Defects in Natural Language Requirement Specifications at Mercedes-Benz: An Investigation Using a Combination of Legacy Data and Expert Opinion

Daniel Ott
Research and Development
Daimler AG
P.O. Box 2360, 89013 Ulm, Germany
daniel.ott@daimler.com

***Abstract*—Natural language (NL) requirement specifications are widely used in industry, but ensuring high quality in these specifications is not easy.**

This work investigates in an empirical study the typical defect type distributions in current NL requirement specifications.

For this study, more than 5,800 review-protocol-entries that originate from reviews of real automotive specifications according to a quality-model were categorized by us at Mercedes-Benz. As a result, we obtained (a) a typical defect type distribution in NL specifications in the automotive domain, (b) correlations of quality criteria to defect severity, (c) indicators on ease of handling quality criteria in the review-process and (d) information on time needed for defect correction with respect to quality criteria. To validate the findings from the data analysis, we additionally conducted 15 interviews with quality managers.

The results confirm quantitatively that the most critical and important quality criteria in the investigated NL requirement specifications are consistency, completeness, and correctness.

***Keywords*-Large Specifications; Natural Language; Requirements, Empirical Study; Quality; Review; Interview; Automotive**

I. Introduction

It is still common that requirement specifications in industry are written in natural language (NL) [1] and the mercedes benz car development is no exception [2]. However, it is a difficult task to ensure high quality in NL requirement specifications. According to Glinz [3], it is almost impossible to produce and freeze a complete and unambiguous NL requirement specification.

It is the aim of this research to form a baseline of the distribution and severity of defect types in current NL specifications. As the research is conducted in the automotive domain, the results are mainly valid there. But it is likely that the results are transferable to other domains with similar types of specifications.

The unavailability of absolute numbers for the defect type distribution in NL requirement specifications is the problem encountered in this study. The absolute number of defects in a specification cannot be measured, because there isn't any defect detection method yet, which is certain to detect every defect. Instead review-protocols as data basis were taken, because the review is one of the most common approaches to measure the quality of non-executable requirements. Since the absolute numbers of defects cannot be measured through review analysis as mentioned above, we additionally validated the results of the reviews by procuring opinions of experts.

From a methodical point of view, we conducted two historical methods as defined by Zelkowitz et al. [4]: We performed the method "legacy data" to analyze the defect type distributions in reviews and compared the results of this analysis with the results of the method "expert opinion".

We categorized and analyzed over 5,800 review-protocol-entries from NL automotive specifications of the Mercedes-Benz passenger car development (PCD) to quality criterias. This is the first time that such a large body of data from industrial specifications has been used for an empirical study of defect type distributions in requirement specifications.

Additionally, we questioned 15 experts in requirements engineering and quality management to validate our results of the categorization and to get background information on the review-process that might have biased our findings.

In the next section research related to our work is described. Section III presents our research method. In Section IV the results of the categorization and the interviews are shown and explained. Thereafter, the limitations of our research is discussed in Section V.

II. Related work

This sectional discourse is divided into two parts: (1) other empirical studies with similar research scope or research concerning defect findings in early phases of the software life cycle and (2) other research which had influences on our work.

Through a systematic literature review, Walia et al. [5] developed a requirements related taxonomy of errors that may occur during the requirement phases of the software life cycle. The main goals behind this taxonomy are quality improvement in specifications and the support of developers in the requirement inspection process. In contrast to our research, they deal with the error behind each finding in a specification, instead of categorizing the findings to quality-attributes. For example, they investigate if the source of a

978-1-4673-2785-5/12/$31.00 © 2012 IEEE

RE 2012, Chicago, Illinois, USA

missing requirement lies within human failure, process, or documentation error.

Porter et al. [6] showed in an experiment with 24 students reviewing three requirement specifications that a scenario-based detection method is superior to ad hoc or checklist based methods. Of special interest for the present task is the separation of defects into eight quality-attributes. Unfortunately, the results show only parts of the defect type distribution in the three specifications. Additionally, the reviewed specifications are small: they consist of merely 16-31 pages, while our specifications are several hundred pages long.

Lauesen and Vinter [7] empirically investigated defect prevention techniques in a smaller company (700 employees). In their work, the defects of one product consisting of two requirement specifications (107 and 94 requirements) were analyzed and categorized. They came to the following result: Missing requirements, correctness and mistaken requirements are the major sources of defects in specifications.

Davis et al. [8] defined 24 quality-attributes that software requirement specifications should exhibit. These quality-attributes are the basis for our quality-model (see Section III-D. Their research also shows that, due to dependencies between several quality-attributes, it is not possible that all quality-attributes are fully met by a specification.

One of the most common approaches to improve the quality in requirement specifications are inspections. Aurum et al. [9] have reviewed the progress in the inspection-process since the initial work done by Fagan [10] until 2002. They give a solid overview about the further developments of inspections since Fagan's early work.

III. Research Method

We divided this section in four parts: The research question. The procedures during the categorization of the review-protocols and the description of the used data. The working steps on the expert-interviews. Finally, we describe the used quality-model in detail.

A. Research Questions

The main goal was the classification of defects in current NL specifications of the automotive industry. This led to the following research questions:

- Which defect type distribution can be found in current NL specifications by the categorization of review-protocol-entries?
- Do the results overlap with the experiences of experts in requirements engineering and quality management collected through interviews?

B. Legacy Data: Categorization of Review-Protocol-Entries to Quality-Attributes

The review-process at Mercedes-Benz PCD consists of (a) a preparation phase in which the individual reviewer checks the specification for defects and (b) a review-meeting in which the reviewers and authors discuss and rate the findings. Thereafter the authors update the specification according to the confirmed findings.

Each protocol-entry contains the following relevant information:

- an explanation of the finding
- the status of the entry
- the criticality of the finding
- the time effort (in minutes) to solve the finding in the specification

One protocol-entry contains one finding. Each finding contains one or more defects. We assigned each defect to one quality-attribute of the quality-model (see details in Section III-D).

The status can be "denied" or "accepted". When an author declares a finding as not relevant, the finding is "denied".

The criticality is the rating of the estimated impact of a finding on later project phases. The criticality can be "optimization", "uncritical", "major", "critical", or "question". The reviewers rate entries with the criticality "question", if they are not sure whether an entry is a real defect or not. It is common in the Mercedes-Benz PCD to assign the criticality of a defect depending on the influence on the customer. For instance, if a defect could cause a malfunction noticeable by customers, it is assigned as "critical".

Computer-aided, we collected the information of the protocol-entries and analyzed (a) how many defects have been assigned to the individual quality-attributes and (b) how critical defects of a quality-attribute were estimated by the reviewers for future project phases. The criticality of a quality-attribute was measured by a factor α with a scale ranging from 1.00 for "optimization" to 4.00 for "critical".

Furthermore, we analyzed indicators for the difficulty of finding defects in individual quality-attributes: The rejection-rate measures the part of defects in one quality-attribute with the status "denied". The question-rate measures the part of defects in one quality-attribute with the criticality "question".

Finally, we calculated the average time effort to solve defects for each quality-attribute.

Altogether we analyzed 49 review-protocols and categorized 5,800 protocol-entries to quality-attributes. Overall we categorized 5,999 defects to quality-attributes.

The corresponding automotive specifications to these review-protocols are in German language and have an average length of 686 requirements (15-16 words each). All of these specifications describe interior E/E systems of a car, like the seat control unit, the interior light control units or the door closure module. More detailed Information on these specifications can be extracted from previous work [11] on the same specifications. All specifications were reviewed in the requirement engineering phase and before the design phase or the transfer to external supplier.

C. Expert Opinion: Expert-Interviews

We performed a 20 minutes-interview with each expert, using the following scheme:

During the first 10 minutes, we explained the background of our research and our quality-model.

In the next 10 minutes, we asked the experts for their opinion on the following questions:

- Which are the quality-attributes with the most frequently assigned defects in specifications?
- Which are the most difficult quality-attributes to find?
- Which are the most important quality-attributes?
- Which criticalities are assigned to quality-attributes?
- Defects of which quality-attributes are frequently denied by the authors?
- Defects of which quality-attributes are frequently marked with the criticality "question" by the reviewer?
- Defects of which quality-attributes are not relevant to or ignored by the reviewer?

The 15 interviewed quality-managers are experts in requirements engineering and quality management. All of them are continuously involved in the review-process or taking part in solving defects in specifications in different roles: five specification authors, six quality managers, two software-/hardware-architects and two tool-/requirement-supporter were interviewed to obtain widespread experience of all relevant roles in the quality management of NL requirement specification in the Mercedes-Benz PCD. On average, each expert has participated in 60 reviews on similar specifications to the current.

D. Quality-Model

Our quality-model is based on the research of Davis et al. [8]. Additionally, we integrated quality-attributes from literature [12], [13], [14], [15].

The quality-attributes of the various literature sources are put in a hierarchical structure (our quality-model) to show dependencies between quality-attributes. Afterwards, quality-attributes that arise from other quality-attributes (e.g. understandability: this quality-attribute arises from unambiguity, correctness, completeness, and conciseness) are excluded.

Furthermore, the hierarchical structure helps to exclude quality-attributes which are never met (e.g. executable) or always met (e.g. shared access) by specifications from the Mercedes-Benz PCD.

The resulting quality-model has a hierarchical depth of up to 4 levels, for example: "Overlapping contents" is part of the "internal consistency". "Internal consistency" itself belongs to "content consistency" which is included in "consistency". The quality-model consists of 78 entries. Due to space restriction and for better comprehensibility and readability, we created a coarse grained version of our model. We simply united all quality-attributes in deeper levels to the superior quality-attribute on the first level. This reduced model (see Table I) was used during the interviews and the full model was used for the categorization of the review-protocol-entries.

IV. Results

A. Legacy Data: Results from the Review-Protocol-Analysis

Before we started the analysis, we scanned all entries and made a pre-selection to filter out useless entries: Out of the 5,999 defects, which we categorized to quality-attributes, 148 defects had no criticality, so we used them for the analysis of the defect type distribution but not for the criticality-analysis. 963 defects with status "denied" were solely taken for the rejection-rate-analysis and 488 defects with criticality "question" only for the question-rate-analysis.

Table II shows the results of our analysis. These results are explained in the following paragraphs and some additional results that cannot be seen in the Table because of the abstraction of the quality-model will be pointed out.

The first column "Distribution" shows the result of our distribution-analysis: We have taken 5036 defects into account in this analysis. We excluded defects with status "denied". The first column shows the distribution in absolute numbers and the second in percentages.

The next column "Criticality" shows the result of our criticality-analysis: The factor α has a scale ranging from 1.00 for "optimization" to 4.00 for "critical".

The second subcategory in column "Criticality" states again the absolute number of defects for each quality-attribute. This number differs from the absolute number by the distribution-analysis in the column "Distribution" because defects with no criticality have been removed.

The columns "Rejection" and "Question" show the results of the rejection- and question-rate-analysis. These results are divided up into each quality-attribute in the number of rejections / questions in absolute numbers, and in percentages, how large this part is in comparison to all defects of this quality-attribute.

The subcategories of the last column "Time effort (in min)" show the time-effort-analysis to solve defects in specifications with the average time effort, the sum of all times and the number of findings. Unfortunately, the time effort analysis is of poor quality, as merely 478 out of the 5800 protocol-entries had registered times. So these results present just an idea of the time effort needed to correct the stated findings in the specification.

B. Expert Opinion: Results from Interviews

Table III shows the results of the interviews to validate the results of the review-protocol-analysis in Table II. We asked the experts to vote for up to three quality-attributes for each of the following questions (see Section III-C). Voting the

Table I
REDUCED QUALITY-MODEL

Quality	Description
Atomicity	A specification entry is atomic, if it cannot be usefully separated in more entries.
Unambiguity	A specification is unambiguous, if (a) all entries have the same meaning for all readers and (b) the specification includes no ambiguous words, phrases or sentences.
Conciseness	A specification is concise, if (a) no unnecessary implementation details are described and (b) the specification contains no unnecessary entries or entries that could be formulated shorter.
Testability	A specification is testable, if (a) each requirement has a method to check, if the system satisfies this requirement and (b) it is possible to derive test cases from the requirement.
Traceability	A specification is traceable, if (a) every reader can retrace the source and the further usage in the project life cycle of every requirement and (b) all redundant, interdependent and complementary information are set into dependency.
Consistency	A specification is consistent, if there are no overlaps in the content of requirements (a) in the specification, (b) between the specification and other relevant documents and (c) all terms are consistently used. Special cases for overlaps are conflicting or redundant requirements.
Formal Correctness	A specification is formally correct, if (a) there are no linguistic defects (e.g. spelling, punctuation or grammatical mistakes) and (b) the specification is structured and supported in a way by graphical representations that every reader has the best possible support receiving information from the specification.
Correctness	A specification is correct, if (a) there are no terms, phrases or sentences in requirements with false content, (b) all requirements are at the right location, and (c) the specification reflects all currently valid requirements.
Completeness	A specification is complete, if (a) every requirement is classified according to its importance, priority, necessity and liability, (b) if no requirements, parts of requirements or additional documents are missing, and (c) each possible quantitative information has been specified, and all terms are defined.

Table II
RESULTS REVIEW-PROTOCOLS

Quality-attribute	Distribution		Criticality		Rejection		Question		Time effort (in min)		
	#	%	α	#	#	%	#	%	$\varnothing$	$\sum$	#
Atomicity	52	1.03%	2.58	52	7	11.86%	3	5.08%	20.00	160	8
Unambiguity	54	1.07%	2.75	52	7	11.48%	13	21.31%	4.67	14	3
Conciseness	228	4.53%	2.28	215	45	16.48%	28	10.26%	4.33	65	15
Testability	3	0.06%	3.33	3	1	25.00%	0	0.00%	-	-	0
Traceability	274	5.44%	2.46	272	23	7.74%	18	6.06%	3.11	28	9
Consistency	433	8.60%	2.45	417	69	13.75%	36	7.17%	7.54	309	41
Formal Correctness	668	13.26%	1.68	665	92	12.11%	33	4.34%	5.11	378	74
Correctness	749	14.87%	2.73	720	147	10.69%	74	4.55%	8.06	774	96
Completeness	2575	51.13%	2.78	2492	545	17.47%	283	9.07%	7.97	1848	232
Total	5036	100.00%	2.55	4888	963	16.05%	488	8.13%	7.48	3576	478
Basis for analysis (defects)	5036		4888		5999				5036		

same quality-attribute more than once was allowed to show its importance.

For the first column “Frequency” the experts were asked to choose the three most common defect types in specifications.

The second column “Difficulty” shows the experts’ top three of the most difficult to find defect types in specifications. Consistency was rated as difficult to find, because many reviewers cannot manually check the huge number of entries in the specifications and additionally needed documents in their limited time for the review. The amount of entries in the additionally needed documents is typically 10-20 times larger than the amount of entries in the specification itself. Completeness issues are also difficult to find for the reviewers, because of missing knowledge caused by limited preparation time.

The next column “Priority” shows the results of the top three quality-attributes with the highest priority for the reviewers. Specifications are mainly used as an agreement between Mercedes-Benz and subsidiaries or external suppliers. For this reason, it is most important that the specification is complete, unambiguous, and correct, to narrow the scope of possible actions for the suppliers.

For the next column “Criticality” we asked the experts, if they could assign a criticality focus to defects of a quality-attribute. Some experts rejected this idea, mentioning that this depends on the specification, the specification author and the release deadline.

The next column “Rejection” shows the results of the top three quality-attributes rejected by authors. The defects in correctness and completeness are often rejected because they are no defects in authors’ opinion. All other quality-attributes are often rejected because of the perceived bad effort-benefit-ratio in the perspective of the authors.

The column “Question” shows the quality-attributes for which the reviewers often needed a consultation with the specification authors.

The last column “Ignore” shows the quality-attributes that reviewers check with low priority or completely ignore. A reason are often rejected quality-attributes by the authors, because of their limited time for updates in the specification. Because the reviewers know this, they tend to check with

Table III
RESULTS INTERVIEWS

Quality-attribute	Frequency	Difficulty	Priority	Criticality		Rejection	Question	Ignore
	#	#	#	α	#	#	#	#
Atomicity	6	1	1	2.00	1	5	-	1
Unambiguity	9	2	7	3.50	2	4	4	1
Conciseness	1	1	2	2.00	1	4	-	-
Testability	7	1	6	-	-	5	2	4
Traceability	2	5	2	3.50	1	1	1	4
Consistency	8	11	4	3.67	3	2	4	4
Formal Correctness	9	1	6	1.63	12	13	4	15
Correctness	7	4	15	3.75	2	6	8	3
Completeness	16	17	26	3.50	13	5	14	5

low priority or ignore such defects with low effort-benefit-ratio in the author's opinion.

The following points summarize some additional impressions the experts gave us during the interviews:

- The long time span between specification updates (up to two years) is a problem for authors, who are mainly developers. Because of this, there are major problems in ensuring correctness, consistency, and traceability between old and new specification parts as well as between all old and new additional documents.
- Ensuring correctness, consistency, and traceability is additionally difficult because of the strongly increasing number of component and system variants. In the case of Mercedes-Benz, there are variants, for example, for different car series, country-specific restrictions, or recently because of new powertrain technologies beyond gasoline engines.
- Testability is often postponed after the specification release to the supplier and so hardly receives attention during the reviews. Testability is a important quality-attribute for authors and reviewers, but it is not relevant that the requirements are already testable when they are sent to the supplier.

C. Primary Findings

Overall, the results of the review-protocols and interviews do not fully overlap, but the reasons for this are depicted in detail hereafter:

The consensus of both analyses is that the majority of defects in the specifications are assigned to content quality-attributes, like consistency, correctness, and completeness. About 70% of the classified defects are assigned to these quality-attributes and at the same time they belong to the most critical quality-attributes. Concurrently, the interviews with the experts showed indicators (for instance the high priority and criticality scale in Table III), that these quality-attributes belong to the most difficult and important quality-attributes.

Although unambiguity as well as testability are frequent and important findings in the opinion of the reviewers, they are also often ignored by them and rejected by authors. Hence they have no significant occurrences in the review-protocols. One reason for this is the large potential for discussion and the uncertainty-factor over the correctness between author and reviewer as the question-rates in Table II and Table III indicate. Another reason is the insufficient effort-benefit-ratio of these defects for the authors. Also, testability is often postponed after the specification release to the supplier. It is therefore often ignored in the reviews as stated in the interviews.

Due to the same reasons, although it has high frequency, formal correctness has low priorities, hence a rather low criticality. So they, too, are often ignored by reviewers and rejected by authors.

V. THREATS TO VALIDITY

This work was subject of a number of threats to validity. A threat to **construction validity** is the unavailability of absolute numbers for the defect type distribution in NL requirement specifications, because until now there exists no defect detection method, which can guarantee that every defect is detected correctly. Instead we took review-protocols as a data basis, because the review is one of the most common approaches to measure the quality of non-executable requirements. To estimate the quality of the review-protocols as a measure for the real defect type distribution, we interviewed 15 experts in requirement engineering and quality management. The outcome of the interviews showed that the results of the review-protocols indeed have certain deficits:

- Testability is often postponed after the specification release to the supplier and so hardly receives attention during the reviews.
- Unambiguity and formal completeness are often rejected by authors because of insufficient effort-benefit-ratio. Because of this, reviewers also tend to check these quality-attributes with low priority or ignore them (see Table III).
- Consistency and completeness issues are rated by the experts as very difficult to find (see Table III). This is also an indicator for possible not found defects in the examined specification.

Therefore, these quality-attributes may have a higher number of defects than the results showed.

Another problem with the review-protocols is that we are not their authors. This leads to the possibility that we have not put every defect to the right quality-attribute category, because of wrongly interpreted entry-contents. This is also a threat to **reliability** because other researchers could have a different interpretation of some of the review-protocol entries than we have.

Threat to the **External validity** are limitations in the transferability of our results on NL specifications drawn from the Mercedes-Benz PCD to specifications from other companies in the automotive industry or even to specifications from other industries. Issues in the transferability are differences to other companies in the specification structure, the creation process, the quality management (e.g. other tool support or other review techniques), or another specification size and complexity.

The aspect **internal validity** is not explicitly discussed, since all factors that can affect the investigated quality-attributes are also threats to to the construction validity which is is discussed above.

VI. Conclusion and Future Work

This paper provides a snapshot of the defect type distribution in recent specifications in usage. Therefore, for the first time, the results of the analysis of a large body of data from the Mercedes-Benz PCD are presented. This data consists of 49 review-protocols from automotive specifications and 15 interviews with quality-managers.

These results showed that the majority of defects are assigned to content quality-attributes like completeness, correctness and consistency. They also showed that these quality-attributes are the most critical, difficult and important.

This work contributes to the understanding of the problems developers have to face in practice ensuring the quality in NL specifications. Additionally, this work will support researchers and practitioners in software engineering:

For researchers, this work can serve as a solid baseline for future research in improving quality, especially content quality, in requirement specifications.

For practitioners, this work gives the opportunity to compare their defect type distributions in specifications from their industries with our present results. This may lead to an exchange of experiences and therefore to additional results or a cross-industry picture of the defects in requirement specifications.

The results of this work give practitioners the opportunity to monitor their review-process. A review-protocol considering all quality-attributes should have a similar defect type distribution as seen in our results.

The results of this work will be used as baseline for future research in semi-automatic approaches to classify content defects and to improve the overall quality in NL requirement specifications, especially at Mercedes-Benz.

References

[1] L. Mich, M. Franch, and I. Novi, "Market research for requirements analysis using linguistic tools," *Requirements Engineering*, vol. 9, no. 2, pp. 40–56, 2004.

[2] F. Houdek, "Challenges in Automotive Requirements Engineering," *Industrial Presentations by REFSQ 2010, Essen*, 2010.

[3] M. Glinz, "Improving the quality of requirements with scenarios," 2000. [Online]. Available: http://www.ifi.uzh.ch/arvo/req/ftp/papers/2WCSQ.pdf

[4] M. Zelkowitz, D. Wallace, and D. Binkley, "Experimental validation of new software technology," *SERIES ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING*, vol. 12, pp. 229–263, 2003.

[5] G. Walia and J. Carver, "A systematic literature review to identify and classify software requirement errors," *Information and Software Technology*, vol. 51, no. 7, pp. 1087–1109, 2009.

[6] A. Porter and L. Votta, "An experiment to assess different defect detection methods for software requirements inspections," in *Proceedings of the 16th international conference on Software engineering*. IEEE Computer Society Press, 1994, pp. 103–112.

[7] S. Lauesen and O. Vinter, "Preventing requirement defects: An experiment in process improvement," *Requirements Engineering*, vol. 6, no. 1, pp. 37–50, 2001.

[8] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaram *et al.*, "Identifying and measuring quality in a software requirements specification," in *Software Metrics Symposium, 1993. Proceedings., First International*. IEEE, 2002, pp. 141–152.

[9] A. Aurum, H. Petersson, and C. Wohlin, "State-of-the-art: software inspections after 25 years," *Software Testing, Verification and Reliability*, vol. 12, no. 3, pp. 133–154, 2002.

[10] M. Fagan, "Design and code inspections to reduce errors in program development," *IBM Journal of Research and Development*, vol. 15, no. 3, p. 182, 1976.

[11] J. Leuser and D. Ott, "Tackling semi-automatic trace recovery for large specifications," in *REFSQ 2010*, ser. Lecture Notes in Computer Science, R. Wieringa and A. Persson, Eds., no. 6182. Springer, Heidelberg, June 2010, pp. 203–217.

[12] C. Rupp, *Requirements-Engineering und -Management*. Hanser, 2009, vol. 5.

[13] K. Pohl, *Requirements Engineering*, 1st ed. dpunkt.verlag, 2007.

[14] B. W. Boehm, "Verifying and validating software requirements and design specifications," *IEEE Softw.*, vol. 1, no. 1, pp. 75–88, 1984.

[15] *IEEE-830: Recommended practice for software requirements specifications*, IEEE Std., Oct 1998.