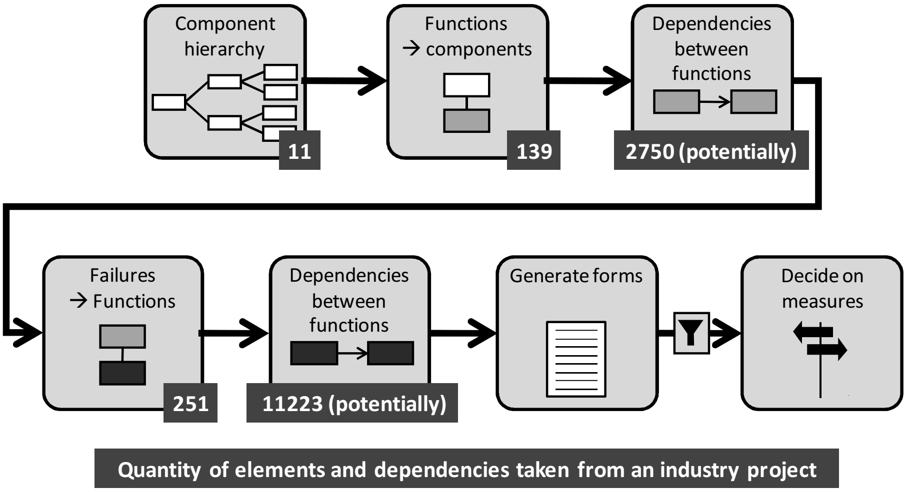

Figure 1. Software-supported process of FMEA compilation.

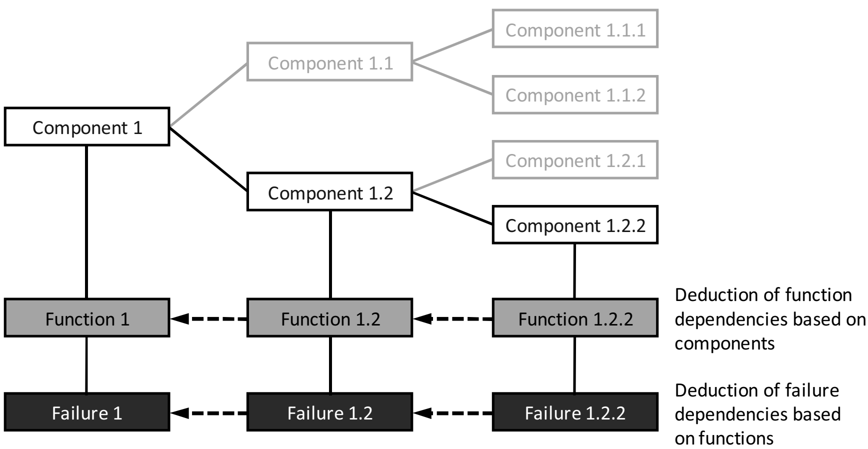

Figure 2. Computation of potential dependencies within functions and failures respectively.

functions. If users specify a dependency between two functions (i.e. one function is required for the other one), the failures attached to these functions obtain provisional links. From the list of so created provisional links users have to specify which failures can really cause further failures. Based on this theory, software solutions compute potential dependencies between functions and failures as follows: Two functions become linked potentially, if they depend on interlinked components (i.e. if one component is subordinate to the other one in the component hierarchy). Two failures become linked potentially, if they depend on interlinked functions (i.e. if one function is required for realizing the other one). In this dependency logic possible dependencies are only computed along the hierarchical component tree, which is created in the beginning of the FMEA compilation. It would be possible (and applicable by available software tools) to interlink functions or failures belonging to different tree branches of the component hierarchy. In this case the resulting amount of potential dependencies would be a lot higher.

Several problems arise from the amount of potential dependencies: If all possible dependencies between functions and between failures shall be considered, systematic processing is required. The actual procedure and available tools do not support the user in assigning probable dependencies as "considered". Thus, only existing dependencies can be retraced later on. Switching between different areas of the network is not easy, as the dependencies can only be accessed by the tree structure. As well, the actually applied representation only allows interruptions of the FMEA execution, if the tree structure is systematically gone through (e.g. top-down). However, workshop breaks are required often due to the large extend of typical FMEA systems.

The problem of insufficient efficiency in FMEA application results from the requirements on completeness of dependency consideration and insufficient ergonomics of available tool support. This could be counteracted by deriving potential dependencies between failures from existing dependencies between functions. Whereas the feasibility has been already introduced above, this step gets often partly or completely neglected in practice. This is because the creation of a comprehensive functions network is demanding — and in the end only the dependencies between failures are required. However, without the existence of a functions network, potential dependencies between failures cannot be derived automatically.

3. OBJECTIVE

The objective of the here presented approach is to enhance the FMEA process by a matrix-based method for dependency modeling. This shall improve the generation of functions and failure networks:

Firstly, the completeness in consideration of possible functions and failure networks shall be assured. Secondly, ergonomics of the acquisition of dependencies shall be improved. Users shall be able to reconstruct at any time the dependencies considered already. Thirdly, the efficiency of the FMEA conduction shall be increased. Therefore, potential failure dependencies must be derived from the functions network. In addition, possibilities of filtering shall be created for excluding failure dependencies of minor interest from a subsequent manual consideration.

The matrix-based method shall not replace the established procedure and tool application. In fact, it shall be integrated into the existing approach. For this reason, possibilities for the import and export have to be designed for the new matrix-based method.

4. SCIENTIFIC BACKGROUND

Matrices are commonly applied for the representation of system elements and their interdependencies. The Design Structure Matrix (DSM) is applied for the acquisition and analysis of system dependencies within elements belonging to one domain [2], e.g. functions. The DSM possesses a major advantage over a graph representation: all possible dependencies (represented by the matrix cells) between elements can be systematically gone through (e.g. row by row or column by column).

The Multiple-Domain Matrix (MDM) is a logical enhancement of the DSM and allows considering dependencies between elements of different domains [3], e.g. components, functions and failures.

Figure 3 shows the general layout of an MDM with three domains. As it can be seen, dependencies have been implemented between components, from components to functions and from functions to failures. This corresponds to the solid lines in Figure 2.

When applying a MDM the major functionality is the deduction of (indirect) dependencies [3]. For the system modeled in Figure 3 it is possible to derive dependencies between functions based on the dependencies between components and functions. That means that two functions get connected, if they are realized by two interlinked components. The deduction of indirect dependencies is based on matrix multiplication. Details about different computational use cases can be taken from [3]. This deduction of indirect dependencies can be supported by standard spreadsheet software or specialized tools, e.g. LOOMEO [4]. This software tool allows acquiring, exploring and analyzing any system structures. Elements and dependencies are modeled in MDMs and force-directed graphs. Thus, LOOMEO supports users in better understanding and managing complex systems. For example, the software gets applied to problems in variant management, change management or organizational planning.

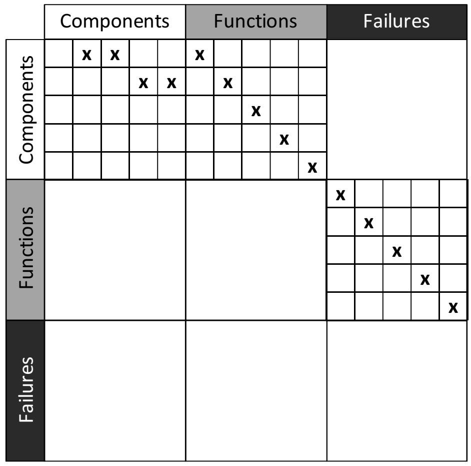

Figure 3. Multiple-Domain Matrix (MDM) comprising three domains [according to 3].

5. APPROACH TO AN ENHANCED FMEA PROCESS

Figure 4 shows an overview of the new process of the enhanced FMEA (named eFMEA), which will be introduced in the following.

5.1. Acquisition of system elements

The initial creation of the hierarchical component structure is executed by using the software APIS [5]. This tool represents a quasi-standard for the accomplishment of FMEAs in automotive industry for OEMs as well as for suppliers. Furthermore, APIS gets applied by many enterprises in the field of mechanical and electrical engineering. Besides creating the component structure also the assignment of functions to components and failures to functions is done by using this software. The system

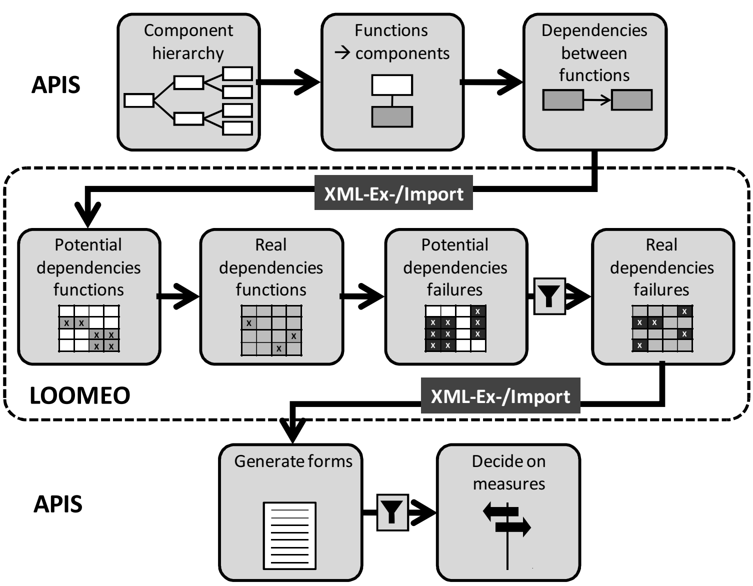

Figure 4. Enhanced FMEA process.

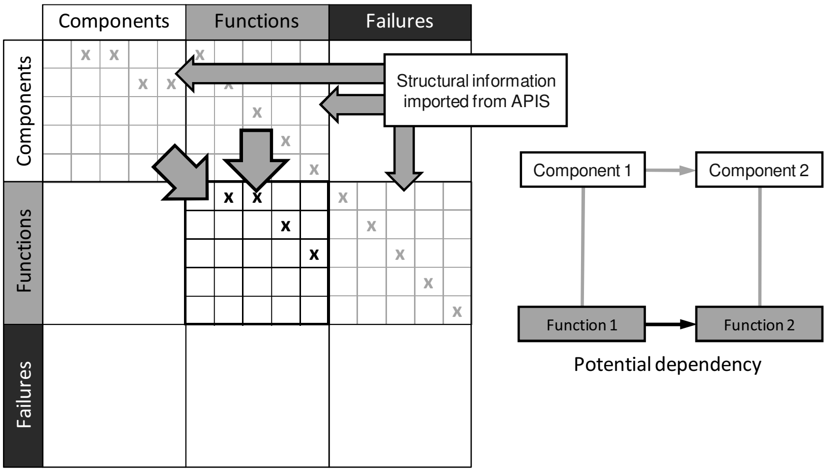

Figure 5. Deduction of potential dependencies between functions.

representation in form of a tree is intuitive for users and therefore particularly suitable for the collection and assignment of elements.

5.2. Creation of the function network

Once the system elements are acquired and denoted in the component-based tree structure, function and failure dependencies have to be acquired. Therefore, the acquired tree structure is transferred from APIS to LOOMEO by use of the XML format.

LOOMEO software depicts the functions on the axes of a square matrix (Figure 5). This software allows deriving potential dependencies between functions automatically (by application of matrix multiplications) from existing dependencies between the components (assigned to these functions). The components are linked to functions because these components realize the functions in the product (see the right side in Figure 5).

These automatically derived dependencies only represent the potential dependencies between functions. As not all of these links exist in practice, they have to be confirmed by experts. They have to go through the potential links subsequently in order to delete those ones that do not exist in reality. Although this manual rework cannot be avoided, the advantage of the new procedure becomes obvious: as only potential dependencies have to be gone through manually (and not all combinations of all functions), the required effort for building up the function network can be reduced significantly. At the same time, completeness is guaranteed, because all potential dependencies are derived automatically and thus cannot be overlooked.

5.3. Creating the failure network

Based on the function network the network of potential dependencies between failures can also be derived by matrix multiplication in the MDM (see Figure 6).

The logic of deriving potential dependencies is identical with the derivation of potential functions before (see Figure 5). As well, the derived potential failure dependencies have to be gone through manually to identify the really existing ones.

5.4. Rating potential failure dependencies

The expenditure of time to go through the potential dependencies between failures can be decreased significantly by filtering the relevant ones in advance. Therefore, all failures have to be rated. Available

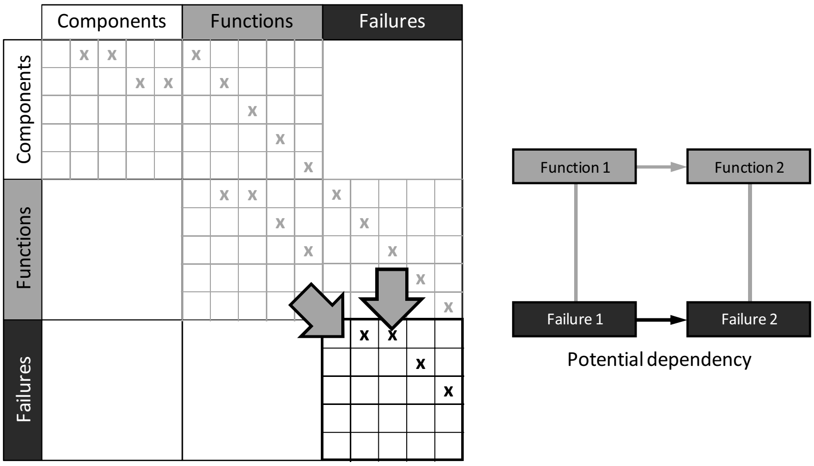

Figure 6. Deduction of potential dependencies between failures.

rating guidelines are neither universal nor standardized and many companies have their own guidelines to reflect their specific needs [6]. The standard FMEA process includes rating the occurrence (O) and detection (D) of failure causes as well as the severity (S) of failure effects. All three ratings can take values from 1 to 10, whereas 1 represents the best case and 10 the worst case. These standard ratings can be used to sort out specific potential dependencies automatically, e.g. those ones that may only lead to failure effects with minor severity. Alternatively, any combination of failure ratings can be used for sorting out potential dependencies.

The most common combination of ratings is the risk priority number (RPN). The RPN is intended for describing the criticality of failure chains consisting of a failure cause, a failure mode and a failure effect. It is represented by the product of the corresponding occurrence, detection and severity ($O \times D \times S$). Then, e.g. all potential dependencies can be ignored, which are only included in failure chains with low RPNs. Only the existence of the remaining potential dependencies needs to be checked after applying the filter (see Figure 7).

Filtering failures, i.e. excluding failures (and failure chains respectively) from further examinations depending on their ratings is also intended in the later stages of the standard FMEA process (represented by the funnel icon between the last two activities in Figure 4). Usually, preventive actions and detection actions are only specified for failure causes that possess a high RPN, i.e. a high occurrence, a low detection and may lead to failure effects with high severity.

It deserves mentioning that the significance of RPNs is controversial, because failure chains possessing different behaviors receive identical ratings. Assuming a detection of 5, an occurrence of

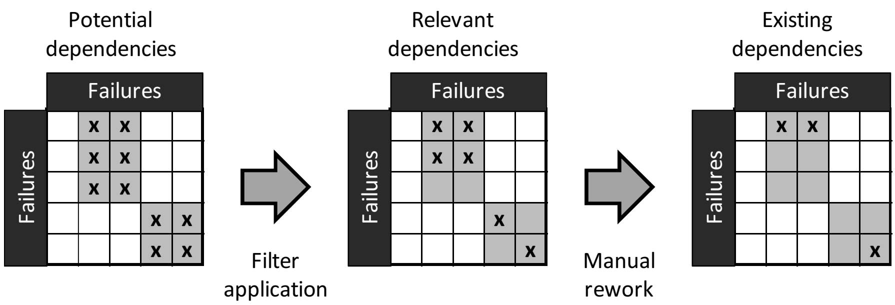

Figure 7. Filtering potential dependencies between failures depending on their relevance.

10 and a severity of 1 would result in a RPN of 50 — just as a detection of 5, an occurrence of 1 and a severity of 10. For that reason alternative concepts for rating the criticality of failure chains have been developed in the recent past [7, 8]. No matter which rating or combination of ratings is preferred: the presented approach allows disregarding specific failures or failure chains in order to increase efficiency depending on any suitable rating.

5.5. Re-transfer of network data to APIS

As mentioned before, the eFMEA process is not intended to replace the standard process but to extend it. Thus, once the dependencies are acquired in LOOMEO (by computing all potential dependencies and going through them) the network data is re-transferred to APIS.

Now, preventive actions and/or detection actions are defined for all failure causes that are included in critical failure chains. A filter algorithm (represented by the funnel icon in Figure 4) allows extracting the most relevant chains. After actions of improvement have been implemented to the system, the failure causes are rated again and the criticality of failure chains is re-computed. This proceeding is repeated until the criticality of all failure chains (usually represented by the RPN) falls below a specified value. The FMEA is completed by the generation of standardized forms that contain all failure chains, ratings and actions. APIS is particularly suitable to carry out these final steps, because it supports planning actions and provides forms meeting all the different international standards.

6. CASE STUDY

The eFMEA approach has been applied to a system in the automotive supplier industry. A conventional FMEA had been created in advance; this allowed assessing effort reductions due to the new approach. The system comprised 11 components, 139 functions and 251 failures. The component hierarchy led to 2750 potential dependencies between the 139 functions. After these dependencies were gone through, 149 remained. The 149 existing dependencies between functions resulted in 1023 potential dependencies between the 251 failures (347 dependencies actually remained). Relevance filter application could reduce these 1023 dependencies (which normally would have to be checked manually) to a minimum of 500 (depending on the underlying rating and combination of ratings respectively and the specified limiting value). Taking into account the 2750 potential dependencies between functions that need to be gone through in any case, the overall efficiency could be increased by 14%.

At the same time, the completeness of all investigations could be assured and improved ergonomics enabled a more flexible and comprehensive proceeding (resulting in additional efficiency increases that have not been quantified so far).

7. CONCLUSION AND OUTLOOK

The enhanced FMEA approach could proof its applicability in industrial practice. The objectives formulated in advance could be achieved:

Completeness is guaranteed at any time, as all potential dependencies between failures are computed automatically. This aspect is of major importance to the application of FMEAs to safety critical systems.

Ergonomics of the information acquisition process is improved by the clear matrix-based depiction of dependencies between components, functions and failures. This kind of system representation allows gaining an overview of the entire structure at any time and thus estimating the efforts required to complete the analysis. Additionally, already considered potential dependencies are marked — no matter if they exist or not. So, potential dependencies can be gone through in any sequence and breaks become possible.

Finally, the efficiency of the FMEA generation is increased. The automatic deduction of potential dependencies between functions and failures prevents wasting time on the consideration of dependencies that cannot exist at all. The application of relevance filters at an early process stage allows excluding dependencies of minor interest, which results in a significant effort reduction and hence cost reduction.

Since the presented approach is integrated into the established process and tool application, it can be introduced easily. All data created during the eFMEA process remains accessible to APIS users, because LOOMEO provides the required import and export functionality.

Future work will have to comprise the development of new filters. Providing a set of application specific filters will be essential for deriving maximum benefit; filtering potential dependencies is the key to an efficient FMEA execution and the appreciation of filter criteria differs from enterprise to enterprise.

REFERENCES

1. AIAG, "Potential Failure Mode & Effects Analysis (FMEA)", 4th Ed., Southfield, MI, Automotive Industry Action Group, 2008.
2. Browning, T., "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions" IEEE Transactions on Engineering Management 48 (2001) 3, pp. 292–306.
3. Lindemann, U., Maurer, M. and Braun, T., "Structural Complexity Management — An Approach for the Field of Product Design", Berlin, Springer, 2009.
4. TESEON GmbH, URL: http://www.teseon.com, Taken: June 26th 2010.
5. APIS Informationstechnologien GmbH, URL: http://www.apis.de/en, Taken: June 26th 2010.
6. Stamatis, D. H., "Failure Mode and Effect Analysis: FMEA from Theory to Execution", 2nd Ed., Milwaukee, WI, American Society for Quality, 2003.
7. ISO/IEC 31010:209-11, "Risk management — Risk assessment techniques", Berlin, Beuth, 2009.
8. VDA Band 4, "Sicherung der Qualität in der Prozesslandschaft", 2nd Ed., Frankfurt, Verband der Automobilindustrie, 2009.