Software FMEA
Opportunities and benefits of FMEA in the development process of software-intensive technical systems

Oliver Mäckel
Siemens AG
Simulation and Risk Management
CT PP 2
81730 München

Technical systems are prevalent in many areas of our society. Nowadays they often include a considerable amount of software. Identification and avoidance of technical risks is of major importance in the development of these software-intensive technical systems. A powerful analysis technique in the development process for technical systems is the *Failure Mode and Effects Analysis* (FMEA). This technique has proved very effective in avoiding failures in many areas of industry. However, there is to date no widespread use of the FMEA technique for software-intensive systems. Objectives and benefits of carrying out FMEAs on software will be discussed along with advantages, areas of application, weaknesses and constraints.

Introduction

Technical systems are prevalent today in many areas of our society. Due to economic rationalization and the necessity to meet increased requirements regarding performance and ergonomics an ever-growing number of complex tasks are being automated. An increasing dependence of society on the safe and reliable operation of these systems is the consequence. As an example, a faulty ticket vending machine is certainly a nuisance for the user and may also lead to substantive damage. The unintentional inflation of an airbag without any underlying vehicle collision on the other hand could lead to serious injury or even fatalities. The catastrophic failure of an on-board aeroplane computer could lead to great loss of life.

Today technical systems often contain considerable amounts of software, which already constitutes an essential part of the system. It is a fact that new motor vehicles these days contain nearly 50 computer systems [1]. Extremely high safety and reliability levels are required of these mainly software-intensive systems. Examples can be found by considering costly capital equipment, especially aeroplanes and rail vehicles. High safety and reliability levels are also required for mass-produced products such as motor vehicle components, for industrial automation equipment etc [2]. These requirements necessitate, especially under the consideration of increased *time-to-market* and *cost-to-market* pressure, a risk-oriented development for software-intensive technical systems.

Failure Mode and Effect Analysis

The Failure Mode and Effects Analysis (FMEA) [3, 4] is an important analysis technique in the development process of technical systems. It was developed by NASA in the USA [7] in the early sixties for the Apollo Project. In the automobile industry it is standard procedure for planning and development [8]. In other areas of industry [9] FMEA can be found as a methodological component of quality management. The FMEA is acknowledged to the industry in many ranges [8, 9, 10].In a preventing way the FMEA takes failure behaviour and causes into consideration and evaluates associated risks with respect to occurrence, severity and detection. The simplicity and efficiency of the technique has proved its value and, furthermore it is recommended in relevant Standards [5, 6] for the development of safety-critical systems.

FMEA for software (SW-FMEA) - Goals and Benefits

In relation to hardware failure behaviour and human error it is gradually becoming more important to view the failure behaviour of software and its effects. This must be taken into account by the development of technical systems. FMEA is an established technique to avoid failures in technical systems. A timely performed FMEA is risk management instead of crisis management [15]. In the early phases of software development where the costs for changes are small (Fig. 1) and willingness to change is high, it makes sense to identify and avoid failures in a preventive way. By evaluating the individual risks a differentiation between high risk and low risk components, modules and functions can be achieved. This makes a risk-oriented development of software-intensive systems possible.

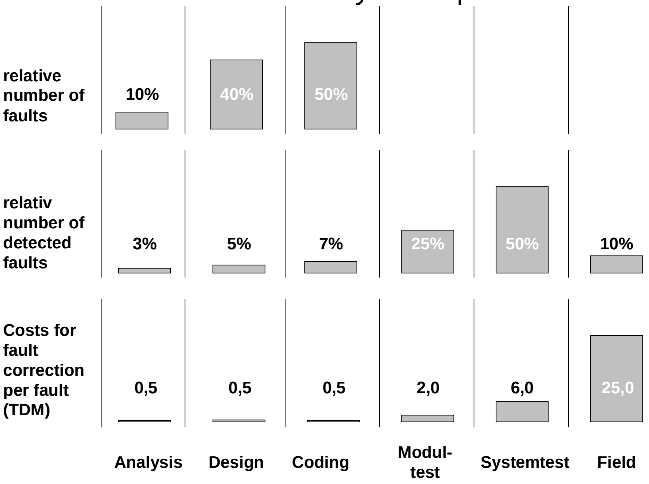


Fig. 1: Fault occurrence, fault elimination and fault correction costs in software development [15]

A SW-FMEA is the consitent continuation of the FMEA of the system (system FMEA: SFMEA) for analyzing software-intensive components of the considered system. Their results find their way back to the FMEA of the system. However, the FMEA technique is not yet widely used for software-intensive systems. General use of these analyses in the development of technical systems is more important the more the requirements for *time-to-market* and *cost-to-market* increase.

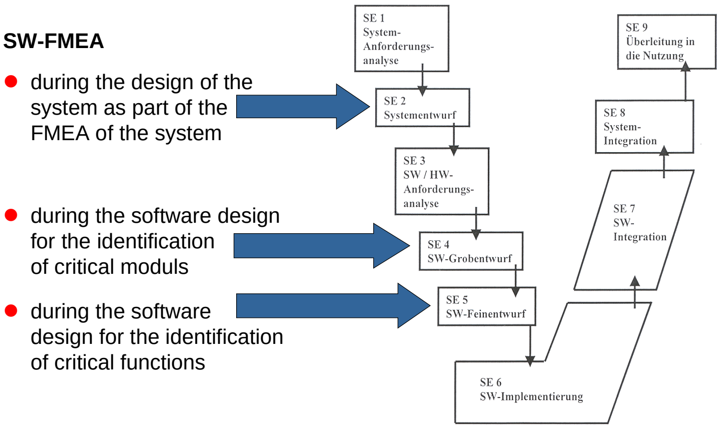

SW-FMEA
● during the design of the system as part of the FMEA of the system
● during the software design for the identification of critical moduls
● during the software design for the identification of critical functions
SE 1 System-Anforderungs-analyse
SE 2 Systementwurf
SE 3 SW / HW-Anforderungs-analyse
SE 4 SW-Grobentwurf
SE 5 SW-Feinentwurf
SE 6 SW-Implementierung
SE 7 SW-Integration
SE 8 System-Integration
SE 9 Überleitung in die Nutzung

Fig. 2: When should a SW-FMEA be performed?

The SW-FMEA is a systematic, structured technique for the review of the software architecture or the software design with respect to technical risks (e.g. safety, reliability or availability). The SW-FMEA is used for knowledge transfer. The knowledge of different departments, like for example system development, software development, test and service, is brought together and used during the FMEA in the team. So the number of views on or into a system and a system's software increases itself.

Procedure

The SW-FMEA is carried out as a supplement to a FMEA of a system. It is used for architecture or design review during the development. The SW-FMEA should be performed before the implementation of the software. It may not be executed on software source code (Fig. 2).

The SW-FMEA should also be executed in a team. This team has got members of different functional areas, like system development, software development, test and service.

The SW-FMEA is carried out in following steps:

1. The software to be examined is divided in components, modules and functions. A tree-similar structure develops.
2. For every component defined in the system structure the function has to be described. The function of a subcomponent represents a partial function of the superordinate component.
3. Corresponding possible failures and faults are assigned to every function of a component. The failure effects can be found then in the superordinate components. The failure causes are as a possible failure or fault listed in the subordinate components.
4. If a risk evaluation is supposed to be carried out,
   - the severity of the failure effects (in German: Bedeutung des Fehlverhaltens: B-value),
   - the probability of occurence (in German: Auftretenswahrscheinlichkeit: A-value) and
   - the detection probability of the failure causes (in German: Entdeckungswahrscheinlichkeit: E-value) will be listed.
5. Then the definition of measures for the improvement of the software through avoidance of possible faults or errors or the optimized detection of failures follows. This can happen for example through improved processes of development or through planning of special test cases. The evaluation of components, modules and functions with more or less risks follows on the basis of the quantitative risk evaluation.

Due to the manifold connection possibilities of components, modules and functions a SW-FMEA should be carried out by the support of a FMEA tool [12]. The management of the SW-FMEA will be much simpler through that and the realization will get more efficient.

Difficulties

In the practice some weak spots during the realization of the SW-FMEA exist. In total the risk evaluation turns out in a more difficult way than in a conventional SFMEA. The experience shows that on the average larger risk priority numbers (RPN) are obtained. From a direct comparison of the risk priority numbers with conventional SFMEAs and/or between SW-FMEAs must be warned.

Fundamentally two aspects which were criticized repeatedly within the framework of the conventional SFMEA [13] should be considered in particular at the evaluation of SW-FMEAs:

- The derivation of thumb rules for the initiation of measures must even be project-specific or singlerisk oriented. Global use of thumb rules for the initiation of measures, as "for all risks with a RPN > 100 a measure has to be defined" are senseless [13] and proved to be useless in particular at SW-FMEAs.
- The same risk will be evaluated from different teams and/or different expert often differently. A comparison over several FMEAs must fail from that.

Due to that a new procedure for the value formation at SW-FMEAs for the probability of occurrence (A-value) and the detection probability (E-value) is defined (based on a procedure discribed in [10]). The aim is an objective and usable risk evaluation.

The evaluation of the severity (B-value) shall be done in analog mode to the SFMEA in order to receive continuously consistent evaluations.

The occurrence and the detection takes off with software-intensive systems significantly from the complexity of the individual modules. At conventional SFMEAs the disturbance rate or probability of components out of the field are used within the risk evaluation for the determination of the A-value. For software this relation and transformation for the specific context must be determined first (Fig. 3). Test, verfication and maintenance strategies for the determination of the E-value are used in the conventional SFMEA. Test and review efficiency can be used useful to determine the E-value for software in combination with the respective module size or complexity (Fig. 4).

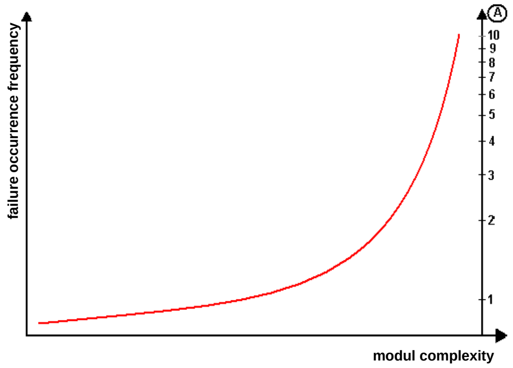


Fig. 3: Thumbsketch of a relation (probability of occurence / module complexity) including transformation to A-value

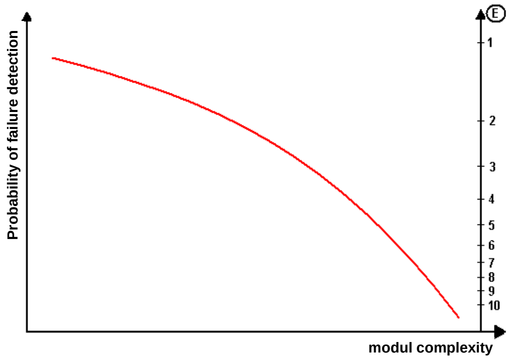


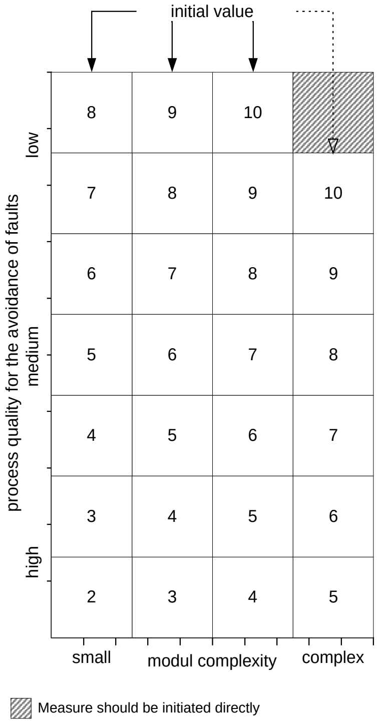
Fig. 4: Thumbsketch of a relation (probability of detection / module complexity) including transformation to E-value

Actually practical values may be determined at a SW-FMEA as follows. The evaluation of the occurrence and detection probability will be done in two steps. First an initial value has to be defined for the occurrence probability depending on the module complexity (Fig. 5).

This will be reduced then depending on the process quality of the carried out avoidance measures. Individual avoidance measures are for example:

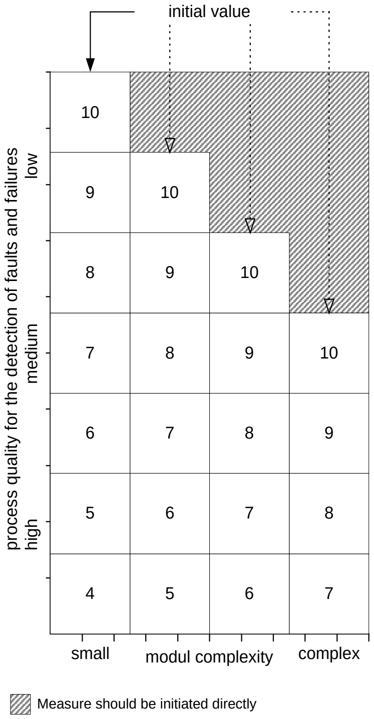
- Structured analysis
- Object-oriented analysis and design
- Formal design methods
- Design and coding standards
- Standardized programming language
- Validated compiler or even compiler which are well-proven in use

The in each case used measure combination causes a more or less effective and efficient process. The reduction of the initial value depends on this.


initial value
process quality for the avoidance of faults
low
medium
high
small
modul complexity
complex
Measure should be initiated directly

8	9	10	▨
7	8	9	10
6	7	8	9
5	6	7	8
4	5	6	7
3	4	5	6
2	3	4	5

Fig. 5: Determination of the value for probability of occurance


initial value
process quality for the detection of faults and failures
low
medium
high
small
modul complexity
complex
Measure should be initiated directly

10	▨	▨	▨
9	10	▨	▨
8	9	10	▨
7	8	9	10
6	7	8	9
5	6	7	8
4	5	6	7

Fig. 6: Determination of the value for detection probability

The same procedure is used for detection probability value. First an initial value of 10 points is assigned independently from the module complexity (Fig. 6).

Then this will be reduced depending on the process quality of the carried out detection measures. Detection measures for fault detection before system delivery are for example:

- Formal verification
- Reviews
- Functional test (black box-tests)
- Equivalence class tests
- Static analysis
- Data and control flow analysis
- Structure-oriented tests with statement, branch or path coverage
- Interface tests
- Stress tests

Detection measures for detection of failures during the systems runtime are for example:

- Defensive programming
- Failure Assertion Programming

The influence for the enlargement of the fault detection is evaluated according to the used measure combination. The respective measure combination leads analog to the avoidance measures to a more or less effective process quality, by which the decrease of the initial value is defined.

Definitively the risk priority number (RPN) is formed from the failure severity (B-value), the occurrence probability (A-value) and the detection probability (E-value).

In this case the evaluation of the process quality turned out in a more difficult way than expected. The possible combinations of the measures are diverse and the corresponding benefit of a measure combination is only very heavily appraisable on an ordinal scale.

Furthermore almost the same A and E-values were turned out for small software-intensive systems through almost identical measures or measure combinations for all failure causes.

Strengths

The SW-FMEA is simple and systematic. In an efficient way the SW-FMEA allows the structured analysis of a software architecture or a software design. With the aid of the SW-FMEA critical functions or modules and their risks will be identified systematically. This enables early a risk based development for example

- through the organization of measures to avoid software faults,
- trough the initiation of measures for the detection of faults bevor the delivery or
- the initiation of measures for the detection of failures during the runtime and
- through the derivation of propositions for the optimization of the software structure.

Risk based and disturbance based test cases and the appropriateness of tests and tests evaluates as soon as critical development instruments are worked out or identified.

In addition to that maintenance rules in order to guarantee the safe and reliable operation of the software-intensive system within the specified environmental conditions are worked out permanently.

Conclusion

The optimization of the software architecture and the software design and the derivation of test cases lead directly to an improvement of the software-intensive system. Since particularly these qualitative results stand in the foreground, the difficulties during the objective formation of values are negligible. Strengths and benefits of a SW-FMEA outweigh the difficulties from that.

The SW-FMEA is well suitable as a systematic risk based review method. It forces the developer to a structured way of thinking. The software developer, who thinks during the development functionally, is forced, to think in an failure-oriented way. He has to build up the entire figure of the failure event from the effects down to the causes.

In the analysis or design phase the respective system may be analyzed with respect to specific risk features by the use of a SW-FMEA. This is based on a systematic and structured dividing of the system.

For safety-related software the derivation of safety-oriented, mostly failure-oriented test cases is just interesting. These are normally usable within the framework of a validation of the software or the system and increase next to the quality also the confidence into the developed software-intensive and safety-related systems.

Perspective

The base forms a careful risk analysis and risk evaluation for the development of technical systems. The conversion of these analyses during the development of technical systems just wins more and more importance regarding to the rising *time-to-market* and *cost-to-market* requirements.

Moreover, missing or unclear risks can lead to gaps in the further development process. This may lead to risks or hazards. The SW-FMEA helps by the capturing of the missing or unclear risk requirements related to software components.

Due to the importance of the FMEA for general quality processes [8, 9, 10] and due to the demands from standards [5, 6,] the SW-FMEA will find in particular further circulation as a method for preventive failure avoidance. The systematic and structured procedure supports an architecture and design review just with regard to risk based questions.

Next to the up to now described possible applications of the SW-FMEA in the analysis and the design phase the SW-FMEA can also be used effectively in the requirement analysis in the sense of a systematic risk based review of the requirements specification in order to increase the quality of the requirements specification [11]. Following advantages turn out in this case:

- Early understanding of the requirements
- Improvement of the communication between the author of the requirements specification and the software design team
- Early recognition of mistakes in the requirements

For big software-intensive systems the SW-FMEA will be recommended as a good method for the review of the requirements specification [11] in the same way.

During the development of safety-critical systems in automotive industry, in aviation technology and in industrial automation the SW-FMEA will establish itself just the same as for availability-critical systems in the telecommunication.

References

[1] Jüttner, P., Schweikl, U., Siemens AT, *Software Development in Automotive Business*, Gastvortrag Uni Oldenburg, Mai 2000

[2] Liggesmeyer, P., *Qualitätsicherung softwareintensiver technischer Systeme*, Spektrum-Verlag Heidelberg, 2000

[3] DIN 25448, *Ausfalleffektanalyse (Fehler-Möglichkeits- und -Einflussanalyse)*, Mai 1990

[4] IEC 812, *Failure Mode and Effects Analysis*

[5] IEC 61508, *Functional safety of electrical / electronic / programmable electronic safety-related systems*

[6] prEN 50128, *Bahnanwendungen – Software für Eisenbahnsteuerungs- und Überwachungssysteme*, Juli 1998

[7] Müller, D. H., Tietjen, Th*., FMEA-Praxis - Das Komplettpaket für Training und Anwendung*, Carl-Hanser-Verlag, München, 2000

[8] Zebedin, H., *FMEA aus Sicht eines Motorenentwicklers*, in: Qualität und Zuverlässigkeit, Vol. 43, Nr. 7, Seite 826 ff., Carl-Hanser-Verlag, München,1998

[9] Gralla, D.; Heinz, S., *Fehlermöglichkeits- und Einflussanalyse FMEA*, in: EI – Eisenbahningenieur, Vol. 49, No.74, S. 43 - 47, Juli 1998

[10] Schiegg, H.; Viertlböck, M.; Kraus, T. *Prozeßbegleitend und frühzeitig - System-Produkt-FMEA mit objektiver Kennzahlbildung bei einem Automobilzulieferer*, in: Qualität und Zuverlässigkeit, Vol. 44, Nr. 7, Seite 879 - 884, Carl-Hanser-Verlag, München,1999

[11] Lutz, R. R., Woodhouse, R. M*., Requirements Analysis Using Forward and Backward Search*, in: Annals of Software Engineering, Special Volume on Requirements Engineering, 1997

[12] Mäckel, O., Schuster, J.-U., Siemens ZT/A&D, *Interner Bericht (A&D GT4/98-17): FMEA Werkzeugvergleich*, München, November 1998

[13] Kistner, W., *FMEA noch besser anwenden*, in: Qualität und Zuverlässigkeit, Vol. 41, Nr. 7, S. 827 – S. 830, Carl-Hanser-Verlag, München, 1996

[14] Möller, K. H., *Ausgangsdaten für Qualitätsmetriken - Eine Fundgrube für Analysen*, in: Ebert, C., Dumke , R. (Hrsg.) *, Software-Metriken in der Praxis*, Springer-Verlag, Berlin, 1996

[15] Mäckel, O., *Mit Blick auf's Risiko - Software-FMEA im Entwicklungsprozess softwareintensiver technischer Systeme*, in: Qualität und Zuverlässigkeit, Vol. 46, Nr. 1, Carl-Hanser-Verlag, München, 2001