Software Traceability in the Automotive Domain: Challenges and Solutions

Salome Maro, Jan-Philipp Steghöfer, and Miroslaw Staron

Chalmers | University of Gothenburg, Sweden
{salome.maro, jan-philipp.steghofer, miroslaw.staron}@cse.gu.se
Published in the Journal of Software and Systems (JSS) 2018

Abstract

In the automotive domain, the development of all safety-critical systems has to comply with safety standards such as ISO 26262. These standards require established traceability, the ability to relate artifacts created during development of a system, to ensure resulting systems are well-tested and therefore safe. This paper contrasts general traceability challenges and solutions with those specific to the automotive domain, and investigates how they manifest in practice. We combine three data sources: a tertiary literature review to identify general challenges and solutions; a case study with an automotive supplier as validation for how the challenges and solutions are experienced in practice; and a multi-vocal literature review to identify challenges and solutions specific to the automotive domain. We found 22 challenges and 16 unique solutions in the reviews. 17 challenges were identified in the case study; six remain unsolved. We discuss challenges and solutions from the perspectives of academia, tool vendors, consultants and users, and identify differences between scientific and "grey" literature. We discuss why challenges remain unsolved and propose solutions. Our findings indicate that there is a significant overlap between general traceability challenges and those in the automotive domain but that they are experienced differently.

Keywords: Traceability, Automotive, Safety, ISO 26262, Automotive SPICE

1. Introduction

Over the past 20 years, the automotive domain has witnessed a tremendous increase of software deployed in cars. In today's modern car, software constitutes up to 40% of the production cost [1]. With upcoming trends such as autonomous driving, the software is not only getting more complex but also controls more and more safety-critical functions. The type of software has also shifted from small isolated functions to systems that contain several functions that interact and depend on each other [2]. Such complex systems can cause life threatening accidents when not properly specified, designed, implemented and tested. The number of artifacts produced during development (e.g., requirements, design models, behaviour models, simulations and tests) is large and their creation is usually distributed over various teams, including teams from different companies due to OEM-supplier relationships. With regards to the size of the systems, a typical high-end car consists of features that amount to about 100 million lines of code. This is a very large number as software in other domains has much less lines of code. For example, the F-22 fighter jet has about two million lines of code and the Boeing 787 has around 14 million lines of code [3]. In addition, not only is there a tremendous number of lines of code in this domain but also a large number of other artifacts. For instance, the specifications of the systems in a 2004 car had already reached 20000 pages at that time [4]. This can be overwhelming if there are no standardized methods established to keep track of these artifacts, their relationships, and how they evolve.

In such situations, traceability plays an important role. In this paper, we define traceability as the ability to relate artifacts created during the development of a system, thus following [5]. Traceability helps in understanding which artifacts are connected to each other and allows to keep track of which features have already been specified, implemented and tested. Traceability plays an even bigger role for maintenance tasks by facilitating change impact analysis and improving understandability of the system for the developer who needs to make changes in the system [6, 7]. In the automotive industry, these aspects are particularly important in light of safety standards that require proof that safety requirements were specified, taken into account during the design and development, validated in test cases, and verified through safety analysis (see, e.g., [8, 9]).

In order to realize the benefits of traceability (and successfully argue their safety cases), software development companies need to establish a traceability strategy that is aligned with their goals. Defining and implementing a traceability strategy is not a trivial task, since it requires a good understanding of the artifacts to be traced as well as the ability to define meaningful links and to make sure the created links are useful [10].

On the one hand, there exists a large body of knowledge on traceability; for instance between 1999 and 2012, 70 studies on traceability were published in just the Requirements Engineering (RE) conference [11]. On the other hand, in practice traceability is either not established at all [12] or only established since standards demand it [13]. In addition, it is unclear how the traceability challenges in the automotive domain relate to general traceability challenges and how they manifest in a practical environment of a company.

The contribution of this paper is therefore to provide an exhaustive empirical evaluation of traceability challenges and solutions in the automotive domain that takes the specific characteristics of automotive software development into account. To achieve this, we conducted a tertiary literature review, a case study, and a multi-vocal literature review. This allows us to explore the traceability problem in the automotive domain from both the practical and the scientific perspective and provides insight into the challenges of traceability as they present themselves in practice as well as solution approaches proposed by academia, tool vendors, consultants, and the users of traceability themselves. Our aim is to give insights on which challenges exist in this domain, the spectrum of solutions available, and highlight difficulties experienced with using some of these solutions in the automotive domain. Our study therefore investigated the following research questions:

RQ 1: What are the general traceability challenges and solutions reported in literature?

RQ 2: What are the particular traceability challenges and solutions in the automotive domain?

RQ 3: Which of the reported traceability challenges in scientific literature and non-scientific literature can be observed in practice in the automotive domain and how have they been solved?

To obtain data for our study, we used three different data sources: a tertiary literature review in which we reviewed 24 secondary publications on traceability; a case study at an automotive supplier company; and a multi-vocal literature review in which we reviewed a total of 245 scientific and non-scientific sources. We found 22 challenges from the literature of which 17 were also found in our case company. Five of the challenges have been solved with solutions proposed in literature, six are partially solved while six remain unsolved even though there are proposed solutions in literature.

This paper extends our work reported in [14] in which we discussed challenges related to creation, maintenance and exchange of traceability by also discussing traceability challenges related to preparation and planning for traceability and the use and measurements of traceability. We have also added the multi-vocal literature review as an additional data source and extend our results and discussion with this new trove of information. Furthermore, we review persisting challenges in detail and give an overview for solutions viable in the automotive domain.

The rest of the paper is structured as follows: Section 2 describes traceability requirements in the automotive domain and our research method is described in detail in Section 3. Sections 4 to 7 present the challenges and solutions and describes them from the perspective of the tertiary and multi-vocal literature and how they relate to the case company. Section 8 provides a discussion of the results. Limitations of the study are discussed in Section 9, Section 10 discusses related work, and Section 11 concludes the paper and outlines future work.

2. Traceability Requirements in the Automotive Domain

In this domain, traceability requirements are imposed by the ISO 26262 [15] – a functional safety standard for road vehicles – and ASPICE [16] – a process assessment model specific to the automotive domain. Both the ISO 26262 and ASPICE prescribe the use of a V-model process lifecycle for *product* development of embedded systems. The traceability links required are shown as dotted lines in Figure 2. It is important to note that due to the overlap in ASPICE and ISO26262 [**?**], these standards are usually used in companies to complement each other, rather than as two separate alternatives. Since ASPICE is a process assessment model, it can, e.g., be used to assess the maturity of a process with extensions that also cover the safety critical aspects prescribed in ISO26262 [**?**]. In summary, both standards impose the following with respect to traceability:

Vertical traceability: Artifacts must be traceable to their children and the children should be traceable to their parents (*bi-directional traceability*). An example of this is that a requirement should be traceable to architectural artifacts (structural and behavioral) that realise it and to the code associated with these artifacts. It should also be possible to trace from the code to architectural artifacts and back to the requirement.

Horizontal traceability: This means that it should be possible to trace from artifacts on the left side of the V-model to their verification artifacts (such as tests or safety analysis) on the right side of the V-model. In addition, traceability links should be created and maintained between any recorded change requests and the work products affected by them to enable change impact analysis.

From a traceability point of view, the main difference between the two is that ISO 26262 requires traceability to be established between safety-related artifacts, i.e., it requires defining links from hazards to safety goals, to safety requirements, to the structure and behavior of these safety requirements, to the code and to tests that are responsible for testing all the safety artifacts. ASPICE on the other hand requires traceability for *all* artifacts, even if they are

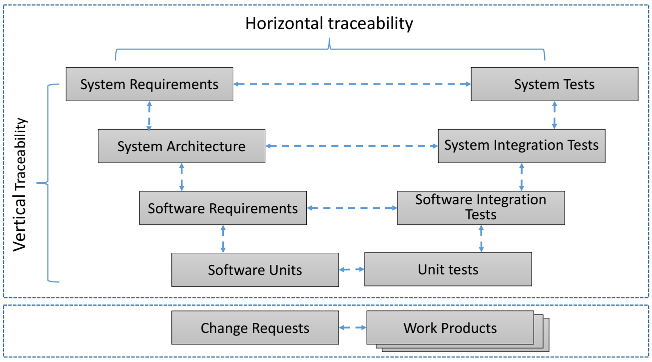

Figure 1: Traceability in the V-model

not safety-related. Another difference is that while the ISO 26262 standard *recommends* bi-directional traceability, it is deemed mandatory in ASPICE. The term "recommended" in ISO 26262 implies that companies are free to choose other alternatives to show that the requirements have been fulfilled. Additionally, the ISO 26262 standard requires the artifacts to be versioned and have unique identifiers in order to be traceable.

Apart from the requirements imposed by the standards, traceability processes and tools in this domain need to deal with characteristics such as complexity, longevity, and variability of the products as well as the distributed development environment. In this situation, traceability can, e.g., help with program comprehension [17], to allow change impact analysis [18], and to document rationale and design procedures [19]

3. Research Method

The aim of our study is to get an understanding of the traceability problem in the automotive domain and thus to answer the following research questions:

RQ 1: What are the general traceability challenges and solutions reported in literature?

RQ 2: What are the particular traceability challenges and solutions the automotive domain?

RQ 2a: What are the challenges and solutions regarding traceability when addressing the demands of automotive standards ASPICE and ISO 26262?

RQ 2b: What are additional relevant traceability challenges and practices in the automotive industry?

RQ 3: Which of the reported traceability challenges in scientific literature and elsewhere can be observed in practice in the automotive domain and how have they been solved?

To answer these research questions, we collected data from general and specific scientific literature, from a case study and from specific non-scientific literature. To achieve this, we used three types of research methods: a tertiary literature review, a case study with an automotive supplier, and a multi-vocal literature review. The tertiary literature review provided data on the challenges and solutions in the literature (RQ 1). The multi-vocal literature review allowed us to include information about challenges and solutions that were not reported in scientific literature and were thus not covered in the tertiary literature review (RQ 2a and RQ 2b, RQ 3). Since the adoption of standards in the automotive domain is evolving quickly, RQ 2a addressed this topic specifically. We regarded RQ 2b as an auxiliary research question, designed mainly to provide us with additional material that does not mention the automotive standards. The case study provided data on which challenges exist in practice and their solutions if any (RQ 3).

3.1. General Guidelines and Scope

We conducted the tertiary literature review according to the guidelines in [20], the case study according to the guidelines proposed in [22] and the multi-vocal literature review according to the guidelines proposed in [21]. Before conducting these studies, we defined the scope relevant to us and which all three data sources cover. Our scope (depicted in Figure 2) indicates that we distinguish four different traceability categories (*Preparation and Planning*, *Establishment*, *Outcome* and *Exchange*) which are inspired by the generic traceability process model defined by Gotel et al. [10]. We used this model because it contains most of the activities needed for establishing traceability. This model is also well-known in the traceability community and since, its definition has been used in other research, (e.g., in [23, 24, 25]) as a basis for understanding and describing traceability.

In the model, the *Preparation and Planning* category, focuses on the processes and tools involved when preparing to include traceability in a company or a particular project. The *Establishment* category deals with the processes and tools involved in the actual creation and maintenance of traceability links. The *Outcome* category focuses on how the links are stored and how they are actually used after they have been established. Since we are studying the automotive domain where the OEM-Supplier relationship means that artifacts are exchanged between companies, we added a fourth category called *Exchange* where we discuss challenges of exchanging traceability within and between organizations.

The details of the tertiary literature review are described in Section 3.2, those of the multi-vocal literature review in Section 3.3, and those of the case study are described in Section 3.4. The entire research process is summarized in Figure 3.

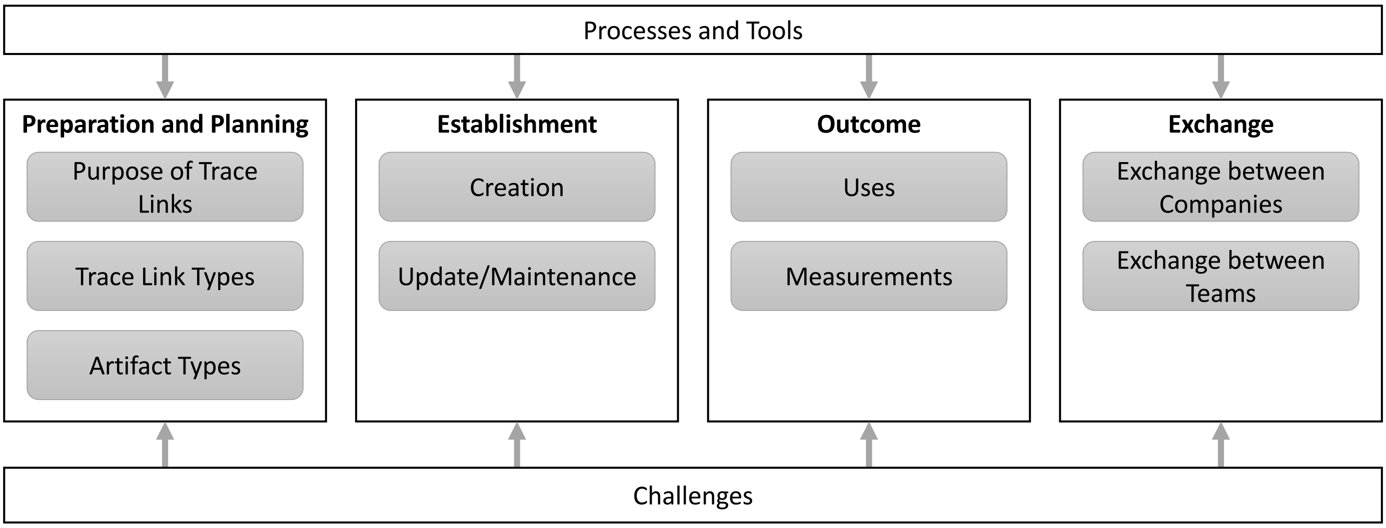

Figure 2: The scope of the case study

3.2. Tertiary literature review

Our tertiary literature review followed the guidelines for conducting a systematic mapping study as proposed by [20]. The guidelines indicate that a systematic literature study should include five steps which are *Definition of research questions*, *Conduct search*, *screening of papers*, *Keywording using abstracts* and *Data extraction & mapping process*. The subsections below describe how these steps were carried out in our study.

3.2.1. Definition of Research Questions

Our aim is to identify both general traceability challenges and solutions from the literature study that we can later compare to the specific challenges and solutions in the automotive domain from the multi-vocal literature review and the case study. Therefore our literature study has to answer the following research question:

RQ 1: What are the general traceability challenges and solutions reported in literature?

3.2.2. Conducting the Search

Since this is a tertiary literature review, our aim was to find literature reviews published on traceability in the domain of computer science. We searched three databases : Scopus, ACM Guide and IEEE Xplore. The search strings used are shown in Table 1. This search led to a total of 522 papers which were reduced to 370 by removing duplicates.

3.2.3. Screening of Papers

By reading the title and abstract, we selected papers that are relevant to our study using the following inclusion criteria:.

1. The paper reviews literature on traceability.
2. The paper is published in a peer-reviewed venue.
3. The paper is in the field of computer science.
4. The paper mentions challenges and solutions of traceability and gives a description of these challenges and solutions.
5. The paper is in English or German.

The initial screening in which we read the title and abstract left us with 27 relevant papers. After this we further read the introduction and conclusion of the papers and excluded eleven more papers because they did not fulfill criteria number one or four. From the remaining 13 papers we used both forward and backward snowballing to look for papers that specifically addressed challenges of traceability. We limited our snowballing to papers published between 2007 and 2017 to ensure that we get current traceability challenges. This led to an addition of eleven more papers. In the end, we had identified a set of 24 relevant papers.

3.2.4. Data Extraction and Classification

We examined all 24 papers, extracted all the challenges and solutions they report and listed them in a spreadsheet. After this process, we reviewed all the challenges in the list and placed each challenge in the best-fitting subcategory in the conceptual model shown in Figure 2. At this stage, we observed that some of the challenges could be placed in more than one sub-category (for instance, the challenge of *Manual work* could be placed in both the creation and maintenance sub-category). We therefore merged these sub-categories. Afterwards we reviewed the challenges and discovered that they could be further distinguished by challenges about technical issues in particular with the tool support, human factors that involved employees, and the organisational setting and established processes. Therefore we divided the merged category of creation and maintenance into three sub-categories: tool support, human factors and organisation and processes. Additionally, in the *Preparation and Planning* category,

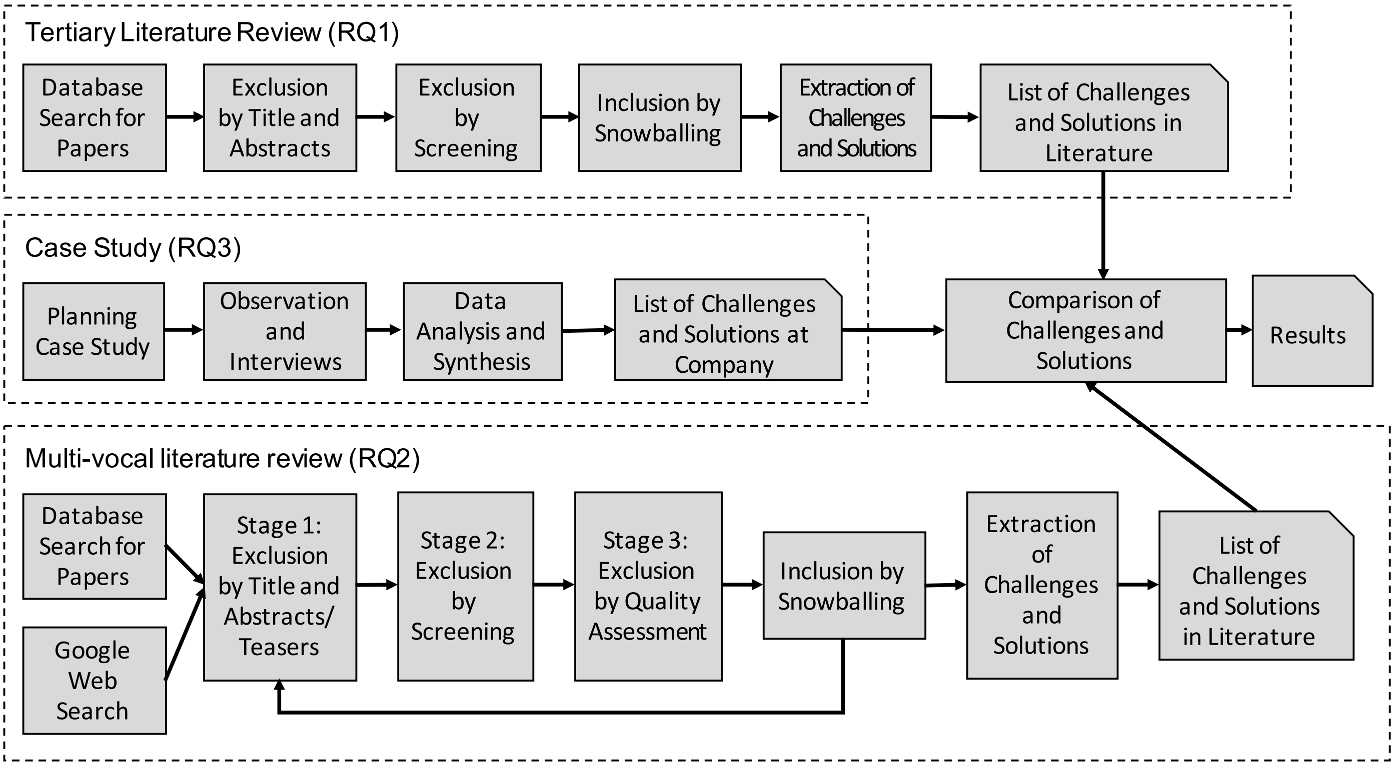

Figure 3: Summary of the Research Method

all of the challenges found were related to the general understanding of traceability. We therefore merged the three sub-categories (Purpose of trace links, trace link types and artifact types) into one category "knowledge of traceability". It also became clear that the distinction between company-internal and external *Exchange of Traceability* was not helpful, in particular since a *Lack of coordination* could be found in both sub-categories. The *Outcome* category remained the same as in our conceptual model.

3.3. Multi-vocal Literature Review

To answer the sub-research questions on practices for traceability in the automotive domain (RQ 2a and RQ 2b), we use a multi-vocal literature review (MLR) [21]. This strategy allows us to include non-academic sources (sometimes called "grey literature"). Since many practitioners do not publish their experiences in scientific venues, but do publish whitepapers, write blog entries, or give presentations at trade shows and conferences, this allows us to better grasp the current state of the art and practice. Due to the large number of sources, this approach requires very strict selection criteria for the sources. We have detailed these criteria below. We used SCOPUS, IEEE Xplore and ACM Guide as sources for scientific literature and Google Web Search to find non-academic sources.

We used the general protocol suggested in [26] for the MLR. It contains three stages of evaluation and combines systematic and "opportunistic" discovery. While the former form of discovery is covered by the searches in scientific databases and Google Web Search, the latter includes papers that were recommended by colleagues, the results of snowballing, or discoveries of sources that were otherwise incidental and not part of the systematic search. The evaluation is based on the source title, teaser text, and quickly following the link to check the source (Stage 1), the entire text (Stage 2), and an assessment of the overall quality of the source (Stage 3).

3.3.1. Definition of Research Questions

Our aim is to find traceability practices relating to standards compliance in the automotive industry and challenges and solutions to traceability that are not reported by the scientific literature. We therefore use the MLR method to answer the following research questions:

- RQ 2a: What are the challenges and solutions regarding traceability when addressing the demands of automotive standards ASPICE and ISO 26262?
- RQ 2b: What are additional relevant traceability challenges and practices in the automotive industry?

The data collected in the MLR will also support our answers to *RQ 3: Which of the traceability challenges reported in scientific literature and non-scientific literature are also evident in practice in the automotive domain and how have they been solved?*

3.3.2. Conducting the Search

As mentioned above, we use Google Web Search to look for non-scientific sources. In order to keep the num-

Table 1: Search strings used in the tertiary and multi-vocal literature review

Data Source	Search String	Number of results
Tertiary Review		
IEEE Xplore	(‘‘Literature Review" OR Review OR Survey OR ‘‘Literature Survey") AND Traceability	160 results
SCOPUS	(Literature Review OR Review OR Literature Survey OR Survey) AND Traceability	40 results
ACM Guide ˆ	(Literature Review OR Review OR Literature Survey OR Survey) AND Traceability	322 results
Multi-vocal review, RQ 2a		
IEEE Xplore	(((((Automotive SPICE) OR ASPICE) OR ISO 26262) OR ISO26262) AND Traceability)	8 results
SCOPUS	((aspice OR automotive AND spice OR iso 26262 OR iso26262) AND (traceability))	25 results
ACM Guide	+(ASPICE "Automotive Spice" "ISO 26262" ISO26262) +(Traceability)	15 results
Google Web Search	(aspice OR "automotive spice" OR "iso 26262" OR iso26262) AND traceability	approx. 34700 results (263 after filtering)
	Google filters the search results to exclude similar results, parantheses and AND operator added for clarity.	
Multi-vocal review, RQ 2b		
IEEE Xplore	((((Automotive Traceability) NOT rfid) NOT barcode) NOT laser) in the time span between 2008 and 2018	40 results
SCOPUS	(TITLE-ABS-KEY (automotive AND traceability) AND NOT TITLE-ABS-KEY (manufacturing) AND NOT TITLE-ABS-KEY (rfid)) AND (LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "COMP")) AND PUBYEAR > 2008 AND PUBYEAR < 2018	163 results
ACM Guide	+(Automobilindustrie Automotive Automobil) +(Traceability Nachverfolgbarkeit Rückverfolgbarkeit Verfolgbarkeit) -Manufacturing -Food -Laser -Barcode -"supply chain" -"logistics chain" -"lot tracking" -chargenrückverfolgung -rfid) in the time span between 2008 and 2018	95 results
Google Web Search	((((Automobilindustrie) OR Automotive) OR Automobil) AND ((((Traceability) OR Nachverfolgbarkeit) OR Rückverfolgbarkeit) OR Verfolgbarkeit))	approx. 189000 results (356 after filtering)

ber of results manageable, we let Google filter redundant entries. The number of results Google reports on the first page of the search results indicates the total number of hits in the index. The final number (after filtering) becomes evident when going through the search results page by page and navigating to the final results page[1]. We used Google Search for our campus location (Gothenburg, Sweden) in early-to-mid August 2017. The search for RQ 2a was conducted with a fresh user profile without any cookies or history in Google Chrome to avoid influencing the results through tracking cookies. The search for RQ 2b was conducted with a simple command line script, again avoiding any tracking. Searches for RQ 2a and RQ 2b have been performed with different IP addresses to also avoid tracking of this information. The search terms have been intentionally kept vague to increase the breadth of the found sources, even if this increased the effort for the screening of the sources.

To answer **RQ 2a**, we combined the two pre-dominant process standards Automotive SPICE and ISO 26262 with the term traceability. Google Web Search reported approx. 34700 results in its index. When browsing through the result pages, the total number of hits is 263. We exported the search results to a spreadsheet and enriched it with meta-data as described below. The number of search results for scientific sources was significantly smaller, with a large overlap between the data bases. The exact search terms and result numbers are shown in Table 1:

To answer **RQ 2b**, we used a very general core search term (“automotive traceability”) that we paired with exclusion terms to reduce the number of hits (cf. Table 1). For the Google Web Search, we also included German search terms to look for information provided by the sizable German automotive industry. This was not done for scientific literature since the used databases focus on En-

[1]For a more technical discussion of this filtering, please refer to https://support.google.com/gsa/answer/6329272. In essence, Google filters duplicates and only displays the two most relevant similar pages on the same domain.

glish language literature. As there is no unique German translation for "traceability", we use different options that have been used in literature based on a preliminary search: "Nachverfolgbarkeit", "Rückverfolgbarkeit", and "Verfolgbarkeit". Unfortunately, these terms are not only used in the context of software, but also in the context of being able to trace work pieces in a manufacturing process or products in a supply chain. We thus decided to refine the search term to exclude these irrelevant results. Based on a preliminary sighting of results, we excluded the terms "Manufacturing", "Food", "Laser", "Barcode", "supply chain", "logistics chain", "lot tracking" "Chargenrückverfolgung", and "rfid". This increased the relevance of the search results significantly. Since we were interested in traceability challenges and practices that are currently relevant, we reduced the relevant time span for scientific literature to 2008 to 2018.

As for the data for RQ2a, we exported the search results to a spreadsheet and enriched them with meta-data. To check if there were additional reports by automotive OEMs that we did not pick up through this search, we also searched for combinations of OEM names with traceability and its German counterparts. No additional sources were found this way.

3.3.3. Screening of Sources

We screened the papers in three stages, similar to the way described in [26]. In Stage 1, we excluded sources based on the title of the page and the description by Google Web Search and followed the link to determine if the information was accurate, which type of source we dealt with, and if the source was available. In Stage 2, we regarded all available information about the source, i.e., its full text and its meta-data (e.g., which website the source was found on, the authors if they were identifiable). In Stage 3, we evaluated the quality of the source, successively excluding sources that did not provide useful information. This exclusion was mostly focused on deciding whether criteria 1 to 6 below were met. Since the process was highly context-dependent, it is difficult to describe the exact criteria used. Possible exclusion candidates were therefore marked by one researcher and then confirmed (or overruled) by another in order to ensure unbiased results. The inclusion criteria for stages 1 and 2 were as follows:

1. The source provides first-hand information and is not an encyclopedic article (such as Wikipedia)
2. The source needs to be publicly available (i.e., not behind a paywall inaccessible by researchers or only available on request)
3. The source is in written format (e.g., not a video or an audio file)
4. The source is written in English or German
5. The source has not been considered previously (to avoid duplicates)
6. The source was not written by the authors of this paper (to avoid all sources that are directly related to our own research and could therefore be perceived as biased)

The inclusion criteria for stages 2 and 3 were different depending on the research question:

1. (RQ 2a) The source discusses approaches to traceability based on the standards and does not only mention them as motivation.
2. (RQ 2a) The source provides information about traceability concepts in the context of the standards (and does not, e.g., only advertise functionality of a tool [2] or describe the need for traceability).
3. (RQ 2a) The source refers to important standards in the automotive industry.
4. (RQ 2b) The source discusses traceability practices in the automotive industry.
5. (RQ 2b) The paper discusses traceability in a software development process (as compared to, e.g., traceability of the origin of parts used in the manufacturing of a vehicle).

3.3.4. Data Extraction and Classification

As part of the enrichment with meta-data, we identified the provenance and type of the source. Possible values for provenance – describing to which group of people the authors of the source belong – were academia, tool vendor, consultant, user, standardisation body, agency, student, Open Source Community, mixed, and unknown. As a source for this information, we used the stated affiliations, the website the source was found on, or meta-data provided with the source. In case the provenance was not obvious from the source directly, author names were identified and used to search for their affiliations. In case the affiliation of the authors was with a company that fit several of the possible values for provenance (e.g., a tool vendor that also engages in consultancy services), we selected the one that fit the type of the source best.

We identified a large number of different source types. Among the most prominent were whitepaper, presentation, tool documentation, blog entry, job posting, course announcement, tool description, and manual. An overview of the frequency of provenance and source types is given in Appendix A.

There is overlap between the search results for RQ 2a and RQ 2b. Of the 659 sources totally regarded for RQ 2b, 78 were already analysed for RQ 2a. The number of sources in German is relatively low with a total of 33 out of the 659 considered. Of these 33, 17 were considered relevant and are included in the analysis. Overall, 125 and 120 unique sources were considered for RQ 2a and RQ 2b, respectively.

The data collected for both sub-RQs has been analysed together to answer the over-arching RQ 2. The overlap between the found sources and the relative semantic

[2] While we acknowledge that tools play an important part for traceability, pure marketing material does not describe practical applications and uses.

proximity of the search terms makes a joint analysis prudent. The same codes have been used in both cases to identify challenges and solutions as well. Two researchers engaged in this activity. The starting point for coding challenges were the codes identified in the tertiary literature review (cf. Section 3.2). Codes for solutions were emergent and refined in several rounds of discussions between the researchers. Cross-checking was performed and edge cases were reviewed and discussed. Sources excluded as off-topic or containing no challenges or solutions by one researcher were reviewed by a second researcher to avoid accidental exclusion of relevant material.

3.4. Case Study Design

As previously mentioned, the case study followed the guidelines reported in [22]. The aim of the case study was to provide empirical evidence of the challenges and solutions found in the literature, show how these challenges manifest in practice and identify new challenges that were not reported in literature. Furthermore, the study provided context for the challenges and solutions found in both the tertiary review and the multi-vocal review and therefore provided data to answer *RQ 3: Which of the reported traceability challenges in scientific literature and non-scientific literature can be observed in practice in the automotive domain and how have they been solved?*

3.4.1. Case and Subject Selection

The study was conducted in one of the world's largest suppliers of automotive components located in Germany. The company is multi-national which means that development is distributed in various locations. The company develops various types of automotive components ranging from hardware-only components to software-only components to embedded systems which include a software deployed on a certain hardware component. For this study we were interested in traceability during automotive embedded systems development.

Our case study has two units of analysis within the same company: two departments both developing embedded systems at the company. Since our aim was to investigate how traceability challenges manifest in practice, we selected these two departments because they already implement traceability in their projects and develop safety-critical embedded systems for which traceability is a mandatory requirement. The two departments were also interested in improving their traceability practices, thus the topic was relevant and of interest to them. To be able to understand how traceability is implemented throughout the development life cycle, we conducted the study with seven participants in the following roles: two senior experts working on traceability (one from each department), four software system architects (two from each department) and one functional developer who belongs to one of the departments. We selected these roles in order to get a full picture on how development is done from when a requirement is received to when it is implemented and tested. The first role of senior expert is responsible for understanding what traceability needs the department has, surveying feasible solutions, acquiring these solutions and making sure that they are used in the department. The second role, system architect, is responsible for receiving requirements from the customer, breaking them down and assigning them to development teams. This role is also responsible for managing the architecture of the systems that the department is developing. The last role, developer, is responsible for implementing the features and testing them. In one department, the role of developer and tester are split into two separate roles assumed by separate people.

3.4.2. Data collection procedure

We collected data through observing demonstrations and conducting semi-structured interviews. Observations enabled us to understand the development process and how traceability activities are carried out and the semi-structured interviews enabled us to gather comparable data on the challenges. The model describing the scope of our study and interview questions were sent to the participants a week before the study took place. This was to allow them time to prepare for the demonstrations and interviews. For each participant, we started with the participant giving a demonstration on how they implement traceability using the scope model as a guide. This was followed by a semi-structured interview. The interviewer only asked questions which were not answered by the demonstration part. Due to legal issues, the interviews were not recorded but the interviewer took notes. The interviews and observation for each person lasted between 90 minutes to four hours with breaks in between. The longer sessions were with senior experts who explained and demonstrated the traceability process in detail. The interview guide for these interviews is available online[3].

3.4.3. Analysis procedure

The data analysis started immediately after the observations and interviews were completed. This was to ensure that all relevant information was recorded for later analysis since the interviews were not recorded for legal reasons. The interviewer drafted a summary of the sessions and what was learned from the study and presented it to one of the senior experts for confirmation. During this presentation, the interviewer described the development process and outlined the challenges that were learned from the interview. The senior expert could then confirm the findings or correct the findings when things were misinterpreted by the interviewer. The senior expert could also ask questions at any time during the presentation. This exercise led to few changes, indicating that most of the initially collected information was correct. After this, we went through the

[3] `https://tinyurl.com/ycjrqal4`

interview notes and identified the challenges. We used the categories in the interview model as analysis codes and placed each challenge found in the appropriate category.

3.5. Results

In the next sections (Section 4 to 7) we report findings both from the tertiary literature review, the multi-vocal literature review, and the case study separated by the categories we also used to scope our case study (cf. Figure 2). The challenges are summarized in Figure 4. In the figure, we also include the relationships between the challenges we discovered during analysis. The arrows is indicate the dependencies between challenges: if one is present then the other is also likely to be present. We describe each challenge, discuss it and its solutions in the context of both the tertiary literate review and the multi-vocal review and then compare them with the challenges and solutions at the company.

4. Results: Preparation and Planning

This section describes the challenges and solutions that are encountered when companies are preparing to include traceability either in a specific project or the entire company. Such challenges are concerned with the availability and perception of knowledge about traceability by managers, engineers and developers.

4.1. Knowledge of Traceability

We found four challenges related to knowledge about traceability in the literature. All four were also found at the company. Two of the challenges have been solved while two have only been partially solved using work-arounds.

4.1.1. Lack of Knowledge about and Understanding of Traceability

Description: In order to prepare and plan for traceability in a company, both the managers and developers need to have an understanding of what traceability is and its purpose. This understanding also needs to be aligned, meaning that all the people in the company should have a common interpretation of what traceability is. For companies, if the concept of traceability is not clear, then the chances of failure are high.

Challenge and its Solutions in Secondary Literature: This challenge has been reported by nine papers from our tertiary review [NdlVS13, KS09, CHGHH+14, BQ06, Ram98, WP10, RMMF12, EAG06, OAMH07]. In [CHGHH+14], for instance, the authors report that some companies, especially those not working in a safety-critical domain, have no notion of the term traceability. Another issue is that different individuals in the company have a different understanding of the purpose of traceability [RMMF12]. The most common is that managers see it as a mandatory task that needs to be done for certification purposes while developers perceive it as bureaucracy and a waste of time [KS09, WP10]. In some cases where traceability tools are well-established, developers may perceive it as important and useful for tasks such as impact analysis [Ram98]. The literature proposes that in order to achieve a common understanding of traceability among all stakeholders, training is important. Early on, the company should invest some time and effort to train its employees on purposes and practices of traceability. The training should also discuss semantics of traceability links, completeness, traceability link quality, and other topics [WP10].

Challenge and its Solutions in the Automotive Domain: This challenge was reported 13 times in the the multi-vocal review. In most cases, it was reported in the context of stakeholders in the automotive domain not understanding what kind of traceability is required by the different standards such as ISO 26262 and ASPICE. This is because the standards do not define how traceability should be established concretely (e.g., [SG27, L2009]). It is also not clear how traceability should evolve in the course of the project (e.g., [AG339]. Training is also suggested as a solution (e.g., [SO9] and [SG30]). Furthermore it has been suggested that certification bodies should provide some guidelines on how to properly establish traceability (e.g., [SG129, SG186]).

Comparison to Case Company: This challenge exists at the company but has already been solved. Given that the company operates in a safety-critical domain, employees are already aware of the concept of traceability. They base their understanding of traceability on the requirements defined by the safety standard they need to comply to (ASPICE). They even have expert roles whose job is to understand what the standards require, form a strategy on what they need to do to comply, and communicate this to the rest of the company.

4.1.2. Difficult to Define Information Model for Traceability

Description: Traceability links can be of different types depending on their purpose and which artifacts they connect. The link types can differ from domain to domain. Traceability link types are usually defined in what is known as a traceability information model. It can, e.g., take the form of a meta-model, a database schema, or an ontology. Link types can be generic and carry little or no semantics (for instance a link type called "related_to" that allows connecting arbitrary artifacts) or they can be specific and carry meaningful semantics (for instance a link type named "tested_by" that can only connect a requirement and a test in the sense that the requirement is tested by the connected test). Defining traceability links with domain specific semantics is advantageous as it allows for analysis of the links based on the semantics. In order to define the traceability information model, one needs to understand which types are needed and useful in the specific domain, company, or even project. These needs can evolve over time as well. That makes it difficult to reuse existing information models and to settle on an informa-

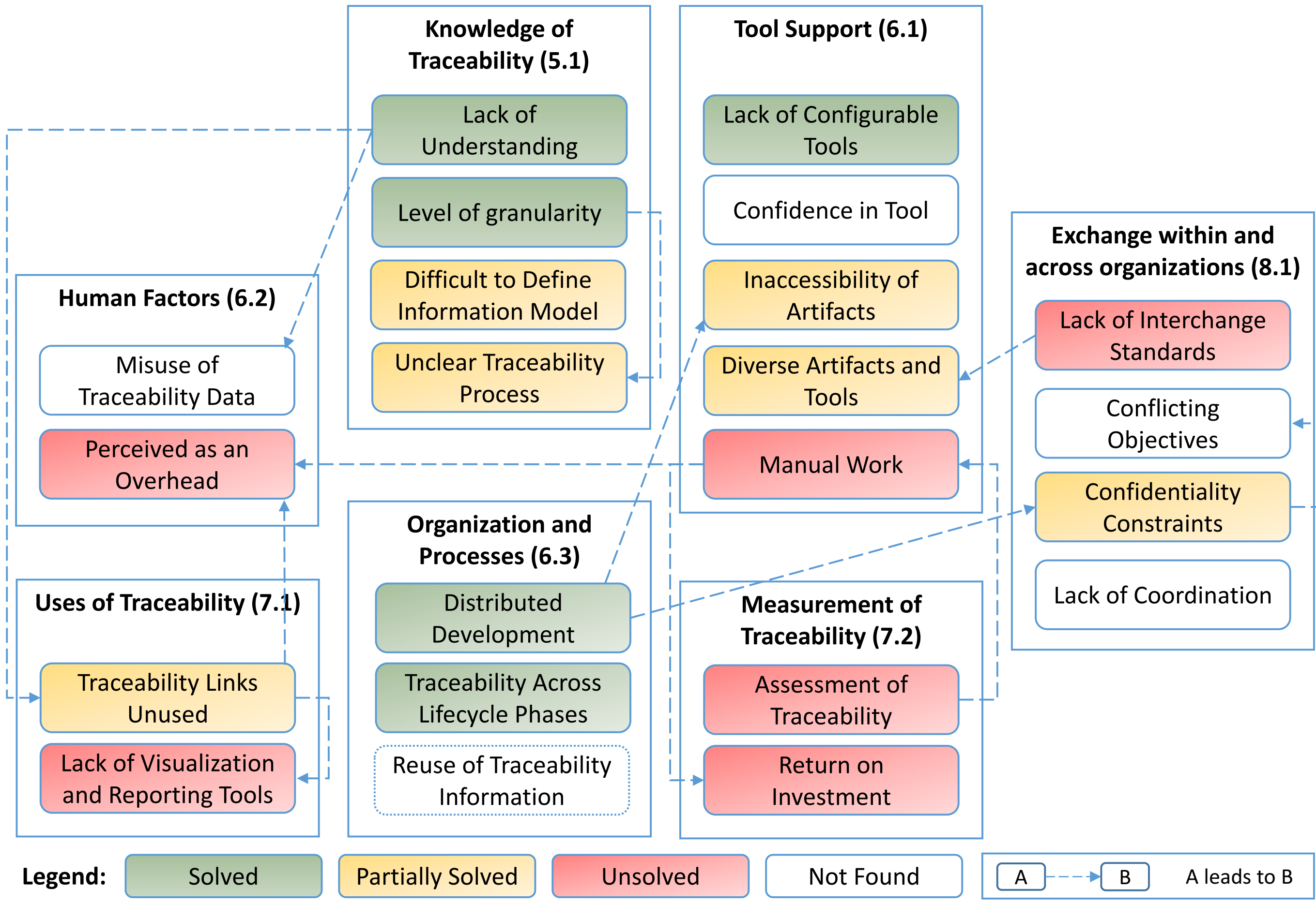

Figure 4: Summary of Traceability Challenges. The solved challenges have a green background, the partially solved challenges have a yellow background and the unsolved challenges have a red background. The challenges that have no background color were only in the literature and not identified in the case study. This means that the data collected was not sufficient to say if these challenges exist in the company. The directed arrows mean that one challenge leads to the presence of another challenge.

tion model that will remain fit-for-purpose over a longer period of time.

Challenge and its Solutions in Literature: This challenge was reported by seven of the papers [RMMF12, ARNRSG06, JZ14, VKP02, EAG06, CHGHH$^+$14, SZ05]. One of the solutions proposed is to define a standard traceability information model [SZ05] after observations in various companies. This model can indeed be used as a starting point for companies to define their traceability metamodel. However, since this is a domain-specific problem, another solution proposed is to document domain-specific guidelines on how to define metamodels [RMMF12]. This can be done through reporting case studies or experience reports.

Challenge and its Solutions in the Automotive Domain: Seven of the sources in the multi-vocal literature report this challenge. Development companies find it difficult to define a traceability information models especially for tracing to non-functional requirements [SG113] and product line artifacts [GIAM2012]. In some cases the companies have no traceability information model at all [AG339]. Solutions to this challenges are to have evolvable semantics [SG129], which means that the traceability information model can be designed in an iterative manner and evolved until it is sufficient. Another solution is to derive the traceability information model from the development process by analysing the process and the involved artifacts [AG293].

Comparison to Case Company: At the company, this challenge exists and it is partially solved. In both departments, the traceability metamodel has already been defined following the ASPICE standard (cf. Figure 2). However, the links are designed specifically to adhere to this standard and it is not clear if these link types assist developers in their development activities. The standard also does not have any guidelines for which links should be created, for development paradigms such as product line development. The two architects interviewed reported that the company has product lines with many variants but they do not know how to include traceability links that take into account variability. This is because the traceability information model defined, does not take into account concepts of variability. Another issue that is not addressed in the standard is how to trace to non-functional requirements such as performance and security. The plan in the company is to use the current metamodel and collect data from its users on what is missing or which links are not working in order to evolve the model.

4.1.3. Level of Granularity

Description: When designing and planning a traceability concept in a company or even a project the level of granularity on which the traceability links should be created on, has to be defined. For instance, a decision must be made if a requirement should be linked to a test file, a test case, or a particular line of code in a test case. This is a challenge, because, if the links are too coarse-grained, they do not provide sufficient detail. If they are too fine-grained, however, their number can become overwhelming and confusing to the end users.

Challenge and its Solutions in Literature: This challenge was reported in three studies [JZ14, MJZCH13, OAMH07]. The solution suggested is that the granularity of the links should be defined explicitly in the traceability information model and the traceability links should be checked regularly to ensure that the links are created with the right level of granularity. This solution however does not suggest which level of granularity a project or company should use.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by 17 sources from the multi-vocal review. The main problem is that at different phases of the system development (requirements engineering, design, implementation etc.) artifacts are defined at different levels of abstraction. This also makes it difficult to determine which abstraction level is appropriate for linking as the mapping from the different development phases is not one to one. For instance, when linking from requirements to architectural components, one has to decide whether to link to high level components of the system or to detailed classes within the components. The multi-vocal review did not point to any solutions to this challenge except that a lot of experience with traceability is important in order to determine the right level of granularity [SG62].

Comparison to Case Company: At the company, this was observed as a solved challenge. The company adopted the level of granularity implied by the V-Model as suggested by the ASPICE standard. The system requirements are derived from customer requirements. The system requirements are then broken down into functional requirements which could be software requirements or hardware requirements. Software requirements are further refined into detailed software requirements. The developer is then assigned a detailed software requirement for implementation. Traceability links are created from customer requirements to software requirements to detailed software requirements. The detailed software requirement is then linked to an implementation file that actually contains the code. The detailed software requirement is also linked to a test.

4.1.4. Unclear Traceability Process

Desciption: Establishing a traceability strategy requires a traceability process (how links are created, used, and maintained) to be put in place. Such a traceability process should be aligned with the software development process that already exists in the company. It is important for the traceability process to refer to work products of the existing development process. This process should also define roles and responsibilities regarding traceability in the company. If such a process does not exist or is vaguely defined, links will be created in an ad hoc manner which results in low link quality.

Challenge and its Solutions in Literature: Ten of the papers reviewed report this as a challenge [ARNRSG06, MJZCH13, Ram98, EAG06, OAMH07, MCCS$^+$12, VKP02, KS09, TGF$^+$12, GF94]. In [MJZCH13], the authors propose that the solution is to create a traceability process based on the traceability metamodel defined at the company. This process should be documented and communicated to all stakeholders early on. Managers should be assigned the role of making sure that this process is followed. In [ARNRSG06], the authors propose putting in place an automated process of creating traceability links by generating skeletons of artifacts from requirements and their traceability links and let these skeletons be filled as development goes on.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by 28 sources from the multi-vocal review. In the automotive domain, standards that need to be followed have an impact on how the traceability process should be defined. Sometimes there is a mismatch between the standards' requirements and the process at the company which makes defining the traceability process difficult [SG129]. Solutions proposed are to use the standards to derive a traceability process [SG99]. This way the company can be sure of its compliance. Another solution is to make sure that the defined traceability process is enforced in order to avoid traceability tasks being performed too late in the development and in an ad hoc manner (e.g., [SG186, SG232]). Moreover, having tool support such as an integrated tool platform where all development activities are done or a structured way of defining artifacts also helps to solve this challenge [AG190].

Comparison to Case Company: At the company, this challenge exists and has been partially solved. A traceability process already exists and although it is a completely manual process, the developers and architects are aware of which links need to be created based on the breakdown of the requirements as discussed previously. In one department, the requirements are defined as use cases and therefore traceability links are created from use cases to design, implementation and tests. In the other department, the requirements are defined as user stories and therefore the links are created from low-level user stories to design, implementation, and test. This challenge is partially solved as there are currently no roles that can check if the process for creating traceability links was followed. Sometimes during review meetings, flaws of traceability links can be detected and fixed.

5. Results: Creation and Maintenance

This section reports on challenges that are associated with the activities of creating and updating the traceability links. The challenges are divided into three categories: tool support, human factors, and organization & processes.

5.1. Tool Support

We found five major challenges in the literature which were reported in this category. Four of these challenges were also found at the case company. On further analysis only two of these challenges have been solved, one has a workaround solution, while two of them still remain unsolved.

5.1.1. Lack of Configurable Tools

Description: Traceability needs can greatly differ from company to company and even from project to project. Therefore, providing a tool that can only be used in a specific context is a limiting factor. It is crucial for tools to allow for customization in terms of link types, supported artifact formats, reporting, selection of relevant information, etc.

Challenge and its Solutions in Literature: This challenge was reported by six studies in our review [KS09, WP10, PLD14, RMMF12, LIA10, CHGHH+14]. The solution described is urging developers of traceability tools to take into account how flexible the tool should be. For instance traceability tools should be flexible in a sense that they allow definition of custom links, allow linking to arbitrary artifacts, be able to define which reports should be created from the links and so on. The more flexible the tool, the better companies can tailor it to fit their project needs.

Challenge and its Solutions in the Automotive Domain: From the automotive literature, this challenge was reported by 13 sources. Most of these sources report that tools do not support the definition of custom traceability links with rich semantics. The solution suggested for this challenge is similar to the ones from the tertiary review. Traceability tool vendors need to design flexible traceability tools that are highly customizable (e.g., [AG15]).

Comparison to Case Company: This is one of the challenges that the company has solved. For requirements management, they have adopted DOORS[4], a tool that is flexible and allows for definition of custom traceability links. Out of the box, the tool allows defining custom link types between requirements. Other artifacts that are stored outside the tool can be linked through OSLC[5] (Open Services for Lifeycle Collaboration) which is a standard for sharing artifacts across tools. For artifacts that do not have OSLC representations, special attributes in the requirements can be defined to store IDs or names of artifacts that are outside the tool. While OSLC enables creating links to artifacts in external tools, maintaining consistency of these external links is a challenge as when artifacts evolve in their tools, the changes are not propagated to other tools for the links to be updated accordingly.

5.1.2. Confidence in Tool

Description: Development companies need to have confidence in the traceability tools that they acquire. One way to establish this confidence is to use tools that have been certified for specific standards. Such a certification provides evidence that the tool works as expected and does not, e.g., introduce errors in the safety analysis that could lead to an unsafe product. It is also important to make sure that the tool is scalable since large and complex systems with a large number of traceability links are common in such domains.

Challenge and its Solutions in Literature: In the tertiary review, three of the sources report this as a challenge [CHGHH+14], [PLD14], [MCCS+12]. Two aspects have been discussed: 1) companies have problems finding tools that will enable them to be adhere to the necessary safety standards [MCCS+12]; 2) companies have no confidence in the scalability of the tools they acquire as they have not been used in large-scale development [CHGHH+14], [PLD14]. There are no concrete solutions suggested for this except that tool vendors should design flexible and scalable tools [RMMF12].

Challenge and its Solutions in the Automotive Domain: Only four of the multi-vocal literature report this challenge. In the automotive domain where requirements can be up to 2000 pages, it is unclear whether existing tools will scale to this level ([BSH2016]). Furthermore companies have to be sure that the tools they acquire will support them in being compatible to the different safety standards. To address the confidence challenge in terms of adherence to safety standards, tool vendors now provide solutions that are certified for these respective standards already (e.g., Polarion [SG40] and Jama [AG324] are both ISO 26262 certified).

Comparison to Case Company: This challenge was not reported at the case company.

5.1.3. Inaccessibility of Artifacts

Description: When creating or updating a traceability link, it is crucial to have access to the artifacts that

[4] http://www.ibm.com/software/products/en/ratidoor

[5] http://open-services.net

need to be connected by the traceability link. In a situation where a project contains a large number of artifacts, tool support is needed to assist in locating the different artifacts. It is also important for traceability information to be accessible by different tools.

Challenge and its Solutions in Literature: Only two of the reviewed papers mentioned this challenge [GF94, KBFS12]. The solutions proposed is that the company, through tools, should ensure that users have all the necessary information and proper access to the artifacts needed to create traceability links. Tools should provide features such as search by ID or search by keywords, to make it easy for the users to find the artifacts they need.

Challenge and its Solutions in the Automotive Domain: Only 7 of the multi-vocal sources report this challenge. The proposed solution is to collect all relevant development information in a centralized data storage.

Comparison to Case Company: For the case company, this challenge is partially solved as the tools used have the ability to search for and locate specific artifacts in an easy way. For traceability links involving artifacts stored in different tools the user still needs to copy the ID manually from one tool to another. While users have access to the artifacts needed due to the presence of centralized storage with appropriate access rights, it is still not possible to access traceability information stored in the requirements management tool (in this case DOORS) directly from other tools.

5.1.4. Diverse Artifacts and Tools

Description: In the software development life cycle different tools are used for the different development activities such as requirements engineering, system design and so on. This means that artifacts are of different formats. Furthermore the artifacts specified in the different tools can contain redundant information which leads to inconsistencies when the system evolves as only some of the artifacts are updated. Development artifacts, especially requirements, can also be specified in various languages. Most traceability tools either do not support linking to artifacts located outside the tool or only support linking to specific tools or specific formats.

Challenge and its Solutions in Literature: Ten of the reviewed studies report this challenge [KS09, VKP02, SZ05, WP10, Ram98, RMMF12, ARNRSG06, GG07, GF94, PLD14] From the studies, there are two different solutions for this challenge. The first option is to use an integrated tool platform that supports all the development activities. A user can interact with heterogeneous artifacts in such an environment using the same user interface and functionality. It includes traceability functionality and the ability to create traceability links between these heterogeneous artifacts. The second solution is tool integration where all existing tools are integrated so that it is possible to exchange information about the heterogeneous artifacts and create traceability links between them. This is however not a trivial task and requires a considerable effort, especially if there are many tools that need to be integrated [27].

Challenge and its Solutions in the Automotive Domain: This is the most reported challenge by the multi-vocal literature. It has been reported by 74 sources, where two of them report that in German automotive companies, some requirements are in English while others are written in German which further complicates traceability [L2009, LO2010]. Two of the solutions proposed are similar to what was proposed in the tertiary review. An additional solution is to define all the artifacts in a structured way so that they can be easily traced. This can be done for instance by specifying artifacts as formal models (e.g., [KKF2014]), tagging artifacts with traceable tags (e.g., [LSS2014]), by enforcing naming conventions (e.g., [SG13]), or by using an integrated modelling language. In this case, homogeneous artifacts are created in one specific modelling language. The model elements can be linked to each other through constructs of that modelling language. Several tools can interact with the artifacts (e.g., [SB2016, SG101].

Comparison to Case Company: In the case company, a total of eight tools are used for different development activities. Tool integration is a technically challenging task. Therefore, the company currently uses implicit links to link to artifacts in different tools which are created by copying IDs from one tool to another. This is not only time consuming, but also error prone and does not allow for any analysis to be done on the links. To overcome this problem, the company is planning to acquire an integrated tool platform that will be able to store all of their artifacts and thus make them accessible for creating traceability links. The main drawback of this solution as reported by one of the architects is that it is hard to find a holistic tool that fully supports all the activities in the development life cycle. Currently, there are no holistic tools supporting activities like simulations which means that even with the holistic tool in place, other tools will still be used. Therefore this challenge is partially solved as linking to tools outside the holistic tool requires implementation of special plugins, which is costly in terms of time and might require rework as the involved tools evolve.

5.1.5. Manual Link Creation and Maintenance

Description: The task of creating traceability links is time consuming when it is done manually. This is exacerbated when there is a large number of artifacts involved. Moreover, traceability links become outdated when the artifacts they connect evolve. This means that they need to be updated in order to remain correct. Manually updating them is time consuming and error prone.

Challenge and its Solutions in Literature: This is one of the most frequently reported challenges in the tertiary literature review. It has been reported by 14 out of 24 papers [NdlVS13, DLFO08, WP10, KS09, TGF+12, SZ05, JZ14, CHGHH+14, RMMF12, ARNRSG06, GG07, PLD14, GF94, MCCS+12]. To overcome this challenge,

Table 2: Challenges and solutions for traceability in the automotive domain

Challenge	TR	MLR	Found at Company	Challenge Solved?	Solutions
Knowledge of Traceability					
Lack of knowledge about and understanding of traceability	11	13	Yes	Yes	Training, Updated guidelines from certification bodies
Difficult to define information model	7	7	Yes	Partially	Defined traceability information model, Updated guidelines from certification bodies
Level of granularity	3	17	Yes	Yes	Defined traceability information model
Unclear traceability process	10	28	Yes	Partially	Defined traceability process, Defined traceability information model, Structured information, Integrated tool platform, Tool integration
Tools					
Lack of Configurable Tools	6	13	Yes	Yes	Flexible tools
Confidence in Tools	3	4	No		Certified Tool Suite
Inaccessibility of Artifacts	2	7	Yes	Partially	Centralized data storage, De-centralized data storage, Flexible tools
Diverse Artifacts and Tools	9	74	Yes	Partially	Integrated tool platform, Tool integration, Integrated modelling language, Structured information
Manual work	14	50	Yes	No	Automation, Just enough traceability, Integrated tool platform, Integrated modelling language
Human Factors					
Misuse of Traceability data	3	1	No		Training
Perceived as an overhead	5	15	Yes	No	Automation, Report generation tools, Just enough traceability
Organization and Process					
Distributed software development	2	11	Yes	Yes	Centralized data storage, De-centralized data storage
Traceability Across Lifecycle Phases	1	35	No		Integrated tool platform, Defined traceability process, Automation, Integrated modeling language
Reuse of Traceability Information	0	6	No		
Uses of Traceability					
Trace links are almost never consulted or used	4	9	Yes	Partially	Report generation tools, Just enough traceability
Lack of proper visualization tools	6	12	Yes	No	Report generation tools
Measurement of Traceability					
Assessing the traceability maintained	5	8	Yes	No	Automation, Defined traceability process, Defined traceability meta-model, Structured data
Return on Investment (ROI).	8	13	Yes	No	Cost-benefit models, Just enough traceability, Automation
Exchange of Traceability Information					
Lack of Coordination in traceability activities	3	23			Collaboration tools, Defined traceability process
Lack of interchange standards	4	8	Yes	No	Common standard
Conflicting objectives	1	1	No		Defined traceability process
Confidentiality Constraints	2	6	Yes	Partially	

the literature proposes the use of automated techniques to generate and update the traceability links. Examples of these techniques are: machine learning [28], information retrieval [TGF+12], event-based techniques [WP10] and model-driven techniques [GG07]. Most of the studies reporting these approaches have been on a theoretical level with small examples and using students as test subjects. For instance the literature review conducted by Borg et al. on information retrieval approaches for recovering traceability links shows that out of 34 publications studied, only one had an industrial evaluation [29]. Additionally, for automated techniques to work, implicit links have to be present so that the algorithms can use them to generate explicit links. In many cases, these implicit links do not exist due to lack of a uniform structure (e.g., naming schemes, meta-data) in the different artifacts.

Challenge and its Solutions in the Automotive Domain: This is also one of the most reported challenges in the multi-vocal literature (reported by 50 sources). Just like in the tertiary review, automation has been suggested as a solution for this. An additional constraint for using information retrieval techniques is that in many German automotive companies, the requirements are written both in English and in German which makes information retrieval difficult. Further solutions are having "just enough traceability" (e.g., [AG56, AG220]), meaning that only links that are needed should be created and maintained. Another solution is to use an integrated tool platform or an integrated modelling language. If all artifacts are accessible from the same tool, then the work of locating artifacts when creating links is reduced (e.g.,[C2014]). An integrated tool also makes it easier to track changes.

Comparison to Case Company: Interestingly, none of these solutions was viable for the company. In general, machine learning, information retrieval and event-based techniques have a low precision and therefore the chance that false traceability links are generated is high. Given that the company produces safety-critical systems and the traceability links are also used for the certification process, false links are not tolerable. Model-driven techniques, on the other hand, require that all the artifacts being linked to and from are represented as models which is not the case for the company, where only some of the artifacts are models.

5.2. Human Factors

In this category we found two challenges that have been reported in the studied literature. Only one of these challenges was found at the case company.

5.2.1. Misuse of Traceability Data

Description: This challenge refers to the fact that in some situations, people responsible for creating and maintaining the traceability links have a fear that this data may be used against them, e.g., during performance appraisals. This happens especially when developers need to create links from artifacts they are responsible for, e.g., bugs reported by users.

Challenge and its Solutions in Literature: This challenge has been reported by three of our reviewed literature [KS09, RMMF12, Ram98]. The authors describe that employees have a fear that traceability data can be used against them and threaten their job security. This is an inappropriate use of traceability data as the data is supposed to be used for quality assurance of the system rather than used for judging employees' performance. The studies propose that both management and employees need to be educated on what traceability is and what the potential benefits are.

Challenge and its Solutions in the Automotive Domain: Only one of the sources in the multi-vocal review reported this challenge. According to [AG139], engineers fear that if they document everything they might become redundant and become replaceable. However, this source does not report any solution to the challenge.

Comparison to Case Company: At the case company, this was not part of the challenges that we identified. However, the company has a system that already logs user activities with respect to creating and modifying development artifacts. If there is a problem in the system it is easy to identify who was working on the artifact and contact them about the problem. This data is not used for performance appraisals. This indicates that the development environment is already very transparent thus employees do not fear the misuse of traceability links.

5.2.2. Perceived as an Overhead

Description: In situations where traceability links are created manually, developers usually perceive this as an extra activity that they need to do or view it as a task that interrupts their workflow. Furthermore, this is a problem since the creators of the links are often not the ones using them. Developers therefore become demotivated and assign a low priority to this task, which can lead to either wrong or missing links.

Challenge and its Solutions in Literature: Five of our reviewed studies report this challenge [GF94, WP10, Ram98, CHGHH+14, MJZCH13]. Proposed solutions for this problem are to ensure that the traceability links created provide immediate benefit to the creators and also to automate the tasks whenever possible. This can be done with tools that enable quick navigation from one artifact to another or visualization techniques that give users an overview of the connection between different artifacts.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by 15 of the sources in the multi-vocal literature. The main problem is that the creators of links are not the ones benefiting and therefore find the task demotivating (e.g., [AG293, KBFS2012, AO5]). Suggested solutions are similar to those proposed in the tertiary review.

Comparison to Case Company: At the case company this is a challenge, due to the diversity of tools and the fact that implicit links are created between artifacts in different tools. It is hard for developers to get an overview of the traces. Across tools they still have to find artifacts by searching for ID and thus do not see the immediate benefits of traceability. All of the interviewees pointed out that being able to navigate easily using the traceability links and having graphical representations of how everything is connected would be a feature that would encourage them to create more correct and complete traceability links. Allowing for easy navigation across tools requires integrating the tools which is also not a trivial task as previously discussed.

5.3. Organization and Processes

In this category, we found three challenges, two of which have been solved at the case company and one which was not reported at the case company.

5.3.1. Complexity Added by Distributed Software Development

Description: In large organizations, it is a common phenomenon that development activities are carried out at multiple sites. This adds complexity to traceability, especially when the different sites need to share the development artifacts. Unless the infrastructure is set up correctly and the sites have a unified software development process, it can be very hard to create traceability links. For companies distributed in various countries, different time zones and languages used in the different locations also make traceability establishment difficult.

Challenge and its Solutions in Literature: This challenges has been reported by two of the reviewed papers. These papers propose a centralized repository for storage of all development artifacts [GF94, RMMF12]. This way, the location of the developers will not matter as everything is centrally stored and shared. Such a repository also needs to be guarded by an access control system to make sure that the right people have access to the artifacts they need.

Challenge and its Solutions in the Automotive Domain: Eleven of the multi-vocal sources report this challenge. The solution proposed is again to use a centralized data storage where all artifacts are stored and therefore accessible by the staff in different locations (e.g., [SG258, SO10]). Another solution is for the company to put in place means of communication and collaboration between the teams in the distributed locations [AG56]. This can be done by using tools that provide collaborative features such as chats and comments on artifact level.

Comparison to Case Company: The company has solved this challenge by having centralized repositories where the artifacts can be stored and different developers are given access rights accordingly. This is in line with what the scientific literature proposes.

5.3.2. Traceability Across Lifecycle Phases

Description: Traceability needs to be established between artifacts that are produced at different stages in the development lifecycle. In principle this is defined in the traceability process (cf. 4.1.4). However, even if such a process is in place, there is still a gap between these lifecycle phases, mainly because they are performed in isolation with different teams and people. It is also common that there is no direct mapping between the artifacts produced in the different phases.

Challenge and its Solutions in Literature: Only one of the sources in the tertiary review reports this challenge [GG07]. One solution has been suggested which is to have a defined traceability process that is supported by tools, e.g., an integrated modeling language that defines which links should be established between models in the different lifecycle phases.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by 35 of the sources in the multi-vocal literature. Several solutions have been reported that can contribute to solving this challenge of both the development process and tools. First of all having an integrated tool platform or an integrated modeling language that integrates all the phases in the development lifecycle ensures that the different artifacts from the different phases are accessible (e.g., [AG318, BLHP+2013]). When an integrated tool platform is not possible, integration of the different tools is suggested via technologies like OSLC to ensure that the different phases are connected tool-wise (e,g., [AG203]). Furthermore a well-defined traceability process and a traceability information model should be put in place and enforced by the development companies (e.g., [SG232, JHHK+2010]). People performing the different activities in the different phases should be aware of their roles and responsibilities when it comes to traceability. Lastly automation can help solve this challenge by, e.g., using model-driven techniques to generate artifacts or skeletons of artifacts from one development phase to the next (e.g., [BF2010, SG98]).

Comparison to Case Company: In the company, this challenge has been solved. Even though the interviewees reported that there is diversity in the tools used in the different lifecycle phases, the development process is well defined and enforced in the company. For instance code will only be written if there is a low-level (detailed) requirement associated with it. This means that the different phases are connected and hence this is not a challenge.

5.3.3. Reuse of Traceability Information

Description: It has already been discussed that establishing traceability is a manual and time consuming process. It is therefore an advantage if the established links can be reused in similar projects or when parts of the projects are being reused, especially in product line environments. This is currently a challenge as it is not clear how to select relevant information for reuse without introducing links outside of the reuse scope. If, e.g., an architectural component should be reused, selecting which of the traceability links connected to it (and thus, which other artifacts) should also be reused is currently a task that is not supported by tools or guidelines.

Challenge and its Solutions in Literature: None of the sources in the tertiary review reports this as a challenge.

Challenge and its Solutions in the Automotive Domain: This has been reported by six of the sources in the multi-vocal review. In the automotive domain, in most cases, systems are not build from scratch but rather reuse existing artifacts such as requirements and code. Developers and stakeholders therefore would like to make use of traceability information when reusing artifacts. Unfortunately, none of the sources reports solutions or best practices. This shows that this is a topic that needs further research.

Comparison to Case Company: At the company, this challenge was not reported by any of the interviewees and also not observed in the process. Currently, traceability information is not reused.

6. Results: Outcome

In this section, we report on challenges related to the outcome of the traceability process. The section is divided into two subsections which are *Use of Traceability* containing challenges encountered when using traceability links and *Measurement of Traceability* containing challenges associated with measuring the quality and benefit of the traceability links.

6.1. Uses of Traceability

For this category, we found two challenges. One of the challenges has been partially solved and one challenge is unsolved.

6.1.1. Traceability Links are Almost Never Used

Description: Even with the large amount of time and effort invested in establishing traceability, traceability links are not used at all or under-utilized. The main use of traceability is still for certification.This is mainly due to the following: 1) lack of tools that facilitate utilization (for instance, good visualizations); 2) the number of links is too high and therefore unusable; and 3) lack of trust in the quality of the traceability links.

Challenge and its Solutions in Literature: This challenge has been reported by four of the reviewed papers [WP10, MJZCH13, TGF$^+$12, CHGHH$^+$14]. In [WP10], it is reported that traceability links are not used either because the links recorded are not helpful to support development activities or because the tools do not provide an efficient way of using the links. The authors point out the importance of tailoring traceability according to the needs of the users and not just creating traceability links for every artifact. In [MJZCH13], the authors point out common flaws that cause traceability links to be ignored. These flaws are, e.g., redundant traceability paths, missing links and out-dated links.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by nine of the sources in the multi-vocal review. Again the most common use of traceability is for certification purposes. The solution proposed in the multi-vocal literature are similar to the ones proposed in the tertiary review.

Comparison to the Case Company: At the company, the main driver for establishing traceability is the requirement from OEMs to be ASPICE compatible. Therefore the main use of the traceability links is for certification purposes. During the interviews we also found that traceability links are used to track the progress of the project, for instance, to check how many requirements already have test cases. The architects and developers however noted that they would like to utilize the links more but that there is no convenient way to do that at the moment. For instance, it is sometimes necessary to copy IDs from one tool to another to search for the connected artifact. This makes it very hard to get an overview of the system or feature through the traceability links. This challenge is therefore partially solved and would be fully solved if better tools that facilitate usage of traceability links are put in place.

6.1.2. Lack of Proper Visualization and Reporting Tools

Description: When traceability is properly established, it can result in a large number of links, in particular if the project consists of a large number of artifacts. The end users of these links need proper visualization tools in order to understand them and powerful reporting tools to produce overviews and statistics for reviews. This is currently a challenge as traceability links are usually presented in large tables or lists where it is hard to comprehend what they mean and even harder to detect flaws in them.

Challenge and its Solutions in Literature: This challenge was reported by six of the reviewed papers [NdlVS13, OAMH07, PLD14, SJV$^+$12, JZ14, MJZCH13]. In [PLD14], the authors point out that it is very important, especially with automatically generated traceability links, to have meaningful graphical representations so that traceability links can be easily inspected for inconsistent and outdated links. Visualization techniques that will facilitate development activities are proposed in [NdlVS13]. For instance, it is useful to have a visualization that will allow the user to see which requirements are already implemented and tested or which tests do not have corresponding requirements.

Most common visualizations of traceability links are a matrix, graphical notations, and hyperlinks. In the matrix view artifacts are displayed in a table with a mark on the cell where the artifact in the column and that in the row are connected by a traceability link. The graphical view represents the artifacts as nodes and the links as edges in a graph. In the hyperlinks view, traceability links are displayed as hyperlinks from an artifact and can be clicked to navigate to the linked artifacts. The authors in [WP10] propose that a traceability tool should have a combination of the three representation as all have advantages and disadvantages and are used for different purposes. The authors illustrate that a project manager may only need an overview of the project but a developer making a change to the system may find hyperlinks more useful as navigation to and from artifacts is facilitated [WP10].

Challenge and its Solutions in the Automotive Domain: This challenge has been reported by twelve of the sources in the multi-vocal review. The solution suggested is also similar to the one suggested in the tertiary review which is, tool vendors need to develop tools that allow custom reports to be generated from traceability information based on user needs (e.g., [AG31, AG94, SG260]).

Comparison to Case Company: In the case company, this was also reported as a challenge that is not solved. This was mainly noted by the developer and the architects who suggested that the traceability links would be more useful for them if they had better graphical representation. They specifically asked for visualization where one is able to get an overview of the project or a specific feature through the traceability links. Also the traceability links that are created manually, for example by copying an ID of one artifact and adding it in another, are not supported by the visualization available in the requirements management tool used in the company.

6.2. Measurement of Traceability

For this category, we found two challenges, both of them unsolved.

6.2.1. Difficult to Assess the Quality of Traceability Links

Description: When traceability is properly established, it can result in a large number of links. In order to trust and use the traceability links, it must be possible to assess their quality by, for instance, measuring how correct and complete the set of traceability links is. This is a challenge as the most reliable assessment method is still manual checking.

Challenge and its Solutions in Literature: Five of our reviewed papers note this as a challenge [NdlVS13, SJV^{+}12, SZ05, VKP02, CHGHH^{+}14] It is hard to assess if the traceability maintained is of high quality as reported in [MJZCH13], where the authors note that even in safety-critical domains the traceability links submitted for certification contain either missing links or redundant links. In [SZ05], it is reported that especially for generated traceability links, it is a challenge to evaluate their correctness and completeness. One proposed solution is to attach confidence values to the generated link and have a threshold based on the confidence value to determine which links are correct. However, this approach does not guarantee that the links will be complete or correct. Another solution is to use the semantics defined in the traceability metamodel to assess the traceability links. For instance, if the information model defines that every requirement should be linked to a test, then missing links can be detected by checking if all requirements have a link to a test. This however only guarantees finding missing links, completeness and correctness still needs to be checked manually.

Challenge and its Solutions in the Automotive Domain: Eight of the multi-vocal sources report this as a challenge. In the automotive domain, it is unclear to companies how traceability links can be assessed to ensure compliance with safety standards. This is because of the lack of guidelines on assessment and in some cases inconsistent guidelines and conflicting requirements from different standards [SG129]. Several solutions are suggested to tackle this challenge. One of them is using semantics of the defined traceability information model as suggested by the tertiary review. Additionally, the multi-vocal review suggests having structured data that can be checked (e.g., [SG84, SG129]). For instance if high-level requirements and low-level requirements have unique naming schemes, then it can be checked that a high level requirement is indeed linking to a low level requirement. Another solution is to define the assessment strategy of trace links when defining the traceability process (e.g., [SG86, SG129]). Even if the strategy is a manual one (e.g., reviews by developers), if it is well-defined and enforced it can improve the quality of the links.

Comparison at the Case Company: At the case company, this is currently one of the unsolved challenges. For traceability links that are created between artifacts in DOORS, there is a possibility to check for missing links easily since the tool allows identifying requirements with no links. Also, since the tool supports defining custom trace links, it is possible to limit which kinds of artifacts a link can connect. The advantage of this feature is that it prevents the creation of links that are semantically wrong. For links that are created with artifacts that are not in DOORS this kind of check is harder as it requires implementation of extra plugins that can do such checks. Correctness and completeness on the other hand needs to be checked manually. This can be done during review meetings but consumes a lot of time and effort.

6.2.2. Difficult to Measure the Return on Investment

Description: Since the most common way of establishing and maintaining traceability in practice is manually, this is a cost-intensive task that requires the company's investment both in terms of money for the tools and in terms of time. It is therefore important for a company to be able to measure what the return on investment of the established traceability links is. This is a challenge as the cost is significant while the benefits cannot be easily measured.

Challenge and its Solutions in Literature: Eight out of the reviewed papers report that traceability establishment is an expensive process [KS09, TGF^{+}12, GF94, WP10, SZ05, JZ14, RMMF12, ARNRSG06] This is because developers need to spend extra time to create and maintain traceability links. Most managers think that a project that implements traceability is more expensive than one which does not [KS09]. Currently there are no measurements that can provide evidence of these direct benefits of traceability. Research proposes cost-benefit models that can be used to show how much traceability has contributed to activities such as maintenance and understandability [6], but these still need to be validated in practice. This is not a trivial task as such benefits are mostly visible at the end of the project. To minimize the effort spent on traceability creation and maintenance, researchers have proposed having "just enough traceability" where links are created only to artifacts of high value (e.g., high priority requirements) [RMMF12].

Challenge and its Solutions in the Automotive Domain: This challenge has been reported by 13 sources in the multi-vocal review. Automotive companies have difficulties proving that traceability is beneficial especially for cases where full time employees are dedicated to this task [AG178]. The multi-vocal literature does not suggest any cost benefit models but rather suggests that there is a need to have more cost-effective ways of establishing traceability. This can be through automation of traceability tasks where possible and also by creating just enough traceability, only links that are needed should be created and maintained [AG56].

Comparison to Case Company: The results of the case study indicated that this challenge has not been solved. All of the interviewees including the managers confirmed that they think traceability is expensive and they do not have evidence of the value it adds to the projects. The only reason that justifies investing in traceability is because it is a mandated task, they have to do it. Value-Based Traceability is also not a feasible solution for them as *full* traceability is a mandatory requirement for safety-critical applications. It is also hard to maintain an exclusive list of high priority requirements that need traceability as priorities can rapidly change over time.

7. Results: Exchange of Traceability Information

This section reports challenges associated with how traceability information can be interchanged between teams within an organization and between different organization.

7.1. Exchange Within and Across Organizations

In this category we found four challenges from the literature. At the case company, one challenge is partially solved even though there was no proposed solution in literature , one is not solved and two of the challenges were not observed.

7.1.1. Lack of Interchange Standards

Description: To facilitate the sharing and transfer of traceability information from one company to another, there is a need for a common standard. Currently, such a standard does not exist and traceability information is stored in various forms ranging from implicit links established through copying IDs from one artifact to another, to explicit traceability links that utilize formal notations such as models. Some links are also stored together with the artifacts while others are stored in a separate trace model with only references to the connected artifacts. Depending on the tool the formats of the traceability links can also vary substantially. This makes it difficult for traceability to be exchanged and reused in different companies.

Challenge and its Solutions in Literature: Four sources in the tertiary review report this challenge[RMKP13, SJV$^+$12, VKP02, CHGHH$^+$14]. The literature proposes the need for one standard that can be used by companies in order to facilitate this sharing and exchange of traceability information [CHGHH$^+$14].

Challenge and its Solutions in the Automotive Domain: This challenge was reported by eight of the sources in the multi-vocal review. The OEM and supplier relationship in the automotive domain means that artifacts are exchanged between the two companies. Some of these artifacts contain traceability information. If there is no standardized format for the links, then they are inaccessible. It is reported in [SG51] that OEMs sometimes acquire entire subsystems from suppliers but have no way of accessing the traceability information from these subsystems. The solution proposed here is similar to the one proposed by the tertiary review. A common standard of accessing and exchanging traceability information is needed. The multi-vocal review suggests that OSLC can be a common standard for information access, but the question of common semantics is still open.

Comparison to Case Company: This is a challenge that the company faces. For instance, OEMs can send requirements which could have traceability links as well. But if the tools at the company cannot identify these links then that information is lost and has to be re-created from scratch.

7.1.2. Conflicting Objectives

Description: When more than one company is involved in the development of a system, it is important to align organizational objectives of all the companies. This is true also for traceability. If the objectives for traceability in one company contradict the ones in another, there might be a conflict. For instance, if the supplier and OEM created traceability links that are not compatible (in terms of types and granularity), then the links end up being unusable between the organizations because each organization has a different objective. If the OEM has the objective of using the traceability links from its suppliers in an aggregated manner to get an overview of the entire system, this will only work if all of the suppliers create the needed traceability links.

Challenge and its Solutions in Literature: Only one of the reviewed papers [RMKP13] reports this challenge. It proposes that at the beginning of the project, all the stakeholders need to align their objectives, including traceability objectives. It is important to define early on what each stakeholder requires and is expected to deliver in terms of traceability.

Challenge and its Solutions in the Automotive Domain: In the multi-vocal review only one source reports this challenge. The OEM and the supplier may have different objectives that can be conflicting [AG56]. The solution proposed is similar to the one proposed in the tertiary literature review.

Comparison to Case Company: This challenge did not come up in the study at the company. Since the company is a supplier, one of their objectives is to satisfy the OEM. In this case, the demand for traceability actually comes from the OEMs. The OEMs specifically asks the company to be compliant to the ASPICE standard in which traceability is one of the requirements.

7.1.3. Confidentiality Constraints

Description: Establishing traceability links that cross the organizational boundaries is a challenging task due to confidentiality implications. It is difficult for suppliers for example, to create traceability links when some artifacts are not accessible to them since they are confidential due to protected intellectual property from the OEM.

Challenge and its Solutions in Literature: In the reviewed literature, two of the papers [RMKP13, KBFS12] mention this challenge but there are no proposals for how to establish traceability when the artifacts are restricted due to legal reasons.

Challenge and its Solutions in the Automotive Domain: This challenge was reported by 6 sources in the multi-vocal review. Most of the time the suppliers only receive partial requirements which makes traceability harder [CSLT2013]. Only one source suggests a solution [AG205] where the development process, including all artifacts exchanged between the OEM and supplier, should be transparent. This can be hard to implement since OEMs keep some artifacts confidential, e.g., because they contain intellectual property that distinguishes them in the market.

Comparison to Case Company: The company also faces this challenge when some of the artifacts they want to trace to cannot be shared by the OEMs. Currently they do not have a solution for this. For some OEMs, the company shares requirements via web interfaces. The OEMs can then limit which fields are visible to the OEM and which fields are visible to both the supplier and OEM. This is an initiative towards sharing confidential information.

7.1.4. Lack of Coordination in traceability activities

Description: During software development different roles need to coordinate. This becomes more important in system development because various parts of the system are developed by different disciplines, from different companies and have to be integrated in the end. For example the software team needs to coordinate with the hardware team to make sure that their software will work on the hardware. This coordination is also important when it comes to updating the traceability links. Coordination becomes difficult because the different disciplines use different vocabularies, have different objectives, and most of the time the development is isolated. When development is done across companies, the different companies involved may also have different development processes.

Challenge and its Solutions in Literature: This challenge was observed by four of the papers we reviewed [TGF+12, KBFS12, RMMF12, RMKP13]. In [TGF+12], just enough traceability (value-based traceability) is proposed as a means to reduce the amount of links created and hence reduce the time people need to coordinate on traceability link maintenance. In [KBFS12], the authors report that change notification is useful for coordination. When an artifact connected by a traceability link has changed, the person responsible for the it should be notified in order to decide how the link should evolve.

Challenge and its Solutions in the Automotive Domain: 23 of the sources in the multi-vocal review report this challenge. Systems development involves various disciplines and establishing traceability between artifacts from the different disciplines is difficult if the disciplines do not collaborate. In the automotive domain this challenge becomes more complex due to the OEM and supplier relation where parts of the tracing need to be done at the OEM and parts need to be done by the supplier [L2009]. There is currently no defined process on how to do this. Two solutions have been suggested. One is to have tools that support the different disciplines with collaboration features such as chats, forums and notifications. This can be part of an integrated tool platform. Second is having a defined process on how the teams should collaborate, in [SG62], it has been suggested that cross-discipline work assignments should be designed to make the different disciplines collaborate more.

Comparison to Case Company: At the company this was not observed as a challenge. On further analysis this can be due to the fact that the requirements management tool has a feature called "suspect links". It highlights the links that connect artifacts which have changed. The user can thus investigate the change and decide how to update the traceability link and the connected artifacts. When working as a team, the suspect links are also propagated to a developers local workspace when they pull changes from the repository. The developers can navigate to see what has changed in connected artifacts by clicking the suspect links.

8. Discussion

In this section, we discuss our results in relation to the research questions. We will address RQ 1 and RQ 2 that deal with the general traceability challenges and the particular challenges of traceability in the automotive domain in sections 8.1 and 8.2. RQ 3 that addresses challenges that can be observed in practice will be discussed in Section 8.3.

8.1. Differences between the tertiary and the multi-vocal review

While most of the challenges and solutions found in the tertiary review were also found and thus confirmed in the MLR, there are a few differences that stand out. This has partially to do with the different data sources and the different provenance of the information (discussed in Section 8.2) and partially with the fact that the sources in the MLR were more specific to the automotive domain. One challenge has been newly identified from the MLR sources: *Reuse of traceability information.* In addition, *Traceability across lifecycle phases* has only been reported once in the tertiary review, but 34 times in the MLR sources.

The reason why *Reuse of traceability information* was not identified as a challenge from the secondary literature might be due to the focus of the MLR on the automotive domain and the high maturity of product line approaches in this area [30]. Reuse of traceability information is usually described in this context. When a component or a subsystem has to be reused in another product, all attached requirements, design documents, test cases, etc. should also be accessible to the developers of the new products. Since these artifacts are connected via traceability links, this information must also be reused. However, there might be traceability links present to the artifacts of other components that should not be reused. This introduces a challenge in terms of which links to reuse and how to deal with those links that point to targets outside of the reused artifacts. It is not clear which solutions apply to this challenge at this point.

The challenge of *Traceability across lifecycle phases* is one of the most reported challenges in the MLR (cf. Figure 5). Our analysis shows that out of the 34 sources that report this challenge, 16 are written by tool vendors. This might be due to the fact that one of the selling points for tools is the ability to establish traceability across all development phases (cf. Section 2). This also correlates with the fact that the sources that mention this challenge also claim to provide a solution either in terms of an *integrated tool platform* or *tool integration.* However, this seems to solve only one side of the challenge which is how the different tools in the different phases can work together. The other side of this challenge is the process side, which refers to how the people involved in the different phases should create and maintain traceability links. A solution suggested is to have a well defined traceability process even though the specifics of what this process looks like and how it should be established have not discussed.

With regards to solutions, there is again a significant overlap between the solutions proposed in the tertiary review and those in the MLR. However, two thing stand out: 1) While the *Confidence in tool* challenge has no concrete solution from the tertiary review, the MLR proposes to have a *certified tool suite* which has been cleared by a certification authority for use within a process to develop safety-critical systems. Again we think that this solution is used as a marketing point, given that it has been reported by only tool vendors and consultants; 2) On the one hand, the MLR calls for *updated guidelines from certification bodies* so that practitioners can have a clear understanding on what they need to do (in terms of traceability) to be able to comply with the standards. On the other hand, the tertiary review reveals that there is a need for guidelines and best practices from research on how to efficiently establish traceability. This shows that from both the academic and the practitioner side, the task of establishing traceability is still not well understood and requires the collaboration of both practitioners and academia to establish guidelines and best practices.

8.2. Differences by challenge and solution provenance

The provenance of the different challenges and solutions proposed in the regarded sources refers to which stakeholder the source can be traced to. There are marked differences between issues discussed in scientific literature that mostly stems from academics and from teams that are a mix between academics and practitioners and the reports made by tool vendors, consultants, and users of traceability. Unfortunately, the latter category is not well represented in this study with only 27 of 246 sources directly attributable to users. Interestingly, of these 27, 15 are publications in peer-reviewed venues. This indicates that information about state of the practice from the user perspective is available in the scientific literature.

Tool vendors want to push the features of their tools and provide mostly marketing material online. However, they are responding to the needs of their customers, so that the features that traceability tools provide reflect (at least in part) issues that the industry deals with. This can be seen, e.g., in the focus on *report generation* and integration, where the latter is addressed with either an *integrated tool platform* or *tool integration.* Features for *flexible visualisation and report generation* are a response to the challenge that is posed by a lack of such tools. Likewise, the two integration approaches are a response to the challenge of *diverse artifacts and tools.* The fact that these challenges and solutions are reported by sources of all provenances indicates that there is an agreement about the validity of the challenges and the potential solution approaches to them.

The challenges *Misuse of traceability data* and *Conflicting objectives* showed up only once, both times in academic papers. Since these are not reported as challenges by any of the practitioners or are even accepted as widespread in the scientific community indicates that they might not be general problems but have rather been observed on few occasions and might thus be issues of individual companies or even project teams. In terms of *Conflicting objectives*, the automotive industry might also be a special case: OEMs and suppliers in most cases have very long-standing relationships with clear communication channels. It can be expected that the objectives are fairly aligned in such an environment. In addition, ASPICE is indeed a standard to regulate the relationship with the supplier. If a supplier follows the standard, any conflicts between expectations and what is delivered should be minimized.

8.3. Unsolved Challenges at the Case Company

With regards to RQ3 (*Which of the reported traceability challenges in scientific literature and non-scientific literature can be observed in practice in the automotive domain and how have they been solved?*), the findings reported in Section 3.5 show that there is a total of six unsolved challenges. An overview of this is given in Figure 4. We will focus our discussion on the unsolved challenges and why they are so difficult to address since this sheds light on how the special circumstances in the automotive domain influence the applicability of solutions. Table 3 gives a summary of the persistent challenges at the case company, why the solutions from the literature are not applicable and which extensions we propose to solve the issues.

One partially solved challenge, however, deserves some attention: *Diverse artifacts and tools* was the most reported challenge in the MLR. The company has integrated tools where possible so that links can be created to and from artifacts in different tools. For instance, in one team, the requirement tool (DOORS) has been integrated with the design tool (Enterprise Architect). However, this is done only for some tools, to allow traceability to tools that have not been integrated, e.g., the requirements tool and the testing tool, the company has a structured way of naming artifacts uniquely, these unique names are then copied from one tool to another to create traceability links. Even though this is a manual process, it works because there are guidelines in how these naming conventions work and the developers follow these guidelines.

Manual work (*Tools*): Several studies have focused on machine learning [28, 31], information retrieval [32] and rule-based techniques [33] for automating the creation and maintenance of traceability links. However, due to the fact that automated techniques can generate incorrect links, which is in violation to safety standards such as ISO 26262, they have not been adopted in the automotive domain. Furthermore, automation techniques only work if implicit links are already in place. To overcome the problem of incorrect links, researchers have proposed that generated links are manually inspected by humans. However, it has been shown that giving a set of generated links to humans to sort out incorrect links can even decrease quality [34].

Other automation techniques in literature are model-based techniques where traceability links are generated as a by-product of transformations. Model-driven traceability works if all artifacts are models. This is not necessarily the case in the automotive industry. Even if models exist, they are often independent and not connected by transformations. Second, many transformation tools that support the generation of traceability links have their own pre-defined notion of link structure and semantics. This makes it hard to integrate them in traceability tools already used in companies [35].

To practically solve this challenge, traceability tools have to enable the combination of manual, semi-automatic and automatic techniques for creation and maintenance of traceability links. Since each of these approaches has its advantages and disadvantages, they can complement each other. For instance, to make sure the links are correct one can rely on manual creation, but to reduce the effort of maintenance, automatic and semi-automatic techniques can be used. Semi-automatic techniques include sending notifications and warnings to users on traceability issues and suggesting probable solutions on how to fix issues. This kind of solution has been investigated in [36, 37] and the authors show that the solution is promising when properly integrated into the traceability tools.

Lack of interchange standards (*Exchange of Traceability Information*): For requirements, there already is a Requirements Interchange Format (ReqIF)[6], which is being adopted and provided as exports from several requirements management tools. Extending such a standard or creating a similar standard for traceability exchange can resolve this challenge. Several sources from the multivocal review suggest OSLC[7] as an interchange standard for traceability [AG294, SG186, BE2014]. OSLC is an integration technology which enables tools to integrate on the data level, i.e., data from one tool can be made accessible to another tool. With a proper set-up, tools from the OEM and supplier can make artifacts available via OSLC and hence enable creation and use of links across companies. It should be noted that a common standard will not solve the *Diverse artifacts and tools* challenge as data still needs to be shared between companies which can cause inconsistencies as the data evolves. Where not legally constrained, we encourage suppliers and OEMs to share the data repository to avoid such inconsistencies.

Lack of Visualization and Reporting Tools (*Use of Traceability*): At the case company, all interviewees were not satisfied with the visualization provided by their traceability tool. Our analysis shows that this is attributed to the fact that most tools are not well adapted to the requirements of using links in different scenarios. Instead, much of the effort in developing these tools is dedicated to the functionality of creation and maintenance of the links, rather than visualization. To solve this problem, we propose that there is a need to first analyze different use cases

[6] `http://www.omg.org/spec/ReqIF/1.1/`

[7] Open Services for Lifecycle Collaboration, `https://open-services.net`

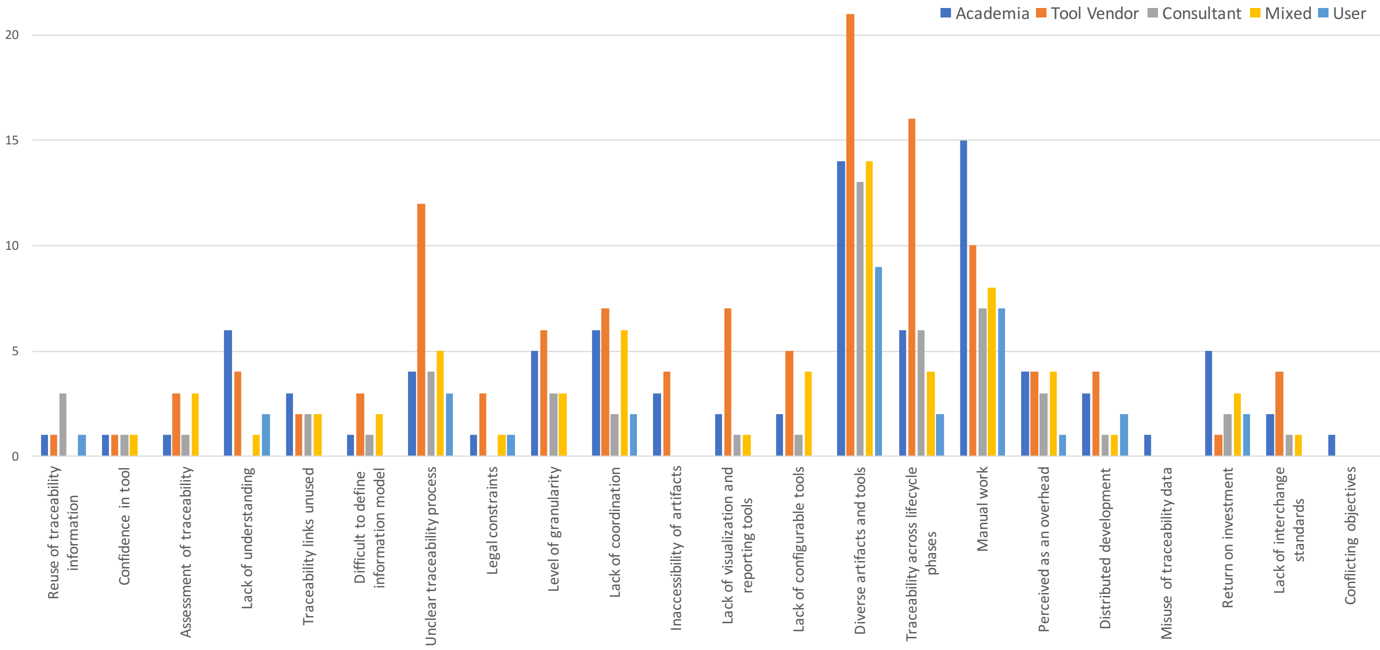

Figure 5: Distribution of challenges by provenance in absolute numbers.

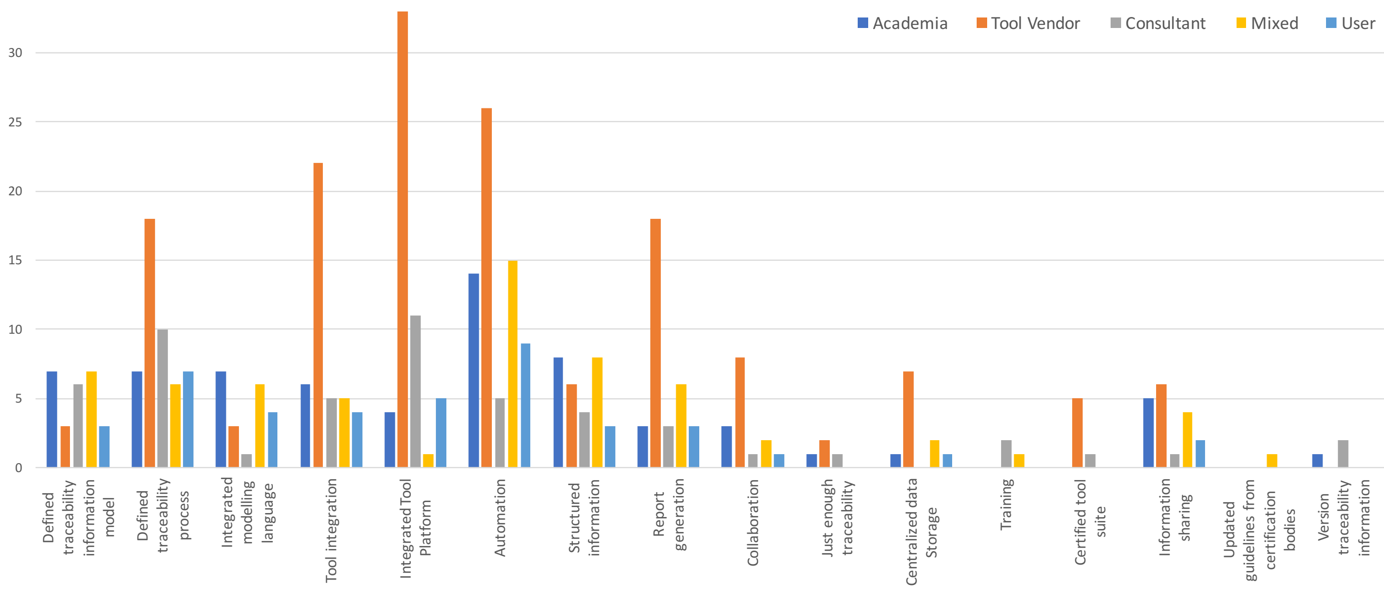

Figure 6: Distribution of solutions by provenance in absolute numbers.

Table 3: Challenges and proposed solutions.

Challenge	Solutions in Literature	Why solutions are not applicable	Proposed Extensions
Manual Work	Machine Learning [28] [GCB2013], Information Retrieval [29] [AG293, AG323], Rule-based [33] and Model-based techniques [41] [SG95]	Machine learning and information retrieval techniques produce incorrect links; not acceptable for safety-critical systems due to the ISO 26262 standard.	Use semi-automatic approaches for maintenance (e.g., to push notifications of artifact changes to responsible users and suggest how links should be updated). Combine manual links with model-based techniques to create links.
Perceived as an overhead	Develop tools that require less effort and produce immediate benefits (e.g, ease of navigation), training on importance of traceability [23, 42, 43] [AG155] .	Large number of heterogeneous artifacts that need to be traced to. Traceability is viewed as only important for certification.	Complement the traceability process with gamification features. Developers can, e.g., be rewarded based on the number of correct links they create and projects can be awarded points/badges based on completeness of traceability links.
Lack of visualization tools	Matrix view, Graphical view and Hyperlinks [44] [SG260, SG25]	Too many traceability links due to large and complex systems.	Provide visualizations suitable for end user needs. Develop tools that enable visualizations to be customised. Users should be able to create different views (graphs, charts, matrices, etc.) based on different data from traceability links.
Assessment of traceability	Use a well-defined traceability information model to facilitate checking for missing links and prevent invalid link creation [45] [AG178], event-based maintenance [37], text-matching	Distributed and isolated development phases	Extend event-based techniques to enable notifications to be sent to artifact owners when links are created involving these artifacts in order to facilitate correctness checks.
Return on investment	Value-based or "just enough" traceability [46] [SG86, AG56]	Links have to be created from all safety-related requirements regardless of their value or priority	Monitor activities supported by traceability to automatically collect evidence on advantages of traceability. Communicate this evidence in the company.
Lack of interchange standards	Create a common traceability standard [6]	A common traceability exchange standard accepted by OEMs and suppliers does not exist.	Adopt OSLC as a way to access data in other tools. Define a common information model.

in which traceability links are used. A study by Bouillon et al. [38] investigated different scenarios in which traceability links are used. Conducting such a study in the automotive domain will lead to usage scenarios that can be used to determine which kind of visualization is appropriate for each use case. When this is clear, it will be possible to add such visualizations to existing tools and support the users when using traceability links. Additionally, tools should provide possibilities for users to create customized reports based on their needs [AG31, AG94, SG260].

Assessment of Traceability (*Measurement of Traceability*): This challenge refers to how the quality of the maintained traceability links can be measured to ensure that the links are both correct and complete. Measuring completeness is tricky since it is difficult to define what completeness means. For instance, while it is possible to check with tools that every requirement has a link to a test, it is not possible to determine that the tests actually cover all aspects of the requirement. Tools are able to flag requirements with no links to test cases, but it is still up to the developers to determine whether the linked tests provide sufficient coverage for the requirement.

Correctness is also hard to assess with tools. For instance a requirement can indeed be linked to a test but the decision if the test is a correct test for the requirement must be made in a time-consuming, manual process. Text-matching [39] that yields a similarity score between the connected artifacts is one approach that can be used to reduce the time spent on this task. The links below a certain similarity score threshold can be shown to the user for a manual check. This solution requires naming standards that ensure that there is always a text similarity between two connected artifacts. Another solution approach is to notify the owners of the linked artifacts when links are created. They can then raise their concern if they think the link is incorrect and discuss the link with the user who created it. This approach is similar to event-based traceability proposed in [40] where the authors suggest notifications to be sent to the owners of connected artifacts when one connected artifact evolves in order to update their artifacts, too.

Return on investment (*Measurement of Traceability*): Most literature on traceability points out benefits such as saving time and effort during impact analysis, tracking progress and improving understandability of the system. However, measuring these benefits in an industrial setting is not trivial because it is hard to isolate the effects of traceability. Also, some traceability benefits manifest only if the project has been progressing and is affected by, e.g., personel churn. Traceability can then save time by helping new developers understand the system and easily navigate to artifacts. Value-based traceability is one solution proposed to reduce the cost of creating traceability links [47, 46]. This means that when planning for traceability, companies need to assure that the links are useful for the project and thus beneficial by assessing why the links are needed and how the benefit will be derived.

In the automotive domain, the main reason for adopting traceability is due to safety standards that demand traceability. This is however not a good motivation as traceability is adopted because people are forced to do it. Being able to quantify the benefits of traceability is one way to show that traceability is indeed useful. For this, we propose monitoring the activities that are supported by traceability links in the company in order to collect data on how useful traceability links are. Additional data can be obtained by conducting surveys with users of traceability and publicizing the results internally in order to promote its adoption in the company even for projects that are not safety-critical and thus controlled by safety standards.

Perceived as an overhead (*Human Factors*): This challenge has two aspects: an organizational and a technical one. The organizational issue is that the people creating and maintaining the traceability links are not the ones using them. A relation to the challenge of understanding traceability thus exists and sufficient training as well as the realization of the immediate benefits of traceability links can help in this regard. The technical aspect is related to the tools that offer little support in terms of visualization, navigation, and analysis. If, based on traceability links, the tools used in the industry can offer features such as easy navigation, visualization, customized reports or even recommendations for artifacts that can be re-used, then the developers creating the links will see their benefits. It should be possible to customize the tools in a way that benefits the creators of the links as well [19]. Another option is complimenting traceability tools with aspects of gamification to make the task of creating and maintaining the traceability links more motivating and engaging. This has been shown to work with other software engineering tasks such as requirements analysis and testing [48].

9. Threats to Validity

In this section we discuss the threats to validity of our study and ways in which we minimized these threats. We use the categories described in [22] but do not discuss internal validity as our study was not not examining a causal relation.

9.1. External Validity

This threat refers to how generalizable the results of the study are. In our case study, we applied data triangulation and interviewed seven employees of three different roles to get data from different sources. However, since we conducted the study in only one company, we cannot generalize the obtained results without further replication of the study which is discussed as future work in Section 11.

With regards to the tertiary review, the most recent publication was published 2014, which reviewed papers up to 2013. There is a chance that papers that propose newer solutions to our identified challenges have been published since then. However, since the multi-vocal literature review covers sources published up to August 2017, we could confirm the data extracted from the secondary sources.

9.2. Construct Validity

To minimize this threat we had to make sure that what we wanted to study (Challenges of establishing traceability) was understood by the participants of the study. To achieve this we first had a meeting with the two experts from the two departments where we explained the intentions of the study. In return, they also explained what their departments do. We also sent the interview guide and scope to the participants one week before the study. As mentioned in Section 3, the interviews we conducted were not recorded due to legal matters but the interviewer took notes. To make sure that we did not misinterpret our findings, we showed our initial analysis to one of the senior experts for confirmation. This is known as member checking [49]. The multi-vocal literature review relies on publicly available sources, it is possible that it does not fully cover the state of the art in the automotive industry. In particular OEMs and Tier-1 suppliers do not make all information regarding their processes publicly available. We have tried to mitigate this thread by being as broad as possible in our search terms and include as many sources from different provenance as possible to construct a picture that is as complete as possible.

9.3. Reliability

To ensure that the results of a study are reliable it is important to make sure that the study can be repeated by other researchers and get the same results. While the settings of the interview cannot be replicated, the artifacts used such as the definition of the scope of the study and the interview guide were well documented and can be used for replication of the study. For the literature review, especially the MLR, even though we have documented our process and have traceability of which source produced which challenges and solutions, repeating the study to obtain 100% similar results is a challenge. This is because for the very short sources (e.g., blogs, presentations, forums), the information given is brief and therefore leaves room for interpretation. To reduce the chances of misinterpretation, two researchers went over the ambiguous challenges and solutions together to code them.

10. Related Work

Regan and colleagues [50], conducted a literature review to identify the barriers of traceability and their solutions from literature. In their work, they propose a framework which consists of the categories of the challenges and their solutions. Their framework is quite similar to the categories of challenges that we have proposed. However, their work does not investigate if these proposed solutions work in practice, which is something that our research does by complementing the literature reviews with an industrial case study.

Further related studies are those by Torkar et al. [51] and Cleland et al [23]. In [51], the authors performed a systematic literature review, with the aim of identifying requirements traceability definitions, tools, practices and challenges. They also complement their work with a case study in two companies. In their results, they give a list of challenges and how they are relevant for the two companies. That study is similar to ours but their literature review only includes papers up to 2007 while ours includes studies of up to 2014. Also in their research the studied companies are not in the automotive domain but in the telecommunication domain and mobile applications domain. In [23], the authors reviewed four recent industrial studies and interviewed eight practitioners on traceability practices. The authors propose several research questions that need to be investigated in order to achieve the seven desired qualities of traceability proposed in [6]. These qualities are that traceability needs to be purposed, cost-effective, configurable, trusted, scalable, portable and valued. These quality attributes correspond to the findings in our study, for instance for traceability to be trusted, there needs to be methods for assessing the quality of links. Also in the study, one of the conclusions is that more collaboration with industrial practitioners and researchers is needed in order to ensure that the solutions from research are actually applicable in practice. Our study is an example of the research proposed here.

A study by Kannenberg & Saiedian [13] reviews the existing literature to investigate why software requirements traceability still remains a challenge. They conclude that manual traceability methods and existing tools are inadequate for the needs of the software development companies, a finding support by our investigation.

11. Conclusion

This paper provides an exhaustive overview of traceability challenges and solutions in the automotive domain and contrasts them with those found in general literature. Our study shows that there is a significant overlap between general challenges and solutions and those found in the automotive domain. It provides evidence that many solutions proposed in the literature are not applicable in the automotive domain due to its specific set of characteristics, such as system complexity, the safety-criticality of the developed systems, and the distributed development split between the OEMs and suppliers.

We used a tertiary literature review to explore general traceability challenges and solutions reported in literature, a multi-vocal literature review to elicit challenges reported in the automotive domain by different provenances such as tool vendors, consultants, academia and users, and a case study to explore how the challenges are experienced in practice.

While the tertiary review revealed challenges and solutions mostly from academia, the MLR was a richer data source (due to the diversity in the provenances). The MLR also gave an indication of which challenges are particularly prominent in the automotive domain. Challenges such as *Diverse artifacts and tools* and *Manual work*, e.g., were reported by all provenances. The same is true for solutions where, e.g., *Integrated tool platform*, *Tool integration* and *Automation* where reported by all provenances. The MLR also showed the difference in challenges that are mainly discussed in academia and those discussed with practitioners such as tool vendors, consultants and users.

The case study validated our findings as most of the challenges were found there as well. In addition, it revealed six unsolved challenges at the company: 1) Manual work of creating and maintaining traceability links, 2) Traceability activities perceived as an overhead, 3) Lack of visualization tools, 4) Manual assessment of links, 5) Hard to measure the return on investment of traceability and 6) Lack of universal standards for exchange of traceability links.

There are proposals for solutions for most of the unsolved challenges. However, for the case we investigated, these solutions were either tried and did not fully solve the problem (e.g., an integrated tool platform to solve the diversity of tools problem) or the solutions could not be applied due to constraints that are specific to the automotive domain such as the requirement to follow safety standards like ISO 26262. This limits, for instance, the applicability of machine learning to generate links for safety-critical applications. Given that our static validation was conducted in one company, this is no indication that these challenges are also unsolved in other automotive companies. Nevertheless, we identify solutions that can be applicable to solve these challenges given the constraints found in the automotive domain. It is therefore still important to investigate how the proposed solutions in literature can be tailored and made applicable to this domain. In cases where tailoring of the solutions will not be enough, new approaches to solve these challenges can be investigated.

For future work, we plan to investigate how solutions proposed in Section 8 will be able to work in practice, by implementing and trying them with practitioners. As part of our research we have developed an open source traceability tool[8] that allows manual creation of links to arbitrary artifacts. In terms of the solutions found in our review, it addresses *Tool integration* and *Report generation*. Our concrete plans are to investigate how to combine automatically created links (for instance from model transformations) with manually created links. We will also investigate how to support users with semi-automatic maintenance of traceability links through notifications and collaborative features such as commenting on links. Furthermore, we will investigate how such a dedicated traceability tool can be integrated into the development process of a company. To contribute to the best practices of traceability, we also plan to work together with our industrial partners, mainly from the automotive domain, to provide different traceability information models for the different systems found in this domains. For instance we will provide information models for traceability when developing product lines and when developing multi-core systems.

Acknowledgements

Funding: This work was supported by Vinnova (grant number 2014-01271) as part of the ITEA^2 project AMALTHEA4Public.

References

[1] M. Broy, Challenges in automotive software engineering, in: Proceedings of the 28th international conference on Software engineering, ACM, 2006, pp. 33–42.

[2] J. Dannenberg, J. Burgard, Car innovation: A comprehensive study on innovation in the automotive industry (2015).

[3] A. Busnelli, Car Software: 100M Lines of Code and Counting, `https://www.linkedin.com/pulse/20140626152045-3625632-car-software-100m-lines-of-code-and-counting`, [Online; accessed 07-10-2016] (2014).

[4] N. Heumesser, F. Houdek, Experiences in managing an automotive requirements engineering process, in: Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International, IEEE, 2004, pp. 322–327.

[5] G. Spanoudakis, A. Zisman, Software traceability: a roadmap, Handbook of Software Engineering and Knowledge Engineering 3 (2005) 395–428.

[6] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, G. Antoniol, The quest for ubiquity: A roadmap for software and systems traceability research, in: Requirements Engineering Conference (RE), 2012 20th IEEE International, IEEE, 2012, pp. 71–80.

[7] C. Lee, L. Guadagno, X. Jia, An agile approach to capturing requirements and traceability, in: Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003), 2003, p. 7.

[8] M. Born, J. Favaro, O. Kath, Application of iso dis 26262 in practice, in: Proceedings of the 1st Workshop on Critical Automotive Applications: Robustness & Safety, CARS '10, ACM, New York, NY, USA, 2010, pp. 3–6. `doi:10.1145/1772643.1772645`.

[9] D. Sexton, A. Priore, J. Botham, Effective functional safety concept generation in the context of iso 26262, SAE International Journal of Passenger Cars - Electronic and Electrical Systems 7 (1) (2014) 95–102. `doi:10.4271/2014-01-0207`.

[10] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, P. Mäder, Traceability fundamentals, in: Software and Systems Traceability, Springer, 2012, pp. 3–22.

[11] S. Nair, J. L. de la Vara, S. Sen, A review of traceability research at the requirements engineering conference re@ 21, in: Requirements Engineering Conference (RE), 2013 21st IEEE International, IEEE, 2013, pp. 222–229.

[12] F. Blaauboer, K. Sikkel, M. N. Aydin, Deciding to adopt requirements traceability in practice, in: J. Krogstie, A. Opdahl, G. Sindre (Eds.), Advanced Information Systems Engineering: 19th International Conference, CAiSE 2007, Trondheim, Norway, June 11-15, 2007. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 294–308. `doi:10.1007/978-3-540-72988-4_21`.

[8] https://eclipse.org/capra

[13] A. Kannenberg, H. Saiedian, Why software requirements traceability remains a challenge, CrossTalk The Journal of Defense Software Engineering 22 (5) (2009) 14–19.

[14] S. Maro, M. Staron, J.-P. Steghöfer, Challenges of establishing traceability in the automotive domain, in: Proceedings of the 9th International Conference on Software Quality, Springer, 2017, pp. 153–172.

[15] International Organization for Standardization, Road vehicles – functional safety, ISO26262:2011.

[16] VDA QMC Working Group 13 / Automotive SIG, Automotive SPICE Process Assessment / Reference Model, Tech. rep., Automotive Special Interest Group (2015).

[17] A. Qusef, G. Bavota, R. Oliveto, A. D. Lucia, D. Binkley, Recovering test-to-code traceability using slicing and textual analysis, Journal of Systems and Software 88 (2014) 147 – 168. doi:http://dx.doi.org/10.1016/j.jss.2013.10.019.

[18] M. A. Javed, U. Zdun, The supportive effect of traceability links in change impact analysis for evolving architectures – two controlled experiments, in: I. Schaefer, I. Stamelos (Eds.), Software Reuse for Dynamic Systems in the Cloud and Beyond: 14th International Conference on Software Reuse, ICSR 2015, Miami, FL, USA, January 4-6, 2015. Proceedings, Springer International Publishing, Cham, 2014, pp. 139–155. doi:10.1007/978-3-319-14130-5_10.

[19] P. Arkley, S. Riddle, Overcoming the traceability benefit problem, in: Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on, IEEE, 2005, pp. 385–389.

[20] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering, sn, 2008, pp. 68–77.

[21] V. Garousi, M. Felderer, M. V. Mäntylä, The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, EASE '16, ACM, New York, NY, USA, 2016, pp. 26:1–26:6. doi:10.1145/2915970.2916008.

[22] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical software engineering 14 (2) (2009) 131–164.

[23] J. Cleland-Huang, O. C. Gotel, J. Huffman Hayes, P. Mäder, A. Zisman, Software traceability: trends and future directions, in: Proceedings of the on Future of Software Engineering, ACM, 2014, pp. 55–69.

[24] P. Rempel, P. Mäder, A quality model for the systematic assessment of requirements traceability, in: 2015 IEEE 23rd International Requirements Engineering Conference (RE), IEEE, 2015, pp. 176–185.

[25] S. F. Königs, G. Beier, A. Figge, R. Stark, Traceability in systems engineering–review of industrial practices, state-of-the-art technologies and new research solutions, Advanced Engineering Informatics 26 (4) (2012) 924–940.

[26] E. Tom, A. Aurum, R. Vidgen, An exploration of technical debt, Journal of Systems and Software 86 (6) (2013) 1498 – 1516. doi:http://dx.doi.org/10.1016/j.jss.2012.12.052.

[27] S. Maro, A. Anjorin, R. Wohlrab, J.-P. Steghöfer, Traceability maintenance: factors and guidelines, in: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ACM, 2016, pp. 414–425.

[28] J. Cleland-Huang, A. Czauderna, M. Gibiec, J. Emenecker, A machine learning approach for tracing regulatory codes to product specific requirements, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, ACM, 2010, pp. 155–164.

[29] M. Borg, P. Runeson, A. Ardö, Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability, Empirical Software Engineering 19 (6) (2014) 1565–1616.

[30] K. Pohl, G. Böckle, F. J. van Der Linden, Software product line engineering: foundations, principles and techniques, Springer Science & Business Media, 2005.

[31] H. U. Asuncion, A. U. Asuncion, R. N. Taylor, Software traceability with topic modeling, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, ACM, 2010, pp. 95–104.

[32] A. D. Lucia, F. Fasano, R. Oliveto, G. Tortora, Recovering traceability links in software artifact management systems using information retrieval methods, ACM Transactions on Software Engineering and Methodology (TOSEM) 16 (4) (2007) 13.

[33] G. Spanoudakis, A. Zisman, E. Pérez-Minana, P. Krause, Rule-based generation of requirements traceability relations, Journal of Systems and Software 72 (2) (2004) 105–127.

[34] D. Cuddeback, A. Dekhtyar, J. H. Hayes, Automated requirements traceability: The study of human analysts, in: Requirements Engineering Conference (RE), 2010 18th IEEE International, IEEE, 2010, pp. 231–240.

[35] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, Y. Shaham-Gafni, Model traceability, IBM Systems Journal 45 (3) (2006) 515–526.

[36] P. Mäder, O. Gotel, I. Philippow, Enabling automated traceability maintenance through the upkeep of traceability relations, in: Model Driven Architecture-Foundations and Applications, Springer, 2009, pp. 174–189.

[37] P. Mäder, O. Gotel, Towards automated traceability maintenance, Journal of Systems and Software 85 (10) (2012) 2205–2227.

[38] E. Bouillon, P. Mäder, I. Philippow, A survey on usage scenarios for requirements traceability in practice, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2013, pp. 158–173.

[39] V. Gaur, A. Soni, A fuzzy traceability vector model for requirements validation, International Journal of Computer Applications in Technology 47 (2-3) (2013) 172–188.

[40] J. Cleland-Huang, C. K. Chang, M. Christensen, Event-based traceability for managing evolutionary change, IEEE Transactions on Software Engineering 29 (9) (2003) 796–810.

[41] I. Galvão, A. Goknil, Survey of traceability approaches in model-driven engineering, Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC (2007) 313–324doi:10.1109/EDOC.2007.4384003.

[42] B. Ramesh, Factors influencing requirements traceability practice, Communications of the ACM 41 (12) (1998) 37–44.

[43] O. C. Gotel, A. C. Finkelstein, An analysis of the requirements traceability problem, in: Requirements Engineering, 1994., Proceedings of the First International Conference on, IEEE, 1994, pp. 94–101.

[44] S. Winkler, J. Pilgrim, A survey of traceability in requirements engineering and model-driven development, Software and Systems Modeling (SoSyM) 9 (4) (2010) 529–565.

[45] P. Mäder, P. L. Jones, Y. Zhang, J. Cleland-Huang, Strategic traceability for safety-critical projects, IEEE software 30 (3) (2013) 58–66.

[46] A. Egyed, P. Grünbacher, M. Heindl, S. Biffl, Value-based requirements traceability: Lessons learned, in: Design requirements engineering: a ten-year perspective, Springer, 2009, pp. 240–257.

[47] J. Cleland-Huang, G. Zemont, W. Lukasik, A heterogeneous solution for improving the return on investment of requirements traceability, in: Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International, IEEE, 2004, pp. 230–239.

[48] O. Pedreira, F. García, N. Brisaboa, M. Piattini, Gamification in software engineering–a systematic mapping, Information and Software Technology 57 (2015) 157–168.

[49] C. B. Seaman, Qualitative methods in empirical studies of software engineering, Software Engineering, IEEE Transactions on 25 (4) (1999) 557–572.

[50] G. Regan, F. McCaffery, K. McDaid, D. Flood, The barriers to traceability and their potential solutions: Towards a reference framework, in: Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on, IEEE, 2012, pp. 319–322.

[51] R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja,

K. Kamran, Requirements traceability: a systematic review and industry case study, International Journal of Software Engineering and Knowledge Engineering 22 (03) (2012) 385–433.

11.1. References from the tertiary literature review

[ARNRSG06] Neta Aizenbud-Reshef, Brian T Nolan, Julia Rubin, and Yael Shaham-Gafni. Model traceability. *IBM Systems Journal*, 45(3):515–526, 2006.

[BQ06] Muhamamd Farhan Bashir and Muhammad Abdul Qadir. Traceability techniques: A critical study. In *2006 IEEE International Multitopic Conference*, pages 265–268. IEEE, 2006.

[CHGHH+14] Jane Cleland-Huang, Orlena CZ Gotel, Jane Huffman Hayes, Patrick Mäder, and Andrea Zisman. Software traceability: trends and future directions. In *Proceedings of the on Future of Software Engineering*, pages 55–69. ACM, 2014.

[DLFO08] Andrea De Lucia, Fausto Fasano, and Rocco Oliveto. Traceability management for impact analysis. In *Frontiers of Software Maintenance, 2008. FoSM 2008.*, pages 21–30. IEEE, 2008.

[EAG06] Angelina Espinoza, Pedro P Alarcon, and Juan Garbajosa. Analyzing and systematizing current traceability schemas. In *Software Engineering Workshop, 2006. SEW'06. 30th Annual IEEE/NASA*, pages 21–32. IEEE, 2006.

[GF94] Orlena CZ Gotel and Anthony CW Finkelstein. An analysis of the requirements traceability problem. In *Requirements Engineering, 1994., Proceedings of the First International Conference on*, pages 94–101. IEEE, 1994.

[GG07] Ismênia Galvão and Arda Goknil. Survey of traceability approaches in model-driven engineering. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, pages 313–324, 2007.

[JZ14] Muhammad Atif Javed and Uwe Zdun. A systematic literature review of traceability approaches between software architecture and source code. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 16. ACM, 2014.

[KBFS12] Simon Frederick Königs, Grischa Beier, Asmus Figge, and Rainer Stark. Traceability in systems engineering–review of industrial practices, state-of-the-art technologies and new research solutions. *Advanced Engineering Informatics*, 26(4):924–940, 2012.

[KS09] Andrew Kannenberg and Hossein Saiedian. Why software requirements traceability remains a challenge. *CrossTalk The Journal of Defense Software Engineering*, 22(5):14–19, 2009.

[LIA10] Grzegorz Loniewski, Emilio Insfran, and Silvia Abrahão. A systematic review of the use of requirements engineering techniques in model-driven development. *Model driven engineering languages and systems*, pages 213–227, 2010.

[MCCS+12] Fergal Mc Caffery, Valentine Casey, MS Sivakumar, Gerry Coleman, Peter Donnelly, and John Burton. Medical device software traceability. In *Software and Systems Traceability*, pages 321–339. Springer, 2012.

[MJZCH13] Patrick Mäder, Paul L Jones, Yi Zhang, and Jane Cleland-Huang. Strategic traceability for safety-critical projects. *IEEE software*, 30(3):58–66, 2013.

[NdlVS13] Saurabh Nair, Jose Luis de la Vara, and Satyaki Sen. A review of traceability research at the requirements engineering conference re@ 21. In *Requirements Engineering Conference (RE), 2013 21st IEEE International*, pages 222–229. IEEE, 2013.

[OAMH07] Rocco Oliveto, Giuliano Antoniol, Andrian Marcus, and Jane Hayes. Software artefact traceability: the never-ending challenge. In *Software Maintenance, 2007. ICSM 2007. IEEE International Conference on*, pages 485–488. IEEE, 2007.

[PLD14] Reza Meimandi Parizi, Sai Peck Lee, and Mohammad Dabbagh. Achievements and challenges in state-of-the-art software traceability between test and code artifacts. *IEEE Transactions on Reliability*, 63(4):913–926, 2014.

[Ram98] Balasubramaniam Ramesh. Factors influencing requirements traceability practice. *Communications of the ACM*, 41(12):37–44, 1998.

[RMKP13] Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Ilka Philippow. Requirements traceability across organizational boundaries-a survey and taxonomy. In *REFSQ*, pages 125–140. Springer, 2013.

[RMMF12] Gilbert Regan, Fergal McCaffery, Kevin McDaid, and Derek Flood. The barriers to traceability and their potential solutions: Towards a reference framework. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, pages 319–322. IEEE, 2012.

[SJV+12] Iván Santiago, Alvaro Jiménez, Juan Manuel Vara, Valeria De Castro, Verónica A Bollati, and Esperanza Marcos. Model-driven engineering as a new landscape for traceability management: A systematic literature review. *Information and Software Technology*, 54(12):1340–1356, 2012.

[SZ05] George Spanoudakis and Andrea Zisman. Software traceability: a roadmap. *Handbook of Software Engineering and Knowledge Engineering*, 3:395–428, 2005.

[TGF+12] Richard Torkar, Tony Gorschek, Robert Feldt, Mikael Svahnberg, Uzair Akbar Raja, and Kashif Kamran. Requirements traceability: a systematic review and industry case study. *International Journal of Software Engineering and Knowledge Engineering*, 22(03):385–433, 2012.

[VKP02] Antje Von Knethen and Barbara Paech. A survey on tracing approaches in practice and research. *Frauenhofer Institut Experimentelles Software Engineering, IESE-Report No*, 95, 2002.

[WP10] Stefan Winkler and Jens Pilgrim. A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling (SoSyM)*, 9(4):529–565, 2010.

11.2. References from the multi-vocal literature review

[AG155] Requirements Manager (m/f) (Driver Assistance) - Job bei Elektrobit ... `http://www.stepstone.de/stellenangebote--Requirements-Manager-m-f-Driver-Assistance-Erlangen-Boeblingen-Braunschweig-Ingolstadt-Muenchen-Ulm-Elektrobit-Automotive-GmbH--4414902-inline.html`.

[AG190] Software Engineering – ITK Engineering. `http://www.itk-engineering.de/en/development-partnership-competencies/software-engineering/`.

[AG15] YAKINDU - Case Study - DENSO - itemis AG. `https://www.itemis.com/en/references/denso/`.

[AG139] Effiziente Erfassung und Pflege von Traceability ... - DepositOnce. `https://depositonce.tu-berlin.de/bitstream/11303/4332/1/figge_asmus.pdf`, 2014.

[AG178] Nachverfolgbarkeit – warum eigentlich? - Systems Engineering Trends. `http://se-trends.de/nachverfolgbarkeit/`, 2017.

[AG203] Atego Trace - Products - Atego. `http://www.atego.com/de/products/atego-trace/`.

[AG205] PLM in der Automobilzulieferindustrie — PRO-CAD. https://www.procad.de/blog/plm-in-der-automobilzulieferindustrie/.

[AG220] Safety and Reliability Analyzer - No Magic. https://www.nomagic.com/images/products/cameo-safety-and-reliability-analyzer/no-magic-csra-brochure-11-2016.pdf.

[AG31] Requirements Software Automotive Industry - Visure Solutions. https://visuresolutions.com/requirements-software-automotive-industry/.

[AG339] Musterbasierte Analyse und Korrektur von Trace ... - UnivIS - FAU. https://univis.fau.de/formbot/dsc_3Danew_2Fresrep_view_26rprojs_3Dtech_2FIMMD_2Flsinfs_1_2Fmuster_26dir_3Dtech_2FIMMD_2Flsinfs_1_26ref_3Dresrep, 2008.

[AG318] www.mbtech-group.com - MBtech safeguards traceability for ... https://www.mbtech-group.com/cz/spolecnost/novinky/news_article/article/mbtech_sichert_nachverfolgbarkeit_in_der_elektronikentwicklung.html?no_cache=1, 2009.

[AG324] Jama Software Announces TUV SUD Certification of Its Solution for ... http://www.marketwired.com/press-release/jama-software-announces-tuv-sud-certification-its-solution-safety-related-development-2144922.htm, 2016.

[AG94] End-to-End Requirements Traceability Report - ReqView. https://www.reqview.com/blog/2016-05-31-news-end-to-end-requirements-traceability-report.html, 2016.

[AO5] Paul Arkley and Steve Riddle. Tailoring Traceability Information to Business Needs. In *RE'06*, 2006.

[BE2014] Andreas Baumgart and Christian Ellen. A Recipe for Tool Interoperability. In *MODELSWARD 2014 - Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development*, 2014.

[BLHP+2013] H. Blom, H. Lönn, F. Hagl, Y. Papadopoulo, M.-O. Reiser, C.-J. Sjösted, D.-J. Chen, F. Tagliabò, S. Torchiaro, S. Tucci, and R.T. Kolagari. EAST-ADL: An architecture description language for automotive software-intensive systems. In *Embedded Computing Systems: Applications, Optimization, and Advanced Design*. 2013.

[AG323] Markus Borg. In vivo evaluation of large-scale IR-based traceability recovery Borg ... In *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, 2011.

[SO9] Elisabeth Brandenburg. USABILITY-OPTIMIZED TRACEABILITY FOR DEVELOPMENT OF EMBEDDED SYSTEM ACCORDING TO ISO 26262. http://optrac.de/wp-content/uploads/2014/06/Usability-Optimized-Traceability-for-Development-of-Embedded-Systems-according-to-ISO-26262.pdf, 2014.

[SG84] M Broy. A Logical Approach to Systems Engineering Artifacts and Traceability. https://pdfs.semanticscholar.org/94ad/d6aa96bee31e8acd59b534f01f3afe66134c.pdf.

[C2014] H. Cho. Traceability-Driven System Development and its Application to Automotive System Development. In *2014 21st Asia-Pacific Software Engineering Conference*, 2014.

[SG258] CollabNet. CollabNet TeamForge Solutions for the Automotive Industry. https://www.collab.net/sites/all/themes/collabnet/_media/pdf/ds/CollabNet_datasheet_TeamForge_Solutions_for_the_Automotive_Industry.pdf, 2013.

[CSLT2013] F. Corbier, M. Soodeen, S. Loembe, and G. Thurston. Creating a systems simulation framework & roadmap. In *SAE Technical Papers*, 2013.

[SG40] Marc Detmers. Polarion Receives ISO 26262 Certification and ... - Polarion Software. https://polarion.plm.automation.siemens.com/hubfs/Docs/Press/Polarion-Software-receives-ISO-26262-IEC-61508-Qualification.pdf, 2012.

[BSH2016] B. Dowdeswell and E. Haemmerle R. Sinh and. TORUS: Tracing Complex Requirements for Large Cyber-Physical Systems. In *2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2016.

[GCB2013] J. Guo, J. Cleland-Huang, and B. Berenbach. Foundations for an expert system in domain-specific traceability. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, 2013.

[GIAM2012] J. González-Huerta, E. Insfran, S. Abrahão, and J.D. McGregor. Non-functional requirements in model-driven software product line engineering. In *4th International Workshop on Nonfunctional System Properties in Domain Specific Modeling Languages, NF-PinDSML 2012*, 2012.

[SG98] Greg Gorman. Systems with IBM IoT Continuous Engineering Solutions - National ... ftp://ftp.ni.com/pub/branches/us/tlf/automotive/ibm_tlf_2015.pdf, 2015.

[JHHK+2010] H. Jost, A. Hahn, S. Häusler, S. Köhler, J. Gačnik, F. Köster, and K. Lemmer. Supporting qualification: Safety standard compliant process planning and monitoring. In *Proceedings - 2010 IEEE Symposium on Product Compliance Engineering, PSES 2010*, 2010.

[SG13] Even-André Karlsson. Traceability in an Agile process - Addalot. http://safety.addalot.se/upload/2016/PDF/1-6%20EAK-Tracebility%20through%20sprints-pub.pdf, 2016.

[KBFS2012] S.F. Königs, G. Beie, A. Figge, and R. Stark. Traceability in Systems Engineering - Review of industrial practices, state-of-the-art technologies and new research solutions. In *Advanced Engineering Informatics*, 2012.

[KKF2014] M. Krammer, M. Karner, and A. Fuchs. System design for enhanced forward-engineering possibilities of safety critical embedded systems. In *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, 2014.

[L2009] J. Leuser. Challenges for semi-automatic trace recovery in the automotive domain. In *2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering*, 2009.

[SG62] Deron Littlejohn. TRW Systems Engineering - Slip Control Systems CAR Breakfast ... http://www.cargroup.org/wp-content/uploads/2017/02/IB_Mar2014_derron_littlejohn_presentation.pdf, 2014.

[AG294] Juan Llorens. Automatic Traceability between models and requirements: The ... http://www.nordic-systems-engineering-tour.com/conference-program-nose-2017/automatic-traceability-between-models-and-requirements-the-trigger-for-systems-engineering-interoperability/, 2017.

[LO2010] J. Leuser and D. Ott. Tackling semi-automatic trace recovery for large specifications. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[LSS2014] T. Lovric, M. Schneider-Scheyer, and S. Sarkic. SysML as backbone for engineering and safety - Practical experience with TRW braking ECU. In *SAE 2014 World Congress and Exhibition*, 2014.

[SG51] Bob Leigh, Joe Wlad, and Brandon Lewis. ISO 26262 Approval of Automotive Software Components - SlideShare. https://www.slideshare.net/RealTimeInnovations/iso-26262-approval-of-automotive-software-components, 2016.

[BF2010] Beine M. and Fleischer D. A model-based reference workflow for the development of safety-related software. In *SAE Technical Papers*, 2010.

[AG293] Neven Pavković, Nenad Bojčetić, Leonard Franić, and Dorian Marjanović. CASE STUDIES TO EXPLORE INDEXING ISSUES IN PRODUCT ... In *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 6: Design Information and Knowledge, Lyngby/Copenhagen, Denmark, 15.-19.08.2011*, 2011.

[SG113] Marie-Agnès Peraldi-Frati and Arnaud Albinet. Requirement traceability in safety critical systems. In *1st Workshop on Critical Automotive applications: Robustness & Safety (CARS 2010)*, 2010.

[AG56] Thomas Reiß. INSTITUT FÜR INFORMATIK Traceability - Aktuelle ... - mediaTUM. `https://mediatum.ub.tum.de/doc/1094508/1094508.pdf`, 2008.

[SG99] Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Jane Cleland-Huang. Assessing the Conformance of Software Traceability to ... - Patrick M. In *ICSE 2014*, 2014.

[SB2016] Harald Sporer and Eugen Brenner. An automotive E/E system domain-specific modelling approach with various tool support. *ACM SIGAPP Applied Computing Review: Volume 16 Issue 1, March 2016*, 2016.

[SG129] Software and Systems Traceability for Safety ... - Schloss Dagstuhl. `http://www.dagstuhl.de/de/programm/kalender/semhp/?semnr=15162`.

[SG232] Functional Design RTL – Implementation R ... - ASICentrum. `http://www.asicentrum.cz/file/downloads/files/advanced_fpga_design_neil_rattray_part2.pdf`.

[SG260] Traceability Matrix and Link Graph — Atlassian Marketplace. `https://marketplace.atlassian.com/plugins/com.kostebekteknoloji.plugins.jira.jira-traceability-matrix`, 2017.

[SO10] Softwareprototyp "Traceability-Demonstrator". `http://optrac.de/wp-content/uploads/2015/09/Traceability_Demonstrator.pdf`, 2015.

[SG25] Seapine Software. Managing ISO 26262 Compliance. `http://downloads.seapine.com/pub/papers/managing-iso-26262-compliance-guide.pdf`.

[SG30] Duncan Seidler and Tom Southworth. ASPICE Made Easy-Case Studies and Lessons Learned - IBM. `https://www.ibm.com/developerworks/community/files/basic/anonymous/api/library/9e66f5de-701e-4994-9291-75646d558240/document/d88a5328-3d2a-4e82-9a3f-e4e5bf9f41be/media/Session22-A-SPICE%20compliance%20made%20easy%20Case%20studies%20and%20lessons%20learned.pdf`, 2013.

[SG27] Dr David Ward. MIRA Safety Analysis - Vector. `https://vector.com/portal/medien/cmc/events/commercial_events/VU_Conference/VUC13/VeCoUK13/VeCoUK13-Day2/Vector_FS_Conf_June13_DDW.pdf`, 2013.

[SG95] Stephan Weißleder. Relation of Model-Based Testing and Safety-Relevant Standards. `http://www.model-based-testing.de/mbtuc11/presentations/Weissleder-FIRST-Standards_and_MBT.pdf`.

[SG86] Svenja Wendler and Franz Graser. Durch Traceability lassen sich unbändige Projekte zähmen. `http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/analyseentwurf/articles/431603/`, 2014.

[SG186] Andreas Willert. Traceability - Willert Software Tools GmbH. `https://www.willert.de/assets/Newsletter/ESER-35-Traceability-in-Theorie-und-Praxis.pdf`.

[SG101] Ralph Weissnegger, Markus Pistauer, Christian Kreiner, Kay Römer, and Christian Steger. A novel design method for automotive safety- critical ... - IDA.LiU.se. In *Proceedings of the 2015 Forum on Specification & Design Languages*, 2015.

Appendix A. Additional data about MLR

This appendix contains tables with additional information about the multi-vocal literature review.

Table A.4: Frequency of different source types in the MLR by provenance

Source Type	Total	Academia	Tool Vendor	Consultant	Mixed	User
Conference Paper	65	23	5	2	23	12
Workshop Paper	11	7	1	0	3	1
Journal Paper	12	3	1	1	6	1
Book Chapter	3	2	0	0	0	1
Peer reviewed	91	35	7	3	32	15
Book	3	2	0	0	0	1
Whitepaper	19	1	10	6	1	1
Presentation	35	6	11	11	3	5
Press release	3	0	2	0	0	1
Blog Entry	7	0	5	2	0	0
Thesis paper	3	2	0	0	0	0
Job posting	3	0	0	2	0	1
News article	6	0	4	2	0	0
Project Deliverable	4	1	0	0	1	2
Forum post	1	0	0	0	0	1
Course announcement	1	0	0	1	0	0
Tool description	43	0	41	2	0	0
Case description	8	0	7	1	0	0
Technical Report	2	1	0	0	1	0
Website	1	0	0	0	0	0
Service description	3	0	1	2	0	0
Magazine Article	8	0	4	3	0	0
Manual	2	0	0	0	0	0
Thesis description	2	1	0	1	0	0
Training Material	1	0	0	0	0	0
Talk abstract	1	0	0	0	0	1
Workshop Proceedings	1	0	0	0	1	0