The Journal of Systems and Software 128 (2017) 25–55

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Automotive software engineering: A systematic mapping study

Alireza Haghighatkhah*, Ahmad Banijamali, Olli-Pekka Pakanen, Markku Oivo, Pasi Kuvaja

Empirical Software Engineering in Software, Systems and Services (M3S), Faculty of Information Technology and Electrical Engineering (ITEE), University of Oulu, Oulu, Finland

ARTICLE INFO

Article history:
Received 5 July 2016
Revised 27 February 2017
Accepted 6 March 2017
Available online 7 March 2017

Keywords:
Literature survey
Systematic mapping study
Automotive software engineering
Automotive systems
Embedded systems
Software-intensive systems

ABSTRACT

The automotive industry is going through a fundamental change by moving from a mechanical to a software-intensive industry in which most innovation and competition rely on software engineering competence. Over the last few decades, the importance of software engineering in the automotive industry has increased significantly and has attracted much attention from both scholars and practitioners. A large body-of-knowledge on automotive software engineering has accumulated in several scientific publications, yet there is no systematic analysis of that knowledge. This systematic mapping study aims to classify and analyze the literature related to automotive software engineering in order to provide a structured body-of-knowledge, identify well-established topics and potential research gaps. The review includes 679 articles from multiple research sub-area, published between 1990 and 2015. The primary studies were analyzed and classified with respect to five different dimensions. Furthermore, potential research gaps and recommendations for future research are presented. Three areas, namely system/software architecture and design, qualification testing, and reuse were the most frequently addressed topics in the literature. There were fewer comparative and validation studies, and the literature lacks practitioner-oriented guidelines. Overall, research activity on automotive software engineering seems to have high industrial relevance but is relatively lower in its scientific rigor.

1. Introduction

Problem statement and motivation. Over the last few decades, software has significantly affected embedded system functionality and innovation in a wide variety of domains, from telecommunications to medical devices and from vehicles to aircraft systems (Liggesmeyer and Trapp, 2009). Many products containing embedded software have been developed for use in several fields, bringing value to society (Feitosa et al., 2010). Intensive mechanical functions are being replaced by software functions that enable innovation, fast delivery through reuse, and potential differentiation of new products. These changes have led to a situation in which software engineering has become a vital discipline in embedded systems and a major player in innovation and competition in the marketplace (Liggesmeyer and Trapp, 2009). Following this trend, the automotive industry is going through a fundamental change by moving from an intensive mechanical to an intensive software industry in which the majority of innovation and competition rely on the competence of software engineering.

Only four decades ago, the automotive industry witnessed the deployment of small software programs that was used to control engines and the ignition system in particular (Broy et al., 2007). The first generation of software functions in the automotive industry were strictly local and they were functionally and technically isolated from other software functions (Broy et al., 2007). Now, software functions are diverse and widespread in vehicle systems, and they range from low-level control software to advanced driver-assistance and infotainment systems. Moreover, they are highly interactive and distributed throughout various electronic control units (ECU), which are connected through several in-vehicle networks. Over 80% of innovations in the automotive industry are now realized by software-intensive systems (Broy et al., 2007). This is evidenced by premium vehicles, which contain about 270 user functions composed of as many as 2500 individual atomic software functions and deployed in more than 70 embedded platforms (Pretschner et al., 2007). Furthermore, in the last four decades, the amount of software in vehicles has evolved from zero to hundreds of millions of lines of code, and this growth is expected to continue in the future (Pretschner et al., 2007; Broy et al., 2007). The reasons for this rapid development are the availability of cheap and powerful hardware resources and the demand for innovation and new functions (Broy et al., 2007). The introduction of software in the automotive industry not only has supported the realization of innovative functions but also has reduced gas consumption and improved performance, comfort, and safety through the electrifi-

* Corresponding author.
E-mail address: alireza.haghighatkhah@oulu.fi (A. Haghighatkhah).

http://dx.doi.org/10.1016/j.jss.2017.03.005

cation of automotive systems (Pretschner et al., 2007). In addition, software is economically relevant because it has a negligible replication cost, allows hardware to be reused in different vehicles, and enables mass-differentiation and customization (Pretschner et al., 2007).

Overall, despite various challenges, software engineering has become an essential discipline in the automotive industry, especially with regard to innovation and competition. The underlying reasons for these challenges are diverse, including the unique characteristics of the automotive domain, the distributed nature of software functions, and the ever-growing demand for non-functional requirements, such as safety, reliability, and performance. To manage this complexity, the automotive industry has developed and adopted several standards, solutions, and platforms, including both process-based and product-based countermeasures (Fabbrini et al., 2008). Thus, today's automotive software-intensive systems are developed based on and in compliance with several standards and models, such as Automotive SPICE, ISO-26262, Motor Industry Software Reliability Association (MISRA), AUTomotive Open System ARchitecture (AUTOSAR), OSEK (open systems and their interfaces for the electronics in motor vehicles), and many more. Even though the automotive industry may generally adopt the results and solutions offered in the body of knowledge accumulated by software engineering in other domains, the unique characteristics, constraints, and requirements of the automotive industry require individual solutions. Research in software engineering has progressed from seeking universal solutions through concepts, models, and technologies to domain-specific solutions that take advantage of domain specificities (Clarke et al., 2008; Bryant et al., 2010). One could argue that the automotive industry is a prominent example of such development through adaptation, design, and exploitation of domain-specific technologies. The initiatives taken in exploring the body of software engineering knowledge to develop software-intensive automotive systems can be identified as resulting in the area of research known as automotive software engineering (ASE). A large body of knowledge and evidence has been developed in recent years in several published studies investigating how software engineering has been applied to the development of automotive software-intensive systems. However, no previous work has provided a panoramic overview and a complete summary of how software engineering technologies have been used to develop automotive software-intensive systems.

Research approach and contribution. In this paper, we aim to investigate the current state of the research pertaining to software engineering in the automotive industry. We conducted a systematic mapping study, which is typically employed to investigate a relatively broad topic and aims to identify, analyze, and structure the goals, methods, and contents of previous primary studies (Petersen et al., 2015). Specifically, in this systematic mapping study we aim to study the peer-reviewed literature in order to characterize and synthesize the available contributions with respect to the research topics, research types, research methods, contribution types, and research gaps from the viewpoints of both scholars and practitioners in the context of automotive software engineering. Our systematic map consists of 679 primary studies in multiple research sub-areas that were identified from an initial set of 4348 papers extracted from different sources. The main contribution of this study is the presentation of a structured body of knowledge about automotive software engineering in addition to the synthesis and analysis of existing contributions. Specifically, our contribution can be classified into three main parts: the identification of the primary studies along with their publication sources; the classification and analysis of existing contributions based on their research focus, research type, research method, and contribution type; and the identification of research trends, open challenges, and promising future directions in the research on ASE.

Organization. The rest of this paper is organized as follows: the related work is presented in Section 2. Section 3 describes the research methodology, including a discussion of the validity and the limitations of this research. Section 4 presents the results of our systematic mapping study. Section 5 discusses the findings and their implications, and concluding remarks are given in Section 6.

2. Related work

The most comprehensive overview of the state of the art in ASE was provided by Pretschner et al. (2007) in 2007. The study presented domain characteristics and their consequences and provided a roadmap for research in this area. From a broad perspective, these research challenges are related to quality and cost, which is reflected in the need to integrate, evolve, and reuse automotive software applications. The study further highlighted model-based development as a possible approach to tackling these challenges. It also presented some particularly relevant research directions to be considered for the improvement of the model-based development of automotive software systems. A few other studies similar in scope and findings, such as Broy (2006), Broy et al. (2007), Broy (2005), Salzmann and Stauner (2004), were found in the literature.

In a similar vein, Clarke et al. (2008) presented important challenges in automotive software engineering and outlined approaches and research directions that could be used to address them. The authors specifically highlighted software product lines, global software development, service-oriented architecture, and mathematical methods applied to software engineering as future research directions for the automotive industry. Several studies, such as Grimm (2003), Gruszczynski (2006), Fabbrini et al. (2008) reported their experiences and highlighted particular challenges and research directions for the field. For instance, Grimm (2003) discussed the challenges of automotive software engineering and highlighted the most important technological core competencies used to overcome those issues. Gruszczynski (2006) presented an overview of software engineering technologies in the automotive industry and highlighted prominent future research directions. Similarly, Fabbrini et al. (2008) presented a survey of the achievements and challenges in software engineering in the European automotive industry and suggested future research directions.

In addition, three studies systematically analyzed the literature on ASE with respect to various sub-research areas. Dersten et al. (2011) conducted a systematic literature review to investigate the effects of the introduction of AUTOSAR from both technical and organizational perspectives. The results showed that although AUTOSAR had various benefits, such as efficient development and short lead times, it also entailed several costs, such as performance risks, learning curves, and the deployment of new development processes in the organization. Moreover, Kasoju et al. (2013) analyzed the automotive testing process by following an evidence-based software engineering process. This research included a case study followed by a domain-specific systematic literature review and value stream mapping to investigate the challenges and improvement proposals with regard to software testing in the automotive industry. The results showed 26 individual challenges and 15 solutions to improve the software testing process in the automotive industry. Similarly, Pernstål et al. (2013) conducted a systematic mapping study on lean approaches to large-scale software development. The main objective of the study was to identify and support the improvement of software process opportunities for large-scale industrial projects at the Volvo Car and Truck Corporation. The results showed that the majority of studies (76%) were non-empirical while only 42% of studies targeted large-scale development. Furthermore, a large number of studies had a narrow focus on a limited number of lean principles and practices.

Overall, several studies have contributed to the body of knowledge on software engineering in the automotive industry. To support practitioners and researchers in ASE, three previous studies (Dersten et al., 2011; Kasoju et al., 2013; Pernstål et al., 2013) provided systematic reviews classifying and synthesizing domain-specific knowledge and evidence with respect to a particular subject area. In addition, several authors (Pretschner et al., 2007; Broy, 2006; Broy et al., 2007; Broy, 2005; Salzmann and Stauner, 2004; Clarke et al., 2008; Grimm, 2003; Gruszczynski, 2006; Fabbrini et al., 2008; Harris, 2013) reported their experiences and discussed software engineering technologies and the challenges that are particular to this domain. However, these studies only partially covered the state-of-the-art literature on automotive software engineering. Furthermore, the literature is not well structured, and it lacks an overview that gives a complete picture and summary of software engineering in the automotive industry.

2.1. Automotive software engineering - characterizing the domain

Pretschner et al. (2007) characterized the automotive domain and provided a roadmap for research in this area. Based on the work of Pretschner et al. (2007), we will highlight the unique characteristics of the automotive domain and their consequences for ASE.

Characteristic 1 - heterogeneous nature of automotive software. Automotive software functions are highly heterogeneous and deployed over different application areas, including the following: 1) infotainment and telematics software; 2) body and comfort software; 3) chassis and safety software; 4) powertrain and transmission software; 5) driver assistance software; and 6) infrastructure software. Each application area has unique characteristics and different requirements and quality attributes, such as soft or hard real-time, strict safety, and availability. From the perspective of integrated systems and software engineering, the development of an automotive software-intensive system required the development of multidisciplinary skills. Moreover, widely accepted approaches and standards are lacking in the current development of automotive applications. The consequences are a heterogeneous multiplicity of methods, processes, and tools, as well as increased complexity.

Characteristic 2 - organization and distribution of labor. Traditionally, automotive development was organized in a highly vertical manner that involved a large chain of suppliers providing different components of the vehicle. This facilitated vehicle development and allowed vehicle manufacturers to optimize the distribution of cost and risk. However, distributed development organizations required the efficient coordination of complex distributed processes, effective communication, clear interfaces, and well-established liabilities. In addition, as the influence of software engineering increased in the automotive domain, the responsibility of vehicle manufacturers has changed from the assembly of parts to system integration. Thus, challenges have arisen with regard to system integration and management of requirements and their volatility during the development process.

Characteristic 3 - distributed nature of automotive software. Automotive software functions are highly interactive. They are distributed among different ECUs and are connected by an in-vehicle network. Furthermore, these software functions are required to interact through several middleware/buses and the multiple real-time and operating systems embedded in the vehicle. This increased complexity leads to a large number of feature interactions, including unintentional or intentional dependencies between individual functions. Because of this complexity, the qualification testing and quality assurance of automotive software systems is highly demanding.

Characteristic 4 - large number of variants and configurations. Because of the desire for differentiation and mass customization, the automotive industry is dealing with a very large number of variants and configurations. These are mainly consequences of business purposes or the development process. The lifecycle of hardware components such as CPUs is much shorter than the long lifetime of vehicles. Therefore, these hardware components or a complete ECU and its respective software might be replaced by newer ones. Over time, this would lead to various versions of each piece of software in a vehicle. The consequence would be challenges with respect to the interoperability and compatibility of automotive software systems. Therefore, variants and configurations must be managed effectively, and long-term maintenance processes must be organized.

Characteristic 5 - predominance of unit-based cost models. The automotive industry has a long tradition of the unit-based cost model, and it operates in a very competitive mass market. Traditionally, the cost per unit produced has played a significant role in the mass production of vehicles. Competition by differentiation requires innovation and a strong brand profile, while competition in terms of price requires constant optimization. To reduce the cost, for instance, automotive engineers focus on different approaches to reduce the required amount of memory and computation power through tailoring and optimization of software. Such intensive optimization typically requires close consideration of hardware characteristics and ties the software application to a particular hardware component. In consequence, the reuse and maintenance of automotive software applications becomes challenging.

Furthermore, given the prevalence of legacy systems, the automotive industry is dealing with several challenges including the interoperability and compatibility of automotive system (Broy, 2006; Fürst, 2010; Fürst et al., 2009). For instance, the carryover and effective integration of legacy components with new systems to decrease overall cost is one factor with which the automotive industry struggles. This was acknowledged in previous studies (Fürst, 2010; Fürst et al., 2009), and it is particularly important in the case of AUTOSAR standardization. However, the carryover of legacy components is challenging in practice, and in several cases, it hinders the innovation possibilities in the automotive industry.

3. Research method

A systematic mapping study was conducted to investigate and classify the literature relevant to the automotive software engineering. The main difference between a systematic literature review and a systematic mapping study is that the systematic literature review aims to “identify best practice with respect to specific procedures, technologies, methods, or tools by aggregating information from comparative studies”, whereas mapping studies target broad areas and focus on “classification and thematic analysis of literature on a software engineering topic” (Kitchenham et al., 2011). Similar to our research, several previous literature reviews aimed to investigate software engineering in a certain context other than automotive. For instance, Glass et al. (2002), Paternoster et al. (2014), Ampatzoglou and Stamelos (2010), and

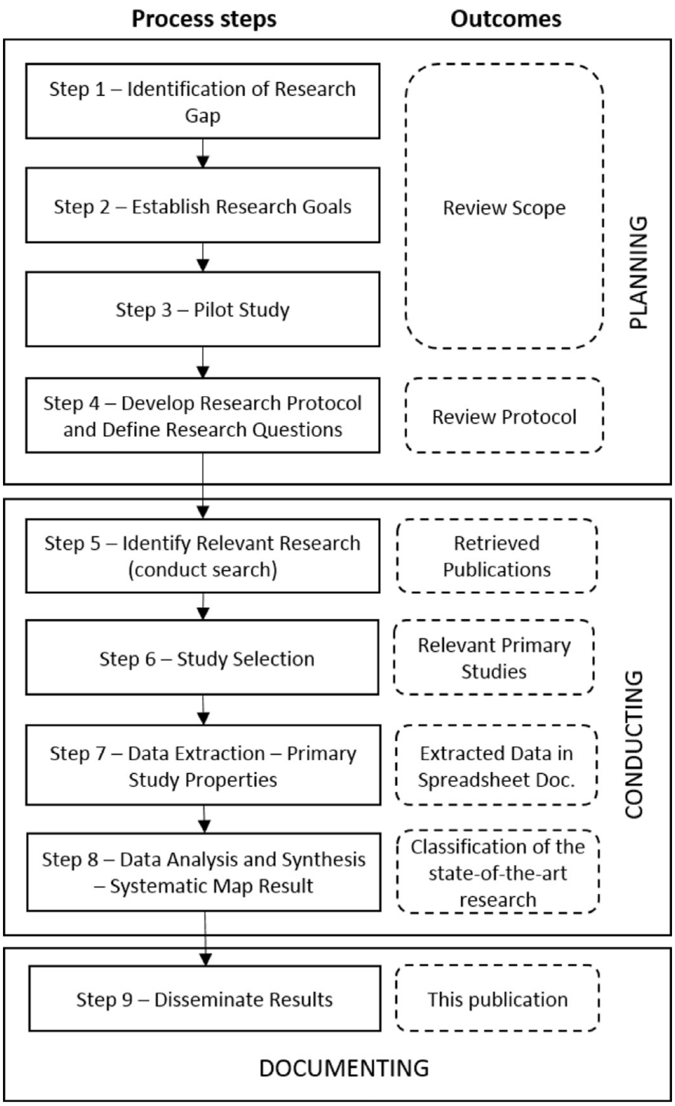

Fig. 1. Systematic mapping research process.

Feitosa et al. (2010) had similar objectives. The scope of our research is similar to these studies because they also aimed to investigate software engineering technologies and challenges in a broad context.

In our research, we followed the guidelines for performing systematic literature reviews proposed by Keele (2007) and adapted these guidelines to a systematic mapping study, as recommended by Petersen et al. (2008); 2015). The research process used in the present study is presented in Fig. 1. The process constitutes various steps, each of which is aimed to produce an outcome. The research process was iterative and included feedback loops between the various steps in order to identify improvement opportunities while we learned about the phenomenon under study. For instance, several pilot search strings were evaluated during the early stages of this study, and the research protocol was updated accordingly based on design decisions. Furthermore, additional research questions and data properties were incorporated during the pilot phase as we learned more about the nature of the primary studies. However, in order to simplify the representation of how the research was conducted in practice, the procedure is presented in sequential fashion and the key design decisions taken into account during the process are emphasized. This section elaborates the design and execution of our systematic mapping study. The study's objective and research questions, search strategy, selection criteria, selection process, data properties and data extraction procedure, and the analysis and interpretation of the data are discussed in that order. Subsequently, threats to the validity of the results and the countermeasures taken during the process, followed by the limitations of the study, are discussed in this section.

3.1. Objective and research questions

The main objective of our study is to investigate the current state of the research pertaining to software engineering in the automotive industry. Specifically, in this research, we aim to investigate the peer-reviewed literature in order to characterize and synthesize available contributions with respect to the investigated research topics, research types, research methods, contribution types, and research gaps from the perspective of both scholars and practitioners in the context of automotive software engineering. The main research question driving this research and reflecting our objective is: *What is the current state of the research pertaining to software engineering in automotive industry?* In order to answer this question, we posed five research questions, each of which addresses a particular aspects of ASE. The research questions and their rationale are as follows:

RQ1. What is the intensity of research activity on ASE? *Rationale*: This research question is designed to identify and analyze the intensity of research activities associated with ASE literature. In this question, intensity is represented by the quantification and analysis of publication trends, authorship information, publication channels, publication titles, and citations in studies relevant to ASE.

RQ2. What are the most frequently investigated research topics? How have they evolved over time? And in what application area have they been investigated? *Rationale*: This research question is designed to identify and analyze the most frequently addressed research topics in ASE literature, their evolution over time, and the automotive application areas in which they have been investigated.

RQ3. What are the most frequently applied research types and research methods? *Rationale*: This question is designed to identify and classify primary studies based on the type of research and the research methodology. The research type indicates the nature of the enquiry employed in the study, whereas the research method refers to the research methodology applied.

RQ4. What kind of contributions are provided by studies related to ASE? *Rationale*: The purpose of this question is to investigate and classify the ASE literature with respect to their contribution facet. In this question, the contribution facet indicates the type of contribution provided by the primary studies relevant to ASE.

RQ5. What are the research gaps and promising future research directions related to ASE? *Rationale*: This question was designed to identify potential research gaps by analyzing the identified research topics with respect to their research type and contribution facet. This analysis enables us to identify areas that may require additional investigation and potential future directions of research on ASE.

3.2. Search strategy

One crucial step in systematic mapping studies is the identification of as many relevant primary studies as possible to answer the research questions (Zhang et al., 2011). To develop and evaluate the search strategy, several practices and techniques are proposed in the literature (Petersen et al., 2015). In our research, we followed an iterative approach that allowed the investigation and improvement of the search string. Several experimental searches were piloted in the early stages of this research (during May–June 2015) in order to determine the search strategy that would minimize noise and appropriately retrieve relevant studies on ASE. The pilot phase started with the following search string: *Automotive AND Software*.

Using this search string to search only on meta-data information in bibliographic databases led to an unmanageable search result of 18,803 articles. Our initial hypothesis was that the noise ratio must be very high because of the use of two very broad separate terms in the search string. Obviously, many articles use those keywords while not falling within the scope of our research. To evaluate our hypothesis in an objective manner, we conducted the search in Google Scholar, and the top 100 results were closely investigated to evaluate the noise and the effectiveness of the search string. In this set of 100 studies, we found 87 articles that were relevant to the scope of our study. The 87 candidate papers then were investigated further. Consequently, we learned that 74 of the total number of studies (85%) in the pilot dataset commonly used "automotive software" in their abstract or body. Thus, we revised our search string and decided to use *"automotive software"* because our initial investigation showed that a relatively large proportion of the relevant studies identified in the pilot phase commonly used this keyword. The number of retrieved studies using this search string was 2539, reducing the number of papers to be reviewed to 86% of the original total. In comparison to the pilot search string, our understanding is that the revised search string had less noise (proportion of irrelevant studies to all retrieved records) and a higher accuracy rate (proportion of relevant studies to all retrieved records). In our initial investigation, we also learned that many relevant articles typically used general terms, such as "cyber-physical systems" or "embedded software" in the title, keywords, or abstract. Therefore, to increase the coverage of the published studies, we decided to conduct searches of full texts whenever possible[1] rather than searching only the meta-data information (i.e., title, keyword, abstract). To complement our search strategy and increase coverage of published studies, we have conducted several complementary searches.

- In addition to the bibliographic databases, we searched the Society of Automotive Engineers (SAE) Digital Library, which is a domain-specific database that includes a wide range of journals, technical reports, and standards documents related to automotive, aerospace, and defense engineering. However, the search functionality provided by this database is limited, and it is not possible to perform full-text searches. Because of the importance of this database, particularly in the automotive domain, and to minimize the risk of missing relevant articles, we decided to apply the wide search string used in the pilot phase and instead limit our search by using filters.[2] The search results returned 9185 records that included journal articles, standards, and technical documents. To export the candidate studies in bulk, we developed a small script that automatically extracted journal articles based on publication type as provided by the database. Using the script, we were able to identify and retrieve 779 journal articles as candidate studies to be reviewed. The remaining records returned by the SAE database (8406) included non-peer reviewed technical documents and standards, which were beyond the scope of this research.
- We have learned that various groups and research communities within the area of ASE research use very different terms to refer to the same phenomenon. These communities typically characterize their work differently and tend to use various terms such as "automotive software", "in-vehicle software", "safety-critical software", "automotive embedded software", and many more. Two example of such communities are Computer Software and Applications Conference (COMPSAC) and Intelligent Vehicles (IV) Symposium. There are several studies published by such communities relevant to ASE, although they are not always characterized themselves as such. Furthermore, there are many techniques and methods in the automotive industry borrowed from other similar domains, such as the aerospace industry and safety-critical research. There are several articles in the safety-critical research literature that might be relevant and might have an implication on ASE. To have a coverage of relevant subject areas e.g. safety-critical research, we have conducted a complementary search and retrieved published studies in the literature. The search was conducted on Scopus database[3] and the search string was "safety critical" AND "software" AND ("automotive" OR "vehicle"). The total number of retrieved studies was 285 records. Furthermore, we have conducted manual search in order to have a better coverage of COMPSAC and IV Symposium. The search was performed on Scopus and the search string was ("automotive" OR "vehicle") AND "software". The search result was limited to the studies published by those venues and total number of retrieved studies was 135 records.
- We also learned that our search string does not have a proper coverage of automotive infotainment and telematic research. To include published studies related to those research areas, we have conducted a complementary search. The search was conducted on Scopus database[4] and the search string was ("infotainment" OR "telematic" OR "entertainment" OR "interaction design" OR "usability" OR "user interface" OR "human interface" OR "multimedia") AND "software" AND ("automotive" OR "vehicle"). The total number of retrieved studies was 610 records.

Table 1
Search results.

Database	Search field	Retrieved records
IEEE Xplore	Fulltext	694
ISI Web of Science	Metadata	148
Science Direct	Fulltext	84
Scopus	Fulltext	1289
ACM Digital Library	Fulltext	324
Complementary Search (SAE Library and Scopus)	Metadata	1809
Total		4348

These design decisions helped us to design an appropriate search strategy that retrieved a large portion of the relevant studies on ASE. The selected bibliographic databases and number of retrieved studies (up to 15 June 2015) are presented in Table 1. These databases were selected based on their coverage of the software engineering literature.

3.3. Selection criteria

Well-established and explicit selection criteria are essential in systematic mapping studies because they guide the selection of relevant studies that are within the scope of the research. To select candidate papers that were relevant to our objective, we established a set of inclusion and exclusion criteria. We avoided imposing many restrictions on the selection of the primary studies

[1] Fulltext searches are not provided by the Web of Science database, so instead we employed topic search, which uses meta-data information and other indexing fields.

[2] The search results were filtered to technical/journal articles (content) and automotive/commercial vehicles (sector).

[3] The search result was limited to computer science subject area and studies published in English.

[4] The search result was limited to computer science subject area and studies published in English.

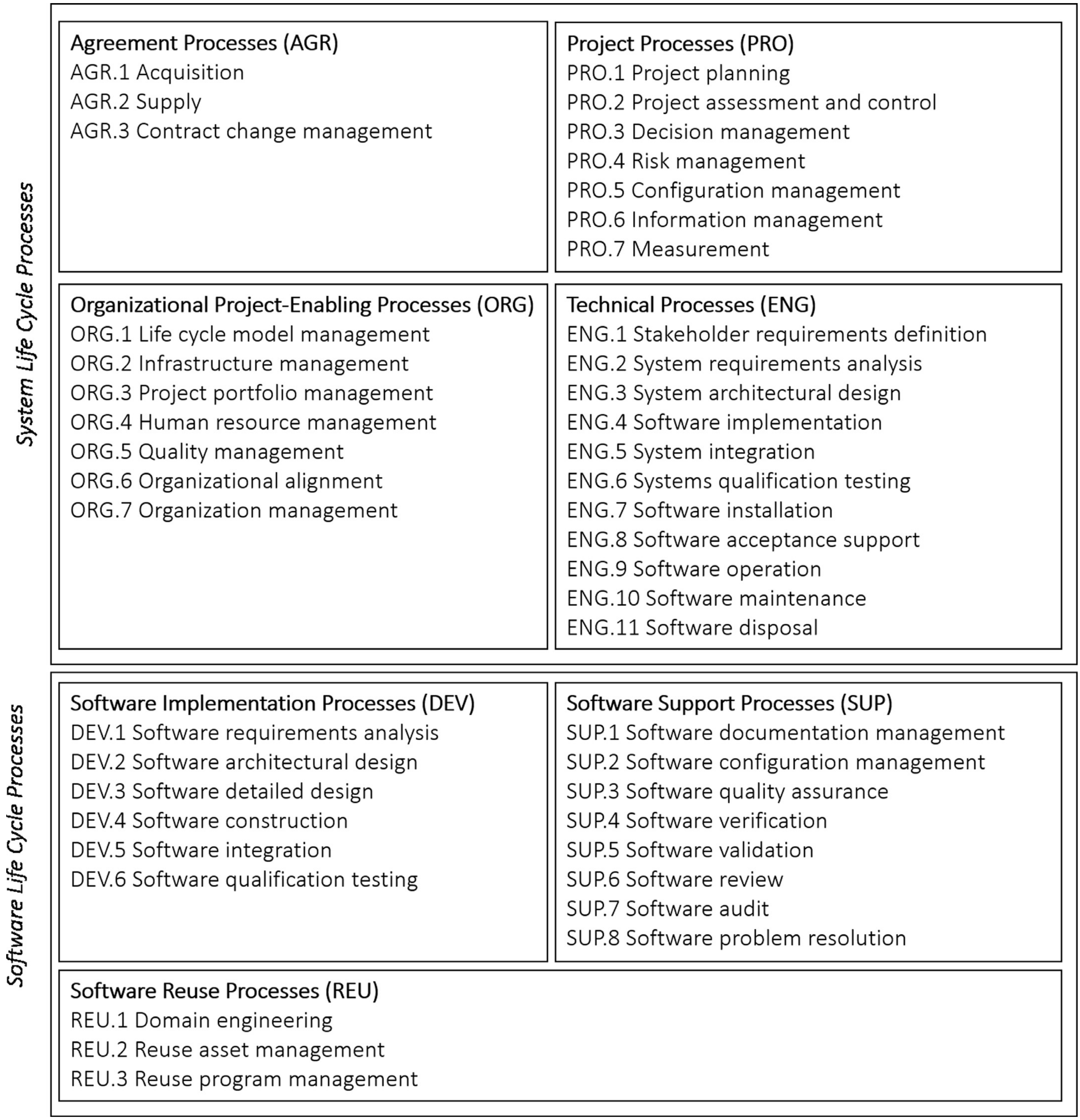

Fig. 2. ISO/IEC 12207 PRM borrowed from (International organization for standardization, 2008).

because our intention was to investigate a broad overview of the previous research on ASE.

To be included in our review, the contribution of the study must be in the context of software engineering in the automotive domain. To narrow our inclusion criteria, we used the ISO/IEC 12207 process reference model (PRM) (International organization for standardization, 2008), which constitutes a wide range of processes used in the development of software-intensive systems. These processes were further classified into different process categories and process groups, as shown in Fig. 2. Thus, any study conducted in the context of the automotive domain that discussed any problem or solution that was relevant to software engineering processes was included. We applied the following exclusion criteria:

1. Beyond the scope of this research, that is, not related to the automotive domain or to software engineering, that is, studies on the mechanical or electronic aspects of the automotive industry
2. Duplicated articles
3. Articles that were not peer reviewed (e.g., presentations, calls for papers, keynote speeches, prefaces, technical reports, technical standards, etc.)

4. Articles not written in English
5. Short papers (e.g., articles published in the short paper section of conference proceedings)

3.4. Selection process

The selection process employed in this study is presented in Fig. 3. To consolidate the studies retrieved from various sources, we used a reference management system (RefWorks) to aggregate and export the data into a unified spreadsheet document. The selection process comprised various stages, which are described in this section.

In the first stage, two researchers simultaneously examined a list of the retrieved studies (4348). Duplicate articles were identified and removed by evaluating the title of the primary study and the periodical. Furthermore, studies not published in English and non-peer reviewed publications, such as calls for contributions, introductions to workshops and proceedings, and tables of contents, were identified and removed accordingly. At the end of this stage, 3261 articles remained. In the second stage, based on the first exclusion criterion, the remaining records were subjected to an in-depth review. In the first step, two researchers simultaneously reviewed the titles of the primary studies and the periodical in which they were published in order to eliminate those that were clearly beyond the scope of this research. For instance, studies that discussed aspects not related to software engineering in the automotive industry, such as network topology, electronic architecture, mechanical design, and so on, were identified and excluded. Furthermore, upon closer investigation of article titles and periodical names, we identified several studies that focused on other domains, such as aerospace, telecommunication, materials and manufacturing, and mechanical engineering. Overall, we were conservative at this stage and excluded only studies that were clearly beyond the scope of this research. In cases when we were not sure about the focus of the article under evaluation, the paper was passed to the next screening phase. Thus, based on the title of the primary study and the periodical, we excluded 1118 articles.

The remaining records were investigated in the next step, in which two researchers individually reviewed the abstracts of the articles and recorded their decisions. By crosschecking the votes, they agreed to include 533 articles and exclude 1243 articles. The remaining 367 records were marked as unsure and subjected to the extended review. The extended review was based on examining the introduction and conclusion, and each article was reviewed individually by two researchers. In many cases, we had to perform a full-text review in order to ensure that the focus of the article was within the scope of this work. In cases of conflict between two researchers, a third reviewer was asked to mediate, and the final decision was made collectively. Thus, of the records (367) remaining from the previous step, we were able to include 212 records and exclude 155 records. Overall, 745 articles were identified as relevant to the scope of our study and were included in the initial pool of candidate studies. To ensure the consistency of the list of primary studies and to become familiar with the available data provided by the primary studies, we conducted an additional full-text review. This stage enabled us to further identify and exclude short papers (32), a few extra non-peer reviewed studies (12) and some articles for which the full text was not available (22). Finally, 679 primary studies remained in the pool, were identified as relevant, and were used in the data extraction and analysis phase.

3.5. Data extraction

To answer the research questions, we designed a data extraction form, which is presented in Table 2. The first two columns indicate the data property identifier and name, and the third column refers to the cardinality of the relationship between an individual primary study and the data property. In addition, the fourth column presents traceability between an individual data property and our research questions. The data properties and the data extraction procedure are elaborated in this section.

Table 2
Data properties.

ID	Title	Cardinality	RQ
DP1	Publication Year	1:1	RQ1
DP2	Publication Channel	1:1	RQ1
DP3	Periodical Title	1:1	RQ1
DP4	Citation Number	1:1	RQ1
DP5	Country (Authorship)	1:*	RQ1
DP6	Institute (Authorship)	1:*	RQ1
DP7	Contributors Type (Authorship)	1:1	RQ1
DP8	Investigated Software Engineering Processes	1:*	RQ2
DP9	Automotive Application Area	1:*	RQ2
DP10	Research Type	1:1	RQ3
DP11	Research Method	1:1	RQ3
DP12	Contribution Type	1:*	RQ4

3.5.1. Data properties

Twelve data properties, which are briefly described below, were established and defined in order to answer our research questions. The detailed definitions of these data properties are presented in Appendix A.

- Generic Data Properties (DP1–DP7): The generic data properties were automatically retrieved from bibliographic databases and used to analyze the intensity of the research on ASE. The number of citations of the paper was manually collected using Google Scholar on 11 Jan 2016. Furthermore we have also extracted authorship information from the primary studies in order to characterize who undertakes research in the ASE area.
- Investigated Software Engineering Processes (DP8): ISO/IEC 12207 PRM (International organization for standardization, 2008) was used to classify the primary studies with respect to their contribution. The classification schema presented in Fig. 2 covers a wide range of processes, from the system to the software engineering lifecycle. Based on the contribution of the primary study, each article was assigned to one or more software engineering processes. We further used the data property to integrate similar processes and generate a unified list of recurring research topics.
- Automotive Application Area (DP9): To classify and analyze primary studies based on their application area, we adopted a classification schema proposed by Pretschner et al. (2007) and Harris (2013). The application areas indicate to the type of software deployed in the automotive systems. The classification schema includes various types, such as 1) body and comfort software, 2) driver assistance software, 3) chassis and safety software, 4) powertrain and transmission software, 5) infotainment and telematics software, 6) infrastructure software, and 7) not applicable.
- Research Type (DP10): To analyze the type of research in the literature on ASE, we adopted the approach proposed by Wieringa et al. (2006). The research type indicates the nature of the enquiry applied in the study and includes various types, such as 1) solution proposal, 2) validation research, 3) evaluation research, 4) philosophical paper, 5) opinion paper, and 6) experience report.
- Research Method (DP11): To classify primary studies based on their applied research methodology, several research methods were considered, including 1) case-study research, 2) experimentation, 3) survey, 4) design research, 5) action research, 6)

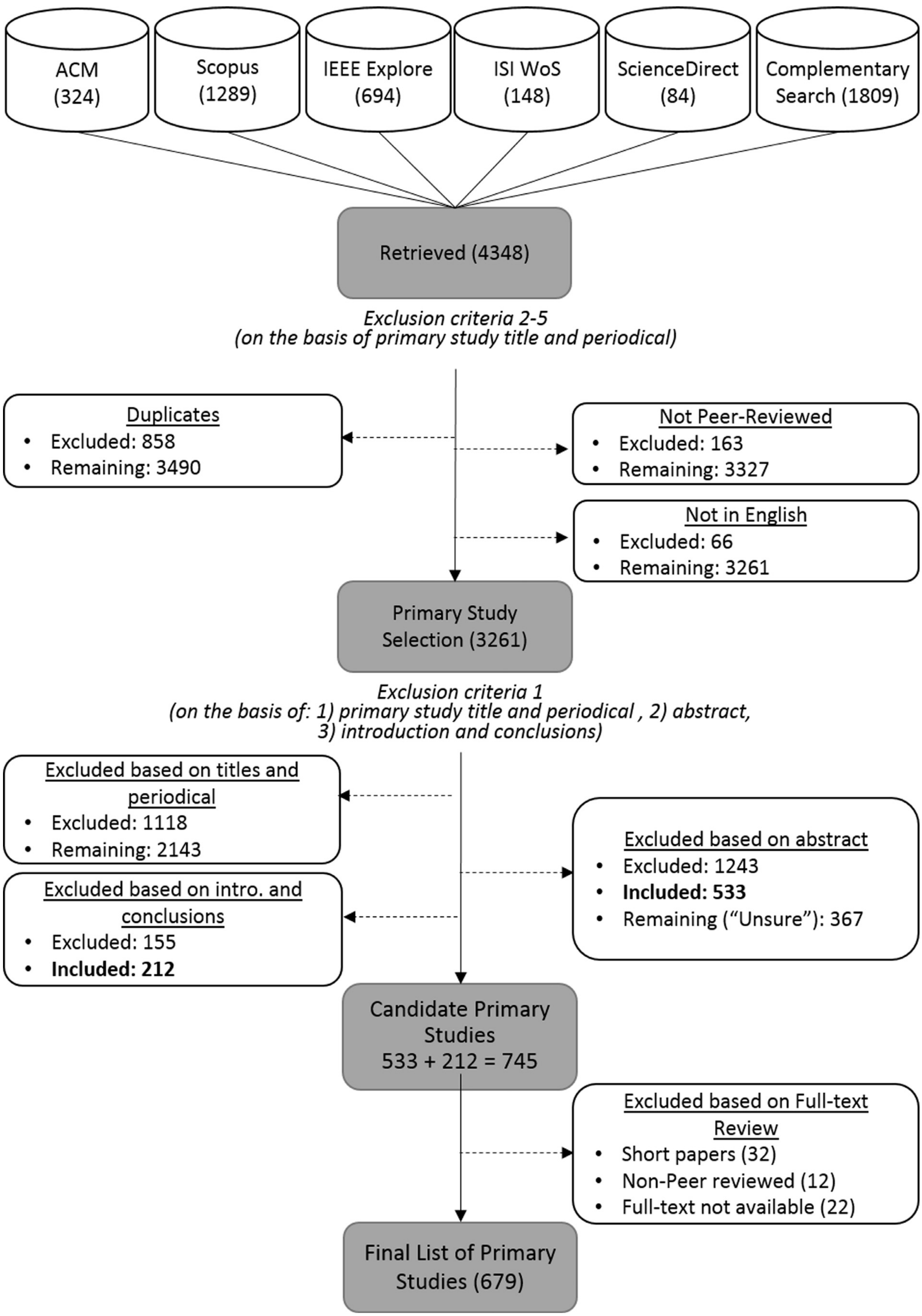

Fig. 3. Selection procedure.

grounded theory, 7) systematic reviews, 8) mixed method, and 9) not applicable.
- Contribution Types (DP12): To classify literature on ASE based on contribution facet, the classification schema for contribution type was adopted from Shaw (2003) and Paternoster et al. (2014). Several contribution types, including 1) framework/method/technique, 2) guideline, 3) lesson learned, 4) model, 5) tool, and 6) advice/implications, were considered.

3.5.2. Data extraction procedure

Based on the data properties elaborated in the previous section, we developed a data extraction form. In the early phases of this stage, we used a pilot group of 50 primary studies to develop a common understanding of the data properties. The purpose of the pilot study was not to calculate any agreement level but solely to evaluate and improve our data extraction form by testing it on a small set of primary studies. The full text of each primary study was reviewed individually by two researchers, and the findings were recorded on separate data extraction sheets. In the next step, we integrated the data extraction sheets and identified conflicts by crosschecking the results. To resolve these disagreements, a third reviewer joined the original two reviewers and the rationale for resolution was discussed and communicated to the rest of reviewers through meetings.

3.6. Data analysis and interpretation

Descriptive statistics and quantitative descriptions of frequencies were mainly used to analyze the literature on ASE with respect to research intensity, frequently investigated research topics, frequently applied research types, research methods, and contribution types. To identify potential research gaps, we performed a combinational analysis and examined each research topic with respect to the research type and contribution type. The results of our systematic map are presented in Section 4. Given our objective in this work, we did not intend to perform an in-depth qualitative analysis, but we present an overview of the existing contributions and the frequently investigated areas of research in the ASE literature.

3.7. Threats to validity

There are several potential threats to the validity of this research that must be carefully considered in interpreting our results. In this section, we elaborate the threats to the validity of our work and the strategies used to minimize their effects. In the context of empirical software engineering, validity threats are classified into four distinct categories, including construct validity, internal validity, external validity, and reliability (Wohlin et al., 2000). In the context of this research, construct validity relates to the data collection strategy and how well the data represent the investigated phenomenon. Internal validity concerns the relationship between constructs and proposed explanations, that is, whether the actual conclusion of the study is valid. External validity concerns the possibility of the generalizability of the results within and beyond the scope of study, whereas reliability concerns the repeatability of the research and the possibility of reaching the same conclusion reached by the original study.

3.7.1. Construct validity

In this section, the threats to the validity with respect to the identification and selection of the primary studies and publication bias are discussed.

Primary study identification. In systematic literature reviews, the identification of as many relevant primary studies as possible within the scope of the research is a critical task (Zhang et al., 2011). This is particularly important in systematic mapping studies because the objective is to structure the literature and provide an overview of the existing contributions in a specific area. However, in practice, this is very challenging, particularly in reviews of broad and trending subjects. The main threat to the validity of this work is related to the search strategy and its completeness. This is perhaps due to the fact that various groups and research communities within the area of ASE research use very different terms to refer to the same phenomenon. Furthermore, there are many techniques and methods in the automotive industry borrowed from other similar domains, such as the aerospace industry and safety-critical research. There are several articles related to the safety-critical systems that might be relevant and might have an implication on ASE, although they are not always characterized themselves as such. Taken together, these issues are due to the limitation of our search strategy and make it challenging to retrieve all studies that are relevant to our scope and represent a moderate threat to the validity of the present review.

To mitigate the above-mentioned issues and reduce the risk of missing relevant studies, we followed an iterative approach and employed a search strategy to identify as many relevant primary studies as possible. On the basis of several experimental searches and our initial investigation, we designed an appropriate search strategy that retrieved a large portion of the relevant studies within our scope. However, a limitation of the current search string ("automotive software") lies in its coverage and whether it incorporates all relevant and important primary studies within the scope of our review. The proposed search string used in the pilot phase ("automotive" AND "software") might be the most comprehensive search option with respect to coverage. However, conducting a search only on meta-data information led to an unmanageable search result of 18,803 articles. In addition, our initial investigation showed that the amount of noise was very high because of the use of two very broad separate terms in the search string. Obviously, many articles in the literature use those keywords while not falling within the scope of our research.

In a close investigation performed during the early stage of this research, we learned that a large portion of the relevant studies (85%) that were identified in the pilot dataset commonly used "automotive software" in their texts. However, general terms, such as "cyber-physical systems" and "embedded software" were common in the titles, keywords, or abstracts. Therefore, to increase the publication coverage, we performed full-text searches in various bibliographic databases with wide coverage of the software engineering literature. The search was further complemented by performing additional searches in order to have a better coverage of studies published by SAE Digital library, safety-critical research, and studies relevant to the infotainment and telematic research in automotive industry.

Thus, our search strategy was iteratively revised based on the above design decisions. Given the broad scope of our subject, one can argue that it is not easily possible to retrieve and analyze "all" relevant studies, as has been acknowledged in previous studies (Wohlin et al., 2013; Petersen et al., 2015). Although the objective of systematic reviews is to find all relevant research on the area of interest, it is more likely that a sample of the relevant research articles would be obtained (Wohlin et al., 2013). The main purpose of our search strategy was to retrieve as many relevant primary studies as possible while minimizing the amount of noise and the amount of effort in the selection process. Other search strategies such as a manual search or snowballing with a set of known-papers might have been employed. However, these strategies come with their own limitations, and there is no reason to believe that they would have been more appropriate.

Primary study selection. The selection of primary studies that are within the scope of the study is a critical task that poses a threat to the validity of the literature review. For instance, relevant primary studies might be excluded because of the researcher's bias, subjective evaluation, or misjudgment of the study under evaluation. To mitigate this threat, a thorough research protocol and a set of inclusion and exclusion criteria was established in the early stages of this research. To narrow our scope, we employed ISO/IEC 12207 PRM as the guideline to establish our inclusion criteria. To minimize the subjectivity of the selection, each step was conducted by pairs of researchers and disagreements were resolved with the help of a third researcher. Overall, the researchers tended to be conservative in the selection process, and unclear cases were passed to the next stage for extended screening or full-text review.

Publication bias. Publication bias relates to "the problem that positive research outcomes are more likely to be published than negative ones" (Kitchenham et al., 2004). Publication bias can be seen as a potential threat in literature studies, particularly in systematic literature reviews that aim to aggregate existing evidence and to evaluate and compare specific methods or techniques. However, in systematic mapping studies, the effect of this threat is trivial because such analyses are usually not performed.

3.7.2. Internal validity

In our work, internal validity concerns those risks that might lead to a wrong or irrelevant conclusion. The main threat to the internal validity of our work is related to the researchers' bias in the extraction, analysis, and interpretation of the data in the primary studies. To minimize the effect of this threat, we defined and established a data extraction form with a detailed description of each data property, which is presented in Appendix A. This protocol was discussed and pilot tested early in this research. The aim of the pilot study was to establish a common understanding among the researchers and to evaluate and improve our data extraction form by testing it on a small set of primary studies. Overall, five researchers were closely involved in this research, which enabled triangulation in all phases and minimized the possibility of researcher bias. The first three authors in this work actively participated in all phases, whereas the last two authors acted as external reviewers to validate the research protocol and guide the review process. The data extraction, analysis, and interpretation of our primary studies were performed by pairs at all stages, and disagreements were further resolved with the help of a third researcher and collective discussion.

3.7.3. External validity

This validity threat concerns the possibility of the generalizability of the findings within and beyond the scope of the study. This work represents a comprehensive panoramic overview of ASE research and includes combined analysis of large number of primary studies within our scope. This study includes in total 679 studies from multiple research sub-areas, published between 1990 and 2015. The primary studies reviewed in this study were published by authors in over 289 unique publication venues (journals, conference proceedings, etc.) and include author affiliations with 398 unique institutes in 35 countries. These figures indicate that our primary studies are not limited and biased towards a specific time period, publication venue, research institute, or geographical area. As acknowledged in the previous studies cited (Wohlin et al., 2013; Petersen et al., 2015), despite the objective of systematic reviews to retrieve all relevant contributions, it is likely that only a sample of the relevant research articles will be obtained. Even though the present study does not have the full coverage of ASE literature, it includes a large and representative sample of relevant studies. Thus adding more primary studies to the analysis should not change the qualitative conclusion of the study. Furthermore, some of our findings might be relevant to other similar domains in particular and embedded software development in general. However, the overall effect of this threat is trivial because the aim of our systematic mapping study is not to generalize the findings outside of the subject of automotive software engineering. Hence, the findings of this study should not be generalized beyond its original scope.

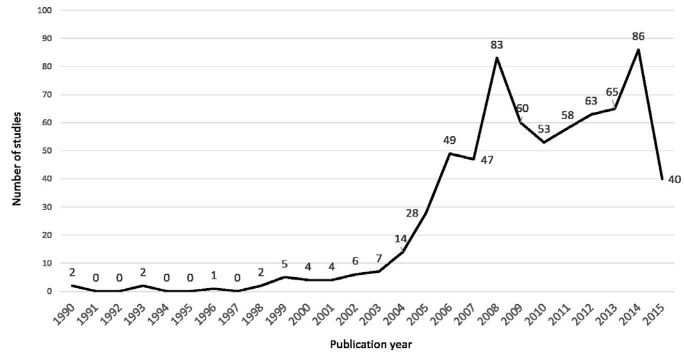

Fig. 4. Number of publications per year.

3.7.4. Reliability

Repeatability required a detailed research protocol and a thorough report of the research process applied. Conducting the search in bibliographic databases and retrieving the studies are part of an objective process that must be repeatable by strictly implementing the same research protocol. However, the selection of the primary studies and the data extraction, interpretation, and classification of the literature were subjective and belong to the research creativity and research contribution.

4. Findings

This section is structured according to the research questions, which are answered by analyzing and synthesizing the results of the data extraction. The full list of primary studies included in this review along with our data extraction results is available on the web[5]

4.1. RQ1. What is the intensity of research activity on ase?

To answer the first research question, five aspects of the included studies were examined: publication trend over time, distribution of primary studies with respect to their publication channel, most frequent publication venues (specific journals, conference proceedings, etc.), authorship information, and the number of citations in the ASE literature.

4.1.1. Publication trend

The 679 publications were distributed over the years between 1990 and 2015. Fig. 4 shows the evolution of the publications in the ASE literature. The research activity in ASE from 1990 to 2002 was linear, with a low number of publications. From 2002 to 2006, the research activity in ASE increased significantly, and in 2006,

[5] Link to the online repository: https://www.dropbox.com/s/n7ix7h5yk9puavs/ASE_SMS_Repository_17022016.xlsx?dl=0.

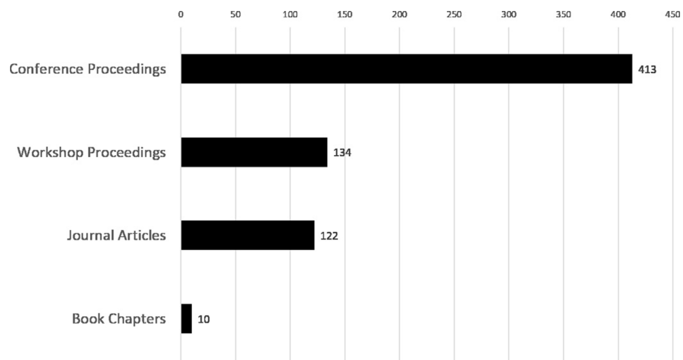

Fig. 5. Publication channel.

Table 3
Top 10 represented journals.

Title	No.
SAE International Journal of Passenger Cars- Electronic and Electrical Systems	39
Journal of Systems and Software	10
Proceedings of the IEEE	5
Electronic Notes in Theoretical Computer Science	4
Information and Software Technology	4
IEEE Software	4
IEEE Transactions on Software Engineering	3
International Journal of Automotive Technology	3
Journal of Software Testing, Verification and Reliability	3
Journal of Software and Systems Modeling	3

there were 49 publications. This considerable increase in the research interest is perhaps due to the advent of networked software in vehicle systems, which became much more prevalent in the late 1990s and early 2000s, along with the exponential growth in both automotive software size and complexity. These factors also explain both the establishment of the first ICSE Workshop on Software Engineering for Automotive Systems (2004) and the significance of early studies by Grimm (2003) and Broy (2005), which were highly influential in the area. Thereafter, from 2006 to 2008, the research activity in ASE increased notably, and the number of published articles reached 83 in 2008. During this period, two seminal articles that highly influenced the area, Broy (2006) and Pretschner et al. (2007), summarized the research opportunities in ASE. Following this trend, the number of publications on ASE decreased slightly and then increased from 2010 onwards, reaching 86 publications in 2014. Perhaps one reason for this slight decrease in the number of publications is the fact that one of the biggest sources of articles, the ICSE Workshop on Software Engineering for Automotive Systems was no longer active. It is also important to highlight that we cannot make a valid conclusion with respect to 2015 as our search covers only part of that year. Overall, even though the number of publications on ASE changed over time, the research activity in this area continues to increase, and its evolution has shown stable growth, particularly during the last two decades.

4.1.2. Publication channel and venues

This mapping study covered 47 different journals, 182 conferences, 60 workshops, and 10 book chapters. With respect to their publication channel, most of the primary studies presented in Fig. 5 were conference papers (413, 61%), followed by workshop proceedings (134, 20%), journal articles (122, 18%), and book chapters (10, 1%). With respect to the venues in which ASE studies were published, Tables 3–5 show the top ten journals, conferences,

Table 4
Top 10 represented conferences.

Title	No.
SAE World Congress and Exhibition	55
SPLC International Software Product Line Conference	15
COMPSAC International Conference on Computers, Software and Applications	14
DATE Proceedings of the Conference on Design, Automation and Test in Europe	13
ICSE International Conference on Software Engineering	10
ISoLA International Symposium on Leveraging Applications of Formal Methods, Verification and Validation	10
DAC Design Automation Conference	9
SEAA Euromicro Conference on Software Engineering and Advanced Applications	7
ECSA European Conference on Software Architecture	7
RE International Conference on Requirements Engineering	6

Table 5
Top 10 represented workshops.

Title	No.
ICSE Software Engineering for Automotive Systems	24
ASWSD Automotive Software Workshop	14
International Dagstuhl Workshop on Model-Based Engineering of Embedded Real-Time Systems	9
CompArch Workshop on Automotive Software Architectures	7
VaMoS Workshop on Variability Modeling of Software-Intensive Systems	5
ISSRE International Symposium on Software Reliability Engineering	5
ICSTW Software Testing Verification and Validation Workshop	4
COMPSAC Computer Software and Applications Conference Workshop	3
ICSE International Workshop on Automation of Software Test	3
ICSE Workshop on Software Engineering for Embedded Systems	2

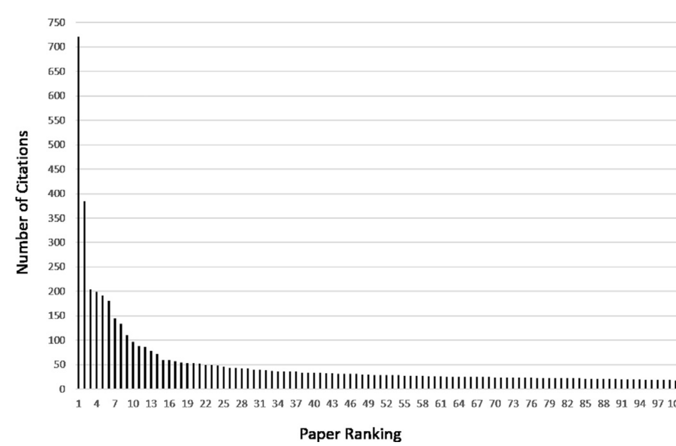

Fig. 6. Number of citations for the 100 most cited papers.

and workshops. The SAE International Journal of Passenger Cars- Electronic and Electrical Systems and the Journal of Systems and Software were the top contributors among the journals. The SAE World Congress & Exhibition and International Software Product Line Conference were the top venues for conference papers. Regarding the workshops, ICSE Software Engineering for Automotive Systems and the ASWSD Automotive Software Workshop produced the highest number of published workshop papers.

4.1.3. Citation impact

Fig. 6 presents the number of citations of the 100 most cited primary studies. Overall, nine papers (Pretschner et al., 2007; Broy,

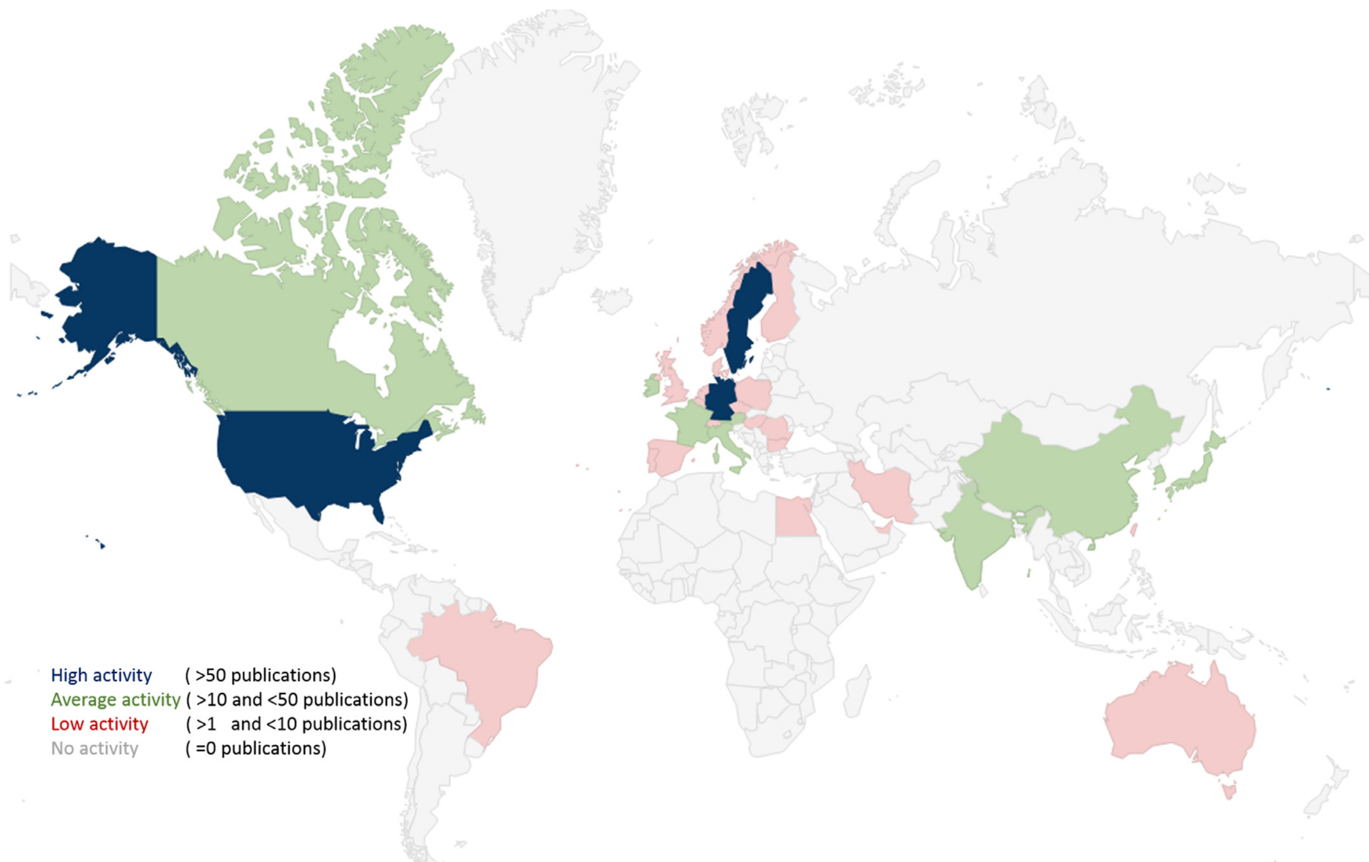

Fig. 7. List of most active countries in ASE research.

2006; Weber and Weisbrod, 2002; Broy et al., 2007; Pretschner et al., 2005; Alur et al., 2003; Grimm, 2003; White et al., 2008; Müller et al., 2006) had more than 100 citations each. The total number of citations for the top 100 papers was 5,332, and the average number of citations was 53.

4.1.4. Authorship information

To characterize who undertakes research in the area of ASE, we analyzed the authorship information of previous primary studies. There are in total 247 studies that have all academic authors, followed by 240 studies with a mix of academic and industry authors (collaborative work), and 192 articles have all industry authors. This result indicates that the level of industry–academic collaboration is quite high and practitioners have significantly contributed to ASE research. Fig. 7 presents the list of countries that are most active in ASE research. Overall, there are 35 unique countries identified in the previous primary studies. Germany (with 289 published studies) is by far the most active country in the research area, followed by the United States and Sweden (98 and 95 studies respectively). The countries marked in green are relatively less active in the research area (more than 10 and less than 50 studies), while the remaining countries in red have a low number of publications in ASE research (less than 10 studies each).

To understand who has contributed the most (in terms of number of publications) among our primary studies, we have aggregated the author affiliations. There are 398 unique institutions among our primary studies, the majority of which are from industry (OEM or suppliers) rather than academic institutions. Table 6 shows the top 20 institutions with the highest numbers of publications.

Table 6
List of most active institutions in ASE research.

Title	No.
Daimler AG	49
Technical University of Munich	47
Volvo Car Corporation	45
Bayerische Motoren Werke AG	32
Robert Bosch GmbH	28
Chalmers University of Technology	28
Mälardalen University	23
University of Gothenburg	19
RWTH Aachen University	15
University of California	14
General Motors	14
Technical University of Berlin	13
Ford Motor Company	13
Delphi Automotive	12
Fraunhofer Institute for Open Communication Systems	12
ETAS Group	12
Graz University of Technology	12
dSPACE GmbH	11
Continental AG	11
MathWorks Inc.	10

4.2. RQ2. What are the most frequently investigated research topics? How have they evolved over time? And in what application area have they been investigated?

This research question concerns the analysis of the most frequently addressed research topics in ASE literature, their evolution over time, and the automotive application areas in which they have been investigated. To address RQ2, we conducted two types of analysis including: 1) keyword analysis by word cloud visualiza-

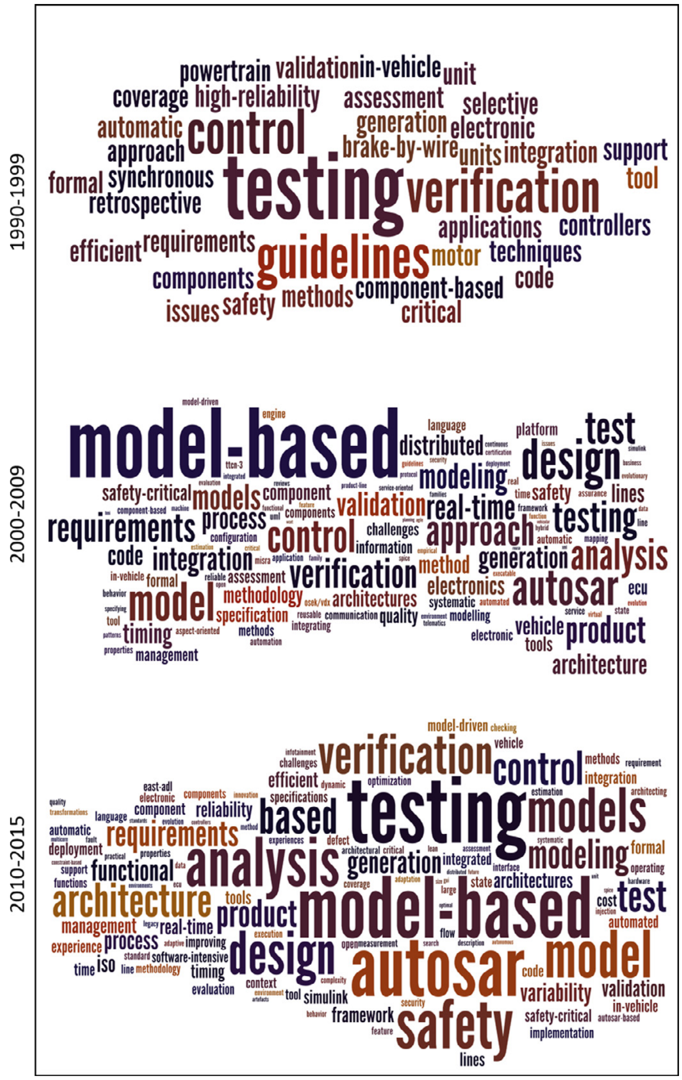

Fig. 8. ASE literature focus areas in each decade.

tion of primary study titles in different decades, and 2) thematic analysis by classification of primary studies with respect to their contribution. Each primary study was analyzed by two researchers and assigned to one or more software engineering processes. In the next step, similar software engineering processes were integrated in order to determine higher-level themes or recurrent research topics. For instance, the system qualification (ENG.6) and software qualification testing (DEV.6) processes were integrated to derive a unified research topic. Furthermore, the reported automotive application areas from primary studies were aggregated and analyzed in order to identify which part of automotive systems were most investigated in the literature.

4.2.1. Keyword analysis: focus areas of the studies through each decade

To characterize research trends in ASE literature, we conducted word cloud analysis in order to identify how the focus areas of studies have been changing by time. Fig. 8 shows the word cloud of primary study titles, classified by the decades of their publications years.[6] The word cloud analysis visualizes the most frequent keywords used in ASE literature. This might be an indication i.e. to characterize focus areas of the studies, to identify new research trends, hot topic areas, and important development paradigms in the automotive software industry. As we can see, topics such as "architecture & design", "requirement", "testing", "verification & validation" , and "integration" were the most common in all three decades, while the focus areas have shifted to i.e. "model-based software development", "distributed systems", "real-time", "safety-critical" and "AUTOSAR" in 2000s, and to different emerging topics i.e. "safety", "deployment", "variability", "reliability" and "security" during 2010–2015.

4.2.2. Thematic analysis: research topics and their evolution over time

Table 7 presents a list of the recurrent topics along with their frequency (number of primary studies) and coverage (with respect to the ISO 12207 PRM). The results showed that system/software architecture and design was the most frequently investigated topic among the primary studies, followed by system/software qualification testing and software reuse. The remaining topics, which included software verification & validation, software implementation, organizational project-enabling support group, software quality assurance & review, system/software integration, and system/software requirement engineering were the next most frequently investigated topics. However, software construction, software maintenance, project support group, agreement support group, and software documentation & configuration management were relatively less investigated topics among the primary studies. In addition to these research topics, 17 primary studies were marked as not applicable. These studies did not belong to any particular topic area but discussed general aspects or challenges in ASE research. A very brief overview of recurrent research topics in the ASE literature is provided in Appendix B.

To characterize the evolution of the identified research topics, we analyzed the research activities in each topic over time. Fig. 9 shows the evolution of each research topic. We should note again that our data for 2015 are incomplete because our search covered only part of this particular year. Overall, the evolution pattern varied among the identified research topics. The research activity concerning the majority of research topics seems to have had stable growth, particularly with respect to system/software architecture & design, qualification testing, system/software integration, organizational project-enabling support group, and software reuse. However, agreement support, project support group, software construction, and documentation & configuration management were only discussed occasionally, and the low amount of research activity seemed to be reflected by the low number of publications.

4.2.3. Automotive application areas

Automotive software applications are highly heterogeneous and deployed over different application areas. To identify which part of automotive systems were most investigated in the literature, the primary studies were classified according to their automotive application area. The classification schema was adopted from Pretschner et al. (2007) and Harris (2013) and further information about these application areas and their characteristics is presented in the literature. Fig. 10 presents the classification of primary studies with respect to their automotive application areas. The results indicate that body and comfort software, powertrain and transmission software, chassis and safety software, infotainment and telematics software, and infrastructure software were the most investigated areas, in that order. However, the majority of the primary studies (60%) in our review were marked as not applicable. These

[6] The word cloud analysis was conducted by the help of an online tool named Wordle (http://www.wordle.net/). The common English language terms and other common phrases i.e. automotive, development, engineering and systems have been removed from the analysis.

Table 7
Research topics.

No.	Title	Frequency	SE processes
1	System/Software Architecture & Design	131	ENG.3;DEV.2;DEV.3
2	System/Software Qualification Testing	127	DEV.6;ENG.6
3	Software Reuse	72	REU.*
4	Software Verification & Validation	71	SUP.4;SUP.5
5	Software Implementation	62	ENG.4
6	Organizational Project-Enabling Support Group	48	ORG.*
7	Software Quality Assurance & Review	48	SUP.3;SUP.6
8	System/Software Integration	44	DEV.5;ENG.5
9	System/Software Requirement Engineering	35	ENG.1;ENG.2;DEV.1
10	Software Construction	22	DEV.4
11	Software Maintenance	18	ENG.10
12	Project Support Group	14	PRO.*
13	Agreement Support Group	5	AGR.*
14	Software Documentation & Configuration Management	3	SUP.1;SUP.2

studies did not report any specific automotive application areas and only used general terms i.e. automotive systems or automotive control software to describe their context.

4.3. RQ3. What are the most frequently applied research types and research methods?

This research question concerns the research types and research methods applied in the ASE literature. Fig. 11 presents the classification and distribution of research types found in the literature. It is interesting that research types were not represented homogeneously. Evaluation research was the most frequently addressed type, constituting 43% of the primary studies (292 records). These studies presented solution implementation and its evaluation in practice, typically through case studies, surveys, mixed methods or a similar empirical methodology. Solution proposals were the next frequently addressed research type (22% or 151 records). This group includes those studies that present either a novel solution or a significant extension of an existing solution. However, solution proposals remain to be evaluated in practice, and their applicability is investigated through demonstration, example, or argumentation. Experience papers constituted 15% of the primary studies (100 records), including those describing industry-based experiences, explaining what has been done in practice, and how. Validation research was conducted in only 7% of the primary studies (46 records), including studies that investigated a particular technique or solution through a methodologically sound research design, such as experimentation (controlled or quasi-experiments), formal analysis, or simulation. Opinion papers (8%, 54 records) included studies that reflected the personal opinions of an expert about a particular technique and whether or how it should be applied. Philosophical papers (5%, 36 records) include those studies that offer new ways of looking at existing things by how they frame a particular area, often supporting automotive software engineers to better comprehend a problem area by providing a conceptual framework or taxonomy. Fig. 12 shows the evolution of research types over time.

The classification of primary studies with respect to their research method is presented in Fig. 13. It is interesting that about 51% of the primary studies were not grounded in any research methodology (350 records). This category included opinion papers, experience papers, solution proposals, and the majority of the philosophical papers. The most frequently applied research method was the case study, which was applied in the evaluation research and included around 40% of the primary studies (273 records in total). Experimentation (controlled or quasi) was the second most frequently used research method applied in validation research studies and included only 5% of primary studies (34 records). The remaining research methods presented in Fig. 12 were less frequently applied in the literature and constituted only 4% of the primary studies. Although quality assessment of the primary studies was beyond the scope of this work, we observed that the majority of the case studies in our review did not report proper details as recommended by Runeson and Höst (2009), such as the research protocol, data collection and analysis procedure, and contextual information. These limitations further limit the future replication, generalization, and aggregation of evidence in the context of ASE.

4.4. RQ4. What kind of contributions are provided by studies related to ASE?

This research question concerns the contribution type represented by the literature on ASE. Fig. 14 presents the classification and distribution of the contribution types found in the primary studies. Similar to research types, contribution types were not homogeneously represented in the literature. Framework, methods, and techniques (50%, 370 records) were the dominant contribution type among the included ASE literature. These studies provided concrete solutions for the implementation and management of automotive software-intensive systems. The second most addressed type was lessons learned, which constituted 22% (159 records) of the contributions. Tools constituted only 11% (84 records) of the contributions and included studies that provided a particular technology, application, or program in order to support the engineering of software-intensive automotive systems. Advice/implications represented 8% (61 records) of the considered contributions and included those that provided discursive and generic recommendations based on personal opinion. In contrast, guidelines supporting practitioners in industry and models representing an abstraction of observed reality through a conceptualization process constituted only 4.5% (32 records) and 4.5% (33 records), respectively, of the total contributions. Fig. 15 shows the evolution of the contribution types over time.

4.5. RQ5. What research gaps and promising future research directions are related to ASE?

To answer this research question, we investigated the potential research gaps by analyzing the combined results of the previous research questions, and we determined whether this combination could indicate less investigated areas or potential future research directions. Fig. 16 presents the results of our systematic map, which includes three dimensions: research topics, research type and contribution facet. In this section, we highlight the potential research gaps by discussing the results of the analysis of the recurrent research topics with respect to their research type and contribution facet. Thereafter, we review the ASE research roadmap

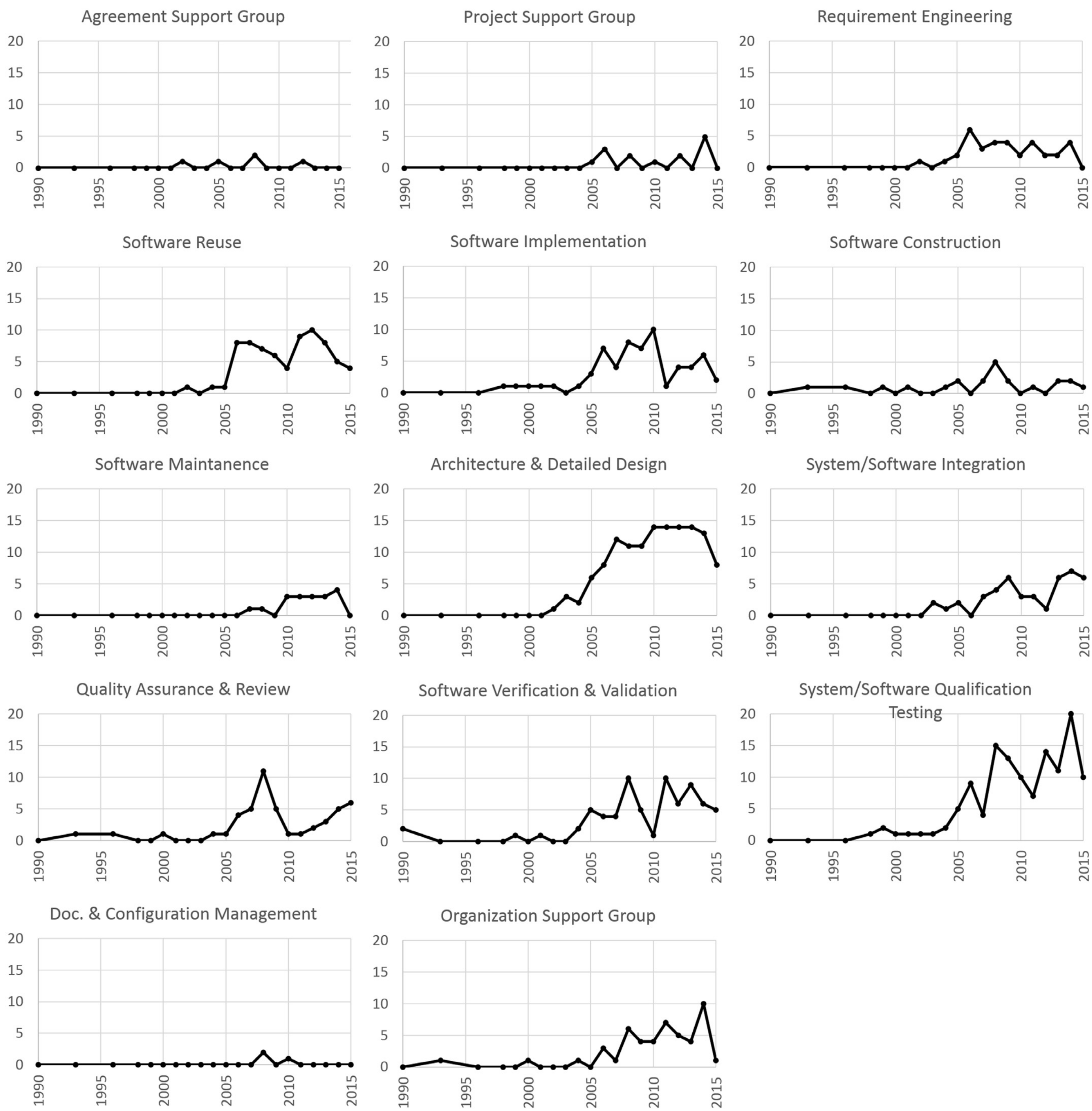

Fig. 9. Research topics by year.

proposed by Pretschner et al. (2007) and Broy (2006) and subsequently discuss how findings from this mapping study relate to the proposed research themes.

4.5.1. Systematic map - research topic, research type and contribution facet

Research topic vs. research types. We have already identified evaluation research as the most common research type (43%), which indicates that a large portion of the studies on ASE topics were conducted in an empirical context. This finding was evident in the majority of topics, such as qualification testing, software implementation, architecture, and design as well as system/software integration. However, the results showed that evaluation research was relatively less common in software construction, agreement support group, software documentation & configuration management.

Solution proposals (22%) were distributed among the various research topics, which is expected in new research areas and fosters research innovation in the field. While this kind of contribution has positive implications for the research area, the applicability is limited from an industrial perspective, as they have not yet been evaluated in practice. Thus, further extension and evaluation of existing solution proposals are needed. Similarly, experience papers (15%) were represented in most of the research topics.

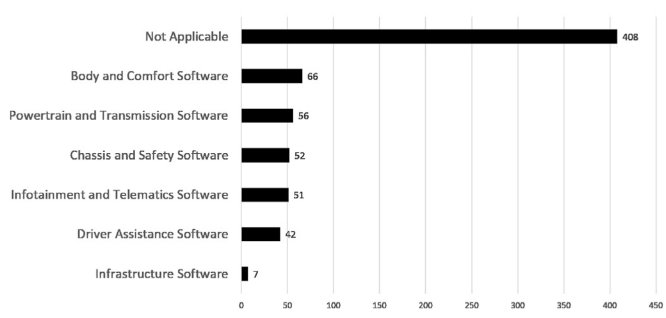

Fig. 10. Automotive application areas.

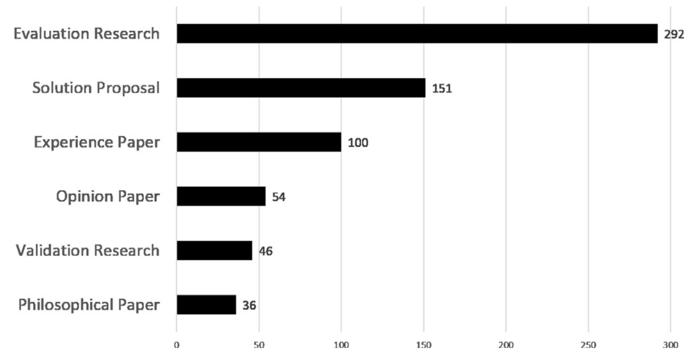

Fig. 11. Research types.

Experience reports include industry-based experiences and explaining what has been done in practice and how. Taking into account the study context and limitations, such contributions are of potential relevance and benefit for practitioners. Further experience papers may be beneficial for both practitioners and researchers because these studies reflect real-world experiences and provide hypotheses for future studies.

Opinion papers (8%) occurred less frequently, but they were still present in the majority of research topics, including studies that reflected the personal opinion of the author about a particular subject. However, validation research was less prevalent in the literature (7%), and only a few research topics were investigated through validation studies. Thus, further validation research studies are required to validate existing solutions through rigorous scientific methodologies and sound research settings. Similarly, philosophical papers (5%) were less presented in the literature. Philosophical studies offer new ways of looking at existing things by framing a particular area in a new way, often supporting researchers and practitioners to better comprehend a problem area by providing a conceptual framework or taxonomy.

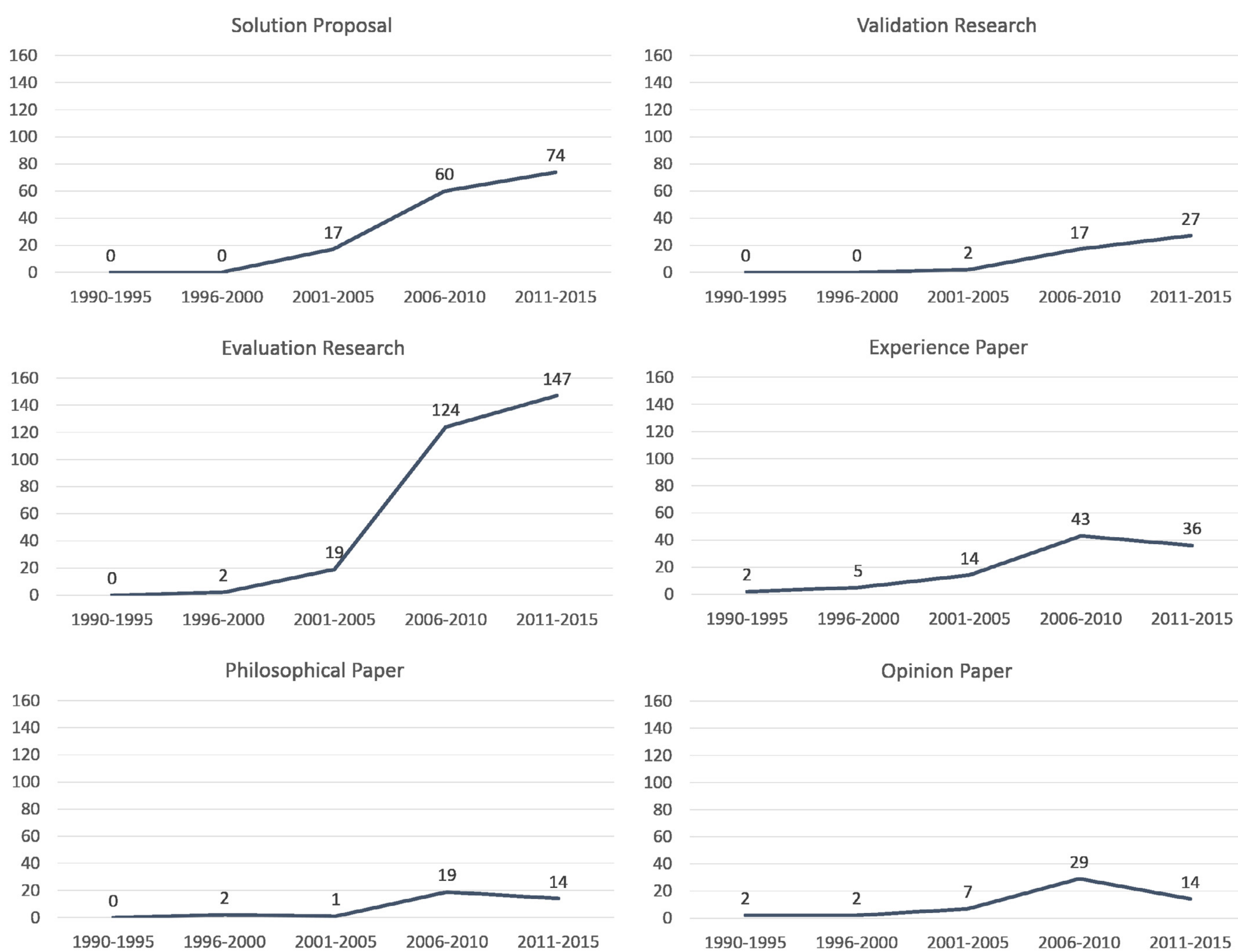

Fig. 12. Research types per year.

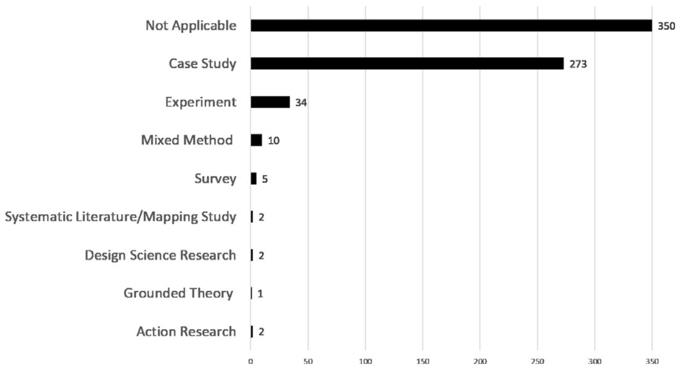

Fig. 13. Research methods.

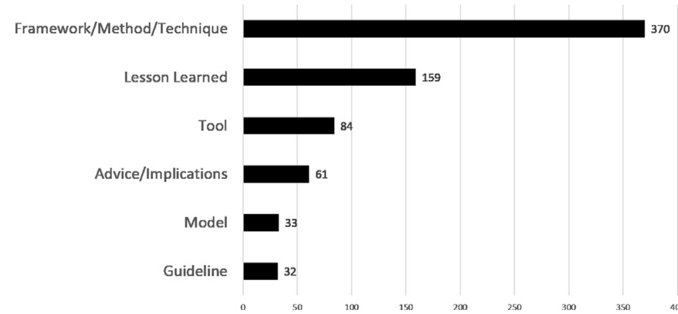

Fig. 14. Contribution types.

Research topic vs. contribution types. As we identified earlier, the most frequently addressed contribution type included frameworks, methods, and techniques (50%). These studies provided concrete solutions for the development of software-intensive automotive systems, and they were represented in all research topics. A similar distribution existed for lessons learned, which constituted 22% of the contributions. Lessons learned were also addressed in all research topics, but on a relatively smaller scale in a few research topics, such as system/software integration, software maintenance, and agreement support group.

Tools were a less frequently addressed contribution type (11%) and were distributed over almost all the research topics. However, it was evident that this contribution type was relatively less investigated in various research topics. Similarly, guidelines (4.5%) were distributed among various research topics. However, this contribution type was relatively less addressed in relation to others. More tools and concrete guidelines may be required, because these con-

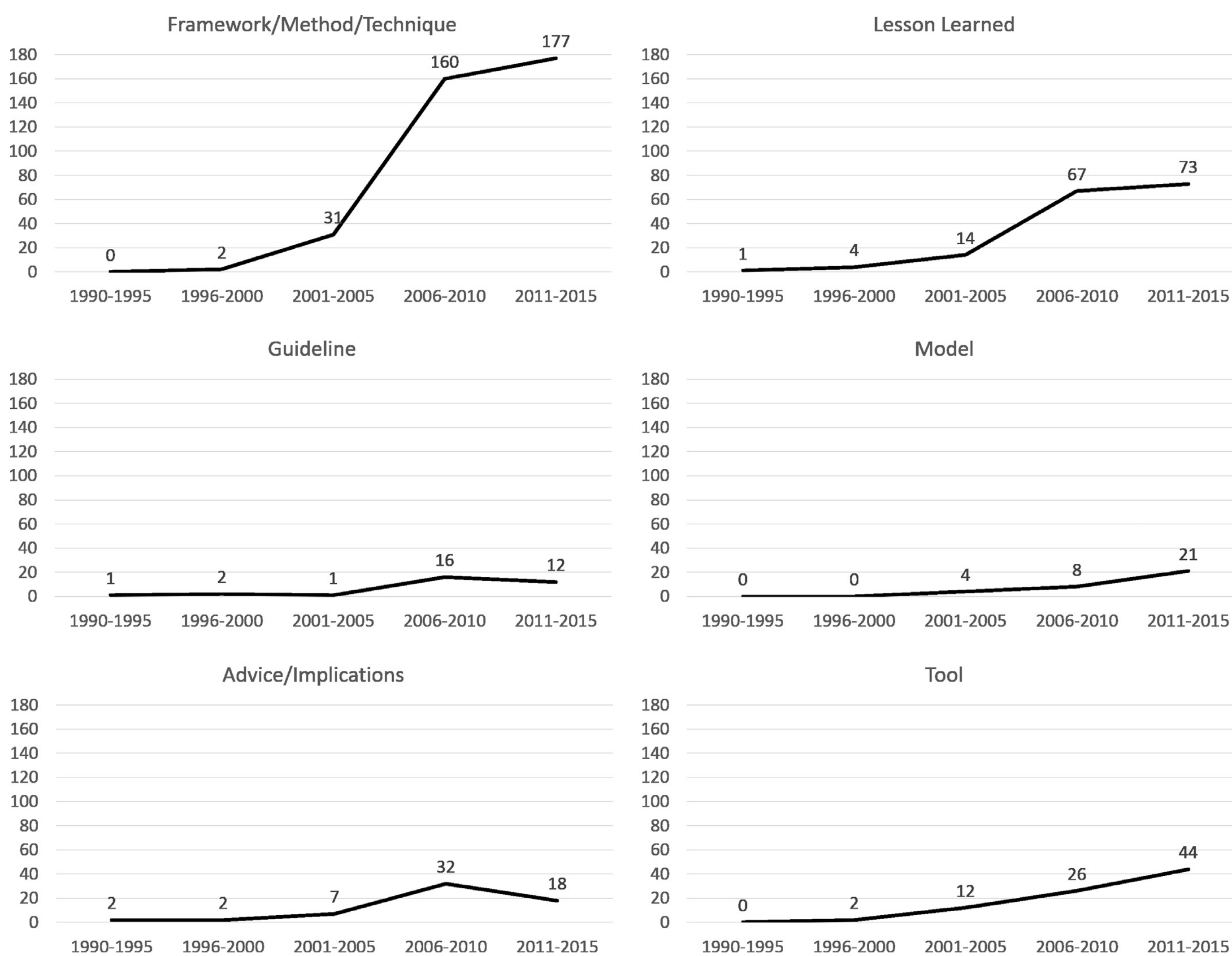

Fig. 15. Contribution types per year.

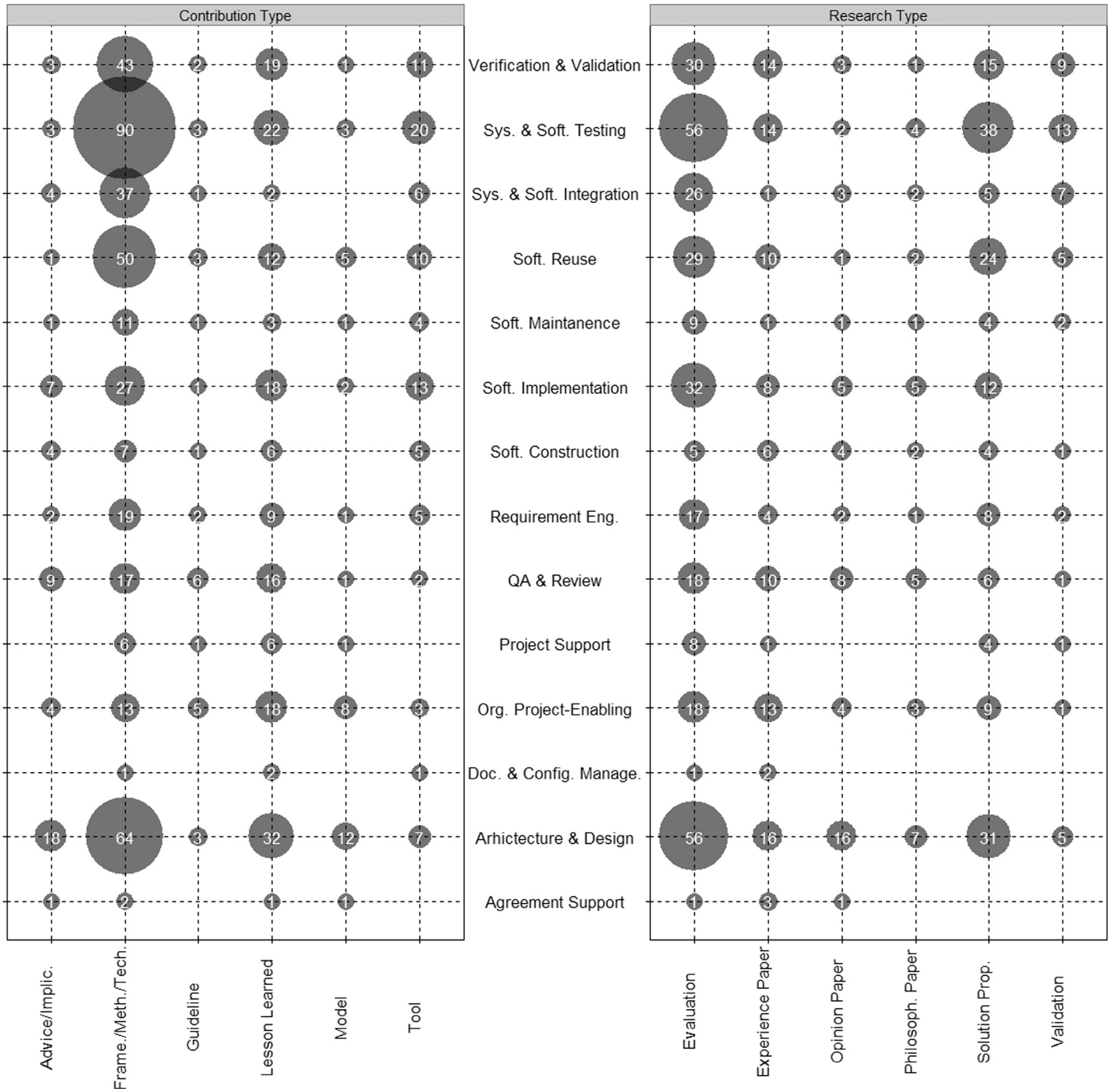

Fig. 16. Systematic map - research topic, research type and contribution facet.

tribution types are highly relevant and valuable from the point of view of industry.

Advice or implications (8%) were given in the majority of research topics, including discursive and generic recommendations based on personal opinions. In contrast, contributions in the form of models (4.5%) were rare and occurred in only a few research topics (i.e., system/software architecture, organizational support group, and software reuse). The scant use of this contribution type was evident in other research topics. Thus, additional contributions made by models, such as conceptual constructs and taxonomies, are required in the field because they support the understanding and comprehension of real-world phenomenon in an abstract manner.

4.5.2. ASE research roadmap: where do we stand today?

The two previous review studies by Pretschner et al. (2007) and Broy (2006) have investigated ASE characteristics and proposed a research roadmap for the area. The authors argue that there will be increasing growth in both the size and complexity of automotive software and future development will be driven by several factors, related to the increasing demand for innovation and improvement, higher quality, shorter time-to-market, and individualization, among many others. Consequently, the automotive industry faces several challenges, with impacts on their organization, key competencies, underlying processes and practices, maintenance, and long-term strategies. On the basis of existing challenges and future development of the automotive industry, the authors identified the following future research themes.

- Theme 1. Comprehensive automotive architecture
- Theme 2. Reducing complexity
- Theme 3. Improving development processes
- Theme 4. Seamless model-driven development
- Theme 5. Integrated tool support
- Theme 6. Learning from other domains
- Theme 7. Improving quality and reliability
- Theme 8. Standardized automotive software and system infrastructure

The need for a comprehensive automotive architecture has been acknowledged in the literature. To that aim, AUTOSAR has been introduced by the automotive industry to address existing challenges and establish an open and standardized architecture. Findings from the literature indicate that AUTOSAR is well-perceived and largely adopted in the industry, and supporting evidence for its positive effects is growing (Dersten et al., 2011; Martínez-Fernández et al., 2015). However, several challenges and limitations have also been identified that must be addressed in the future (Di Natale and Sangiovanni-Vincentelli, 2010; Heinecke et al., 2008; Giese et al., 2010; Dersten et al., 2011; Martínez-Fernández et al., 2015). The importance of seamless model-driven development is highlighted in the ASE roadmap. The authors argue that the everyday use and full potential of model-based approaches in the automotive industry is still limited due to the predefined set of isolated modeling theories and tool environments. To deal with this issue, a pragmatic, demand-driven, ad hoc approach has been taken by practitioners to adapt their engineering methods and processes to the available tool environments. In consequence, the practice is characterized by redundancy, inconsistency, and the lack of automation. These issues negatively affect productivity, the quality of the development process, and the automotive applications. Therefore, a comprehensive development methodology along with seamless model-driven development and integrated tool support are required. This issue was also acknowledged by several other scholars, and various solutions have been proposed in the literature. Furthermore there are several research initiatives[7] that aim to provide a unified automotive development approach and an integrated tool chain. Together with a comprehensive automotive architecture, seamless model-driven development and an integrated tool chain reduce the overall development complexity of automotive systems.

Broy (2006) also highlighted the importance of process quality improvement. He argues that development processes in the automotive industry are not yet effectively adapted to the needs of complex software-intensive systems. Thus, a deeper understanding of system/software development processes are required to support their improvement. Automotive SPICE was introduced to address the specific needs of the automotive sector and to develop a common framework for the assessment and improvement of development processes. There are several studies in the literature reporting on experiences with and the positive impact of Automotive SPICE in practice. Nevertheless, several challenges have been also reported in the literature, which need to be addressed in the future (Fabbrini et al., 2009).

Even though the automotive industry has its own characteristics, it also shares several similarities and challenges with other domains (i.e., avionics industry). The existing body of knowledge from other domains and from software engineering in general need to be systematically explored and adapted to the automotive industry. For instance, the avionics industry has a long tradition in reliability and quality management. There are also several emerging paradigms such as continuous delivery and deployment that are become more prevalent in the software industry. The existing surveys (Rodriguez et al., 2017; Mäntylä et al., 2015) indicate that such approaches bring several advantages and could be adopted in several application areas, such as embedded systems. This opens a wide range of new opportunities for the automotive industry and supports continuous value delivery, effective recall management, and rapid innovation based on real-time data analytics.

The importance of improving the quality and reliability of automotive systems is emphasized by the ASE roadmap. There has been great progress in the area and there are several contributions that aim to improve quality and reliability of software-intensive automotive systems. For instance, approaches to support architecture quality prediction and optimization (Meedeniya et al., 2011b; 2012a), automatic test-case generation (Choi and Byun, 2015; Pfaller et al., 2006; Caliebe et al., 2012; Matinnejad et al., 2015; Lakhotia et al., 2013), software reliability growth models (Rana et al., 2014a; 2013), defect prediction (Rana et al., 2014b; Hryszko and Madeyski, 2015), and optimal synthesis and deployment (Meedeniya et al., 2012b; 2011a; Nejati and Briand, 2014; Nejati et al., 2013; Aleti and Meedeniya, 2011; Aleti, 2015) are just a few examples of such contributions. Lastly, the authors argue that an important step for the automotive industry is to move from proprietary solutions to shared standards. There have been many initiatives, resulting in several standardized solutions in the automotive industry, for example, AUTOSAR, ISO-26262, and Automotive SPICE. Similarly, there are also several communities[8] and research initiatives[9] that aim to bring standardized solutions to ASE.

Findings from this review indicate that significant overall progress has been made in the area. There are several contributions in the literature that particularly address the above-mentioned research themes. Despite this progress, the research themes indicated by the ASE roadmap seem to be both valid and relevant, representing opportunities for future research. Looking at the existing body of knowledge presented in Fig. 16, it seems that the research on ASE is primarily driven by empirical evaluation studies. However, validation and comparative studies are less represented and the literature lacks practitioner-oriented guidelines for the selection of existing solutions, technologies, and practices.

5. Discussion

We conducted this review with the objective of characterizing the current state of the research pertaining to software engineering in the automotive industry. To achieve that aim, we performed a systematic mapping study and analyzed five aspects of the literature: research intensity, research topic, research type and method, contribution type, and potential research gaps. Although several reviews have been published of the software engineering literature, this study is the first attempt to provide a structured overview of the literature on software engineering in the automotive industry.

5.1. Overview of findings and their implications

RQ1. What is the intensity of research activity on ASE?. To address RQ1, we investigated the intensity of the research activity in the ASE literature with respect to several aspects, including publication trends over time, publication channel, most frequent publication venues (specific journals, conference proceedings, etc.), authorship information, and the number of citations. Findings from this review indicate that the area of ASE research has received much attention, particularly over the last two decades, and significant progress has been made in the area. Looking at the publica-

[7] Two example of such initiatives are ITEA AMALTHEA (https://itea3.org/project/amalthea.html) and ITEA AMALTHEA4Public (https://itea3.org/project/amalthea4public.html).

[8] For instance SAE Standards Development http://www.sae.org/standards/.

[9] For instance ITEA APPSTACLE research project https://itea3.org/project/appstacle.html.

tion trends, one sees that the research activity in this area continues to increase, showing stable growth. The majority of previous studies are published in conference proceedings and the literature on ASE is spread over several publication venues. The analysis of authorship information shows that the level of industry–academic collaboration is quite high in ASE research and practitioners have contributed significantly to the area.

We expect that research into ASE will continue to increase over time due to the increasing importance of software engineering in the automotive industry. There will be increasing growth in both size and complexity of automotive software, and future development will be driven by increasing demand for innovation and improvement of automotive functions, higher quality systems, shorter time-to-market, individualization, and many more factors (Broy et al., 2007; Pretschner et al., 2007). Consequently, the automotive industry will face several challenges, with an impact on their organizations, underlying processes and practices, and long-term strategies, among other impacts (Broy et al., 2007; Pretschner et al., 2007). These issues offer a wide range of new research opportunities and create demand for further investigations.

RQ2. What are the most frequently investigated research topics? How have they evolved over time? And in what application area have they been investigated?. To address RQ2, we determined the most frequently investigated research topics and their evolution over time. Overall, 14 recurrent research topics were identified in the relevant literature. These topics are diverse, which indicates the complexity and breadth of ASE research. Three areas including system/software architecture and design, system/software qualification testing, and software reuse were identified as the most frequently addressed topics. Even though the publication trends vary across identified topic, research activity of the majority of topics shows stable growth. Taking into the account the current trends and future development of the automotive industry as characterized by previous works (Broy et al., 2007; Pretschner et al., 2007), one can predict increasing research activities in the future, particularly in the area of quality management and testing, verification & validation, requirement engineering, architecture, reuse, maintenance, and integration of software-intensive automotive systems.

The analysis of automotive application areas reported in the primary studies indicate that the research on ASE was conducted in several different vehicle sub-systems. The body and comfort software, powertrain and transmission software, chassis and safety software, infotainment and telematics software, and infrastructure software were the most investigated areas, respectively. However, due to the limited contextual information reported in the existing studies, it was not possible to classify large number of primary studies (60%). Given the ongoing business changes and trends in the automotive industry, one can expect major changes in above application areas particularly in automotive infotainment and telematics systems. There are currently several studies in the literature that investigated various aspects of automotive infotainment and telematics systems. Few example of such contributions include purpose specific notation languages for the design of human-machine interfaces, techniques for the rapid development of automotive infotainment systems, over-the-air updates, architecture for automotive software ecosystem, and innovation experiment system. Furthermore, there are several industrial and academic initiatives investigating automotive infotainment and telematics systems (such as vehicle-to-everything (V2X) communication and ecosystem-driven automotive software development). Thus we expect that there will be more research opportunities and more studies published addressing those application areas.

RQ3. What are the most frequently applied research types and research methods?. To address RQ3, we investigated research types and research methods that were most frequently applied in the ASE literature. With respect to research type, our results showed that evaluation research and solution proposals were the most frequently used. This result was expected because in a new research area, problem identification and investigation require both novel solution development (solution proposal) and empirical evaluation (evaluation research) of those solutions. The findings from this review indicate that the number of evaluation studies has significantly increased over time. Similarly, there was slight growth in the number of validation studies. These figures have positive implications and indicate the improving scientific rigor of research in the area of ASE.

With respect to the research method, we identified the case study as the dominant research method applied in the literature. However, a large portion (51%) of the studies were not grounded in any research methodology. We also observed that many of the case studies reviewed did not report appropriate details as recommended by Runeson and Höst (2009), such as research protocol, data collection and analysis procedure, and contextual information. These limitations further hinder the future replication, generalization, and aggregation of evidence in the context of ASE. Overall, research activity in ASE area seems to have high industrial relevance but is relatively lower in its scientific rigor, as indicated by the large number of studies that were not based on a research methodology and the small number of studies that conducted validation research.

RQ4. What kind of contributions are provided by studies related to ASE?. To address RQ4, we investigated the most frequent contribution types in the literature. Our results indicated that framework, method, and technique constituted half of the existing contributions. However, lesson learned, guidelines, and models were relatively less represented in the literature. Thus, the findings indicate that research activity in ASE tends to be more technical and constructive-oriented rather than analytical and theoretical in nature. This was indicated by the large number of technically oriented contributions compared to the relatively small number of studies addressing lessons learned, guidelines, and models.

RQ5. What are the research gaps and promising future research directions related to ASE?. To address RQ5, we combined the answers to the previous research questions and performed combinational analyses to identify the less frequently investigated areas and the potential gaps in the body of research. Overall, we observed that research types and contribution types were not homogeneously represented in all research topics, which must be considered in future research as each research and contribution type brings its own value to the ASE community. Thereafter, we reviewed the ASE research roadmap proposed by Pretschner et al. (2007) and Broy (2006) and discussed how the existing works from the literature relate to the proposed research themes. The findings from this review indicate that several contributions in the literature addressed in particular the identified research themes, and significant overall progress has been made in the area. Despite this achievement, the themes identified in the ASE research roadmap still appear valid and relevant, and they represent opportunities for future research.

As acknowledged by previous studies (Pretschner et al., 2007; Broy, 2006), the automotive industry suffers from fragmentation of solutions and lack of standardized software and system infrastructure. There have been many initiatives, resulting in the introduction of several standardized solutions by the automotive industry, such as AUTOSAR, ISO-26262, and Automotive SPICE. Such contributions have had a positive impact on the automotive industry and have brought several advantages. However, many opportunities exist and more initiatives toward standardization of automotive software and system infrastructure are required. While the automotive industry has its own unique characteristics, it also shares several similarities and challenges with other domains (i.e., the avionics industry). Thus, the existing body of knowledge from similar domains and software engineering in general must be systematically explored and adapted to the automotive industry.

5.2. Implications

A systematic synthesis and classification of the literature on ASE is given in this work. Our main contribution is the logical organization of the knowledge of software engineering in the automotive industry. From the academic point of view, this work provides researchers with a structured body of knowledge and a basis for future studies. From the perspective of practitioners, the synthesized and structured body of knowledge will facilitate access to and the comprehension of state-of-the-art knowledge on ASE.

5.3. Limitations

Even though several possible limitations were taken into account and countermeasures taken during the design and execution of this systematic mapping study (see Section 3.7), our findings must be interpreted within its own limitations. The main limitation of this work lies in its coverage and whether it covers all existing knowledge and research articles in the literature on this subject. This is in practice relevant to all systematic mapping studies, particularly those reviews conducted on broad subject areas. As acknowledged in the literature (Wohlin et al., 2013; Petersen et al., 2015), despite the objective of systematic reviews to retrieve all relevant contributions, it is likely that only a sample of the relevant research articles will be obtained. Because of the scope of this study, our approach in this review was to capture as many relevant studies as possible with minimal noise in order to answer the research questions. One other limitation of the present study is the fact that this review is limited to only studies published in English, whereas there is a large amount of work on ASE in written in German. Furthermore, similar to other previously conducted reviews, this study relies on the academic literature as its main data source, yet several other valuable sources of knowledge exist. For instance, a large amount of research in the automotive domain is presented in patents. Similarly, large body of knowledge is encapsulated in the grey literature, which includes technical standards and technical reports, among many more examples. Despite these limitations and without any claim of completeness, this work represents the first attempt to provide a structured body of knowledge on ASE.

5.4. Recommendations for future research

Overall, the research on ASE has attracted much attention from both scholars and practitioners, and significant progress has been made, particularly during the last two decades. Each of the 14 identified recurrent research topics represents opportunities for future research. However, our findings included several high-level observations that might guide future research in this area.

The research on ASE seems to have high relevance to industry, which was evident in the large number of empirical evaluation studies (43% of the primary studies) and the large number of industrial experience papers (15%). However, there was considerably less validation research, that is, experimental studies (7%), in the literature. Surprisingly, a large number of primary studies were not grounded in any research methodology (51% of primary studies). However, there were many solution proposals (22%) as well as framework, methods, and techniques (50% of the total contributions) in the area. Although this result has positive implications for research in this field, from a practical point of view it raises a question with respect to the applicability of the existing solutions (i.e., which approach is more appropriate in what context and why).

Thus, rigorous scientific studies that conduct the systematic exploration, validation, and evaluation of the existing knowledge are required to provide guidelines for the selection of existing solutions, technologies, and practices. For instance, several approaches to the synthesis and deployment of automotive software functions were proposed in the literature. However, critical comparison of these approaches, which guides the selection of existing solutions in practice, is missing. This holds true for many other research topics wherein the research activity tends to be constructive (i.e., development of new solutions) rather than analytical (i.e., critical validation and comparison of existing approaches). Perhaps a good example of such a study is that of Rana et al. (2014a), in which the authors critically evaluated existing standard reliability growth models used in practice and provided a guideline for the selection of existing approaches in the context of automotive software systems.

Similarly, by examining the contribution types addressed in the literature, we showed that most of the contributions were technological in nature (i.e., frameworks, methods, and techniques) rather than analytical and theoretical constructs (i.e., lessons learned, models, and guidelines). This is typically expected in technical areas such as the automotive domain. However, further analytical and theoretical contributions are required to advance the research and support the engineering of future complex automotive systems.

Overall, the literature on ASE is rich and encompasses a wide range of contributions and empirical evidence, offering great opportunities for the automotive industry to move toward evidence-based practice (Kitchenham et al., 2004). In light of the evidence-based software engineering (EBSE) paradigm (Kitchenham et al., 2004), several reviews have classified, aggregated, and analyzed the existing knowledge in a particular topic area. Although many systematic literature reviews exist in the software engineering body of knowledge, systematic aggregation and critical analysis of domain-specific knowledge and evidence are required to support the industry. This is particularly relevant to the automotive software industry, where the unique characteristics, constraints, and requirements require individual solutions that take advantage of domain specificities. Perhaps a good example of such a study is that of Kasoju et al. (2013), wherein the authors analyzed the automotive testing process using EBSE.

6. Concluding remarks

The main objective of this research was to characterize the body of knowledge pertaining to software engineering in the automotive industry. To achieve that end, we conducted a systematic mapping study in order to identify primary studies that were relevant to our scope and extracted and synthesized data from the existing contributions. From an initial set of 4348 articles retrieved from multiples sources, we identified 679 primary studies that were relevant to the scope of our study. These primary studies carefully investigated and they were classified and analyzed with respect to five different dimensions. The most important findings of this review are summarized below according to the research questions.

- **RQ1. What is the intensity of research activity on ASE?** The research on ASE has received much attention, particularly over the last two decades, and significant progress has been made in the area. Looking at the publication trends, one can observe that research activity in this area continues to increase, and its evolution has shown stable growth. The majority of primary studies were published in conference proceedings and the literature on ASE is spread across various publication venues. The analysis of authorship information shows that the level of industry–academic collaboration is quite high in ASE research and practitioners have contributed significantly to the area.

- **RQ2. What are the most frequently investigated research topics? How have they evolved over time? And in what application area have they been investigated?** Fourteen recurrent research topics were identified in the ASE literature. These topics are diverse, indicating the complexity and breadth of ASE research. Three areas, namely, system/software architecture and design, qualification testing, and software reuse, were identified as the most frequently addressed topics. Even though the publication trends vary among the identified topics, the research activity on the majority of topics indicates stable growth. The analysis of automotive application areas shows that body and comfort software, powertrain and transmission software, chassis and safety software, infotainment and telematics software, and infrastructure software were the most investigated areas, in that order. However, large number of primary studies (60%) did not report any specific automotive application areas.
- **RQ3. What are the most frequently applied research types and research methods?** Evaluation research and solution proposals were the most frequent research type. However, comparative studies and validation research were relatively less common in the literature. Looking at the applied research methodology, the results show that case study research was the dominant research method applied in the literature. However, a large number of primary studies (51%) were not grounded in any research methodology at all. We have also observed that many of the case studies in our review did not report proper details as recommended by previous guidelines. These limitations hinder the future replication, generalization, and aggregation of evidence in the context of ASE. Overall, findings from this review indicate that research activity on ASE seems to have high industrial relevance but is relatively lower in its scientific rigor.
- **RQ4. What kind of contributions are provided by studies related to ASE?** Taking contribution type into the account, our findings suggest that the research activity in ASE tends to be more technical and constructive oriented rather than analytical and theoretical. The majority of existing contributions were in form of framework/method/technique, whereas other contribution types such as lessons learned, guidelines, and models were relatively less common in the literature.
- **RQ5. What are the research gaps and promising future research directions related to ASE?** Each of the fourteen identified research topics presents opportunities for future research. We have found that research types and contribution types were not homogeneously addressed in all research topics, which should be considered in future research as each of these brings its own value to the ASE community. There are several contributions in the literature specifically addressing previously identified research themes proposed by the ASE research roadmap. Yet, despite this progress, research themes proposed in the ASE roadmap still appear to be valid and relevant, and they represent opportunities for future research. The findings from this review indicate that validation and comparative studies are less represented and the literature lacks practitioner-oriented guidelines. Thus, rigorous scientific studies that conduct systematic exploration, validation, and evaluation of the existing knowledge are required to provide guidelines for the selection of existing solutions, technologies, and practices.

This work presented a structured review of the ASE literature, highlighted potential research gaps, and provided recommendations for future research in the area. In future work, we intend to compare and synthesize the literature with respect to the state-of-the-practice by conducting a survey of the automotive software industry.

Acknowledgment

This research has been carried out within the ITEA 3 APPSTACLE research program (https://itea3.org/project/appstacle.html), and it has been funded by Tekes (the Finnish Funding Agency for Technology and Innovation).

Appendix A. Data extraction protocol

Table A8
Generic data properties (DP1–DP7), research topic (DP8), automotive application area (DP9).

Title	Description
Publication Year	Indicates the publication year of the primary study.
Publication Channel	Indicates publication type, classified as journal, conference proceedings, workshop, or book chapter.
Target Venue	Indicates title of journal, conference proceedings, workshop, or book chapter.
Citation Count	Indicates the citation count retrieved from Google Scholar (as of 11 Jan 2016)
Author Country	Indicates name of country extracted from authorship information.
Author Affiliation	Indicates authors affiliation extracted from authorship information.
Contributors Type	Indicates contribution type, classified as 1) a paper having all academic authors, 2) all industry authors, or 3) collaborative work among academic and industry authors.
Research Topic	Indicates the contribution of the primary study with respect to the software engineering processes stated in ISO/IEC 12207 PRM
Application Area	Indicates the automotive software application area reported in the primary studies. The classification schema was adopted from Pretschner et al. (2007) and Harris (2013) and includes various types, such as 1) body and comfort software, 2) driver assistance software, 3) chassis and safety software, 4) powertrain and transmission software, 5) infotainment and telematics software, 6) infrastructure software, and 7) not applicable. The detailed description of these application areas is presented in Harris (2013), Pretschner et al. (2007).

Table A9
Research type (DP10) (adopted from Wieringa et al., 2006).

Title	Description
Solution Proposal	A solution to a particular problem is proposed, which can be either novel or a significant extension of an existing solution. The potential benefits, drawbacks, and applicability of the proposed solution are presented via a small example, demonstrators, or through argumentation.
Validation Research	A novel solution or technique is proposed but not yet implemented in practice. The proposed technique is investigated by using a methodologically sound research approach, such as an experiment, prototyping, formal analysis, simulation, or similar approaches.
Evaluation Research	A particular solution or technique is implemented in practice (solution implementation), and its evaluation is conducted accordingly (implementation evaluation). The applicability of the solution or technique and its consequences in terms of benefits and/or drawbacks are evaluated and demonstrated in practice.
Philosophical Paper	These studies present new ways of looking at existing things by structuring a particular area, proposing a theoretical framework, or providing a taxonomy.
Opinion Paper	These studies reflect the personal opinion of the author regarding whether a particular technique is good or bad and/or how it should be applied. These papers typically do not rely on related work or a research methodology.
Experience Report	These studies report industrial experiences and explain what and how something has been done in practice. Typically, these studies reflect the personal experiences of the author in practice, and they do not mention any research methods explicitly.

Table A10
Research method (DP11).

Title	Description
Case Study	This definition is borrowed from Wohlin et al. (2012): “The case study is an enquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between the phenomenon and its context cannot be clearly specified.”
Experimentation	This definition is borrowed from Wohlin et al. (2012): “It is an empirical enquiry that manipulates one factor or the variables of the studied setting. Based on randomization, different treatments are applied to or by different subjects. Other variables are kept constant, and the effects of the variables on the outcome are measured.”
Survey	This definition is borrowed from Wohlin et al. (2012): “A survey is a system used to collect information from or about people to describe, compare or explain their knowledge, attitudes and behavior.”
Design Research	The primary study states this research methodology explicitly.
Action Research	The primary study states this research methodology explicitly.
Grounded Theory	The primary study states this research methodology explicitly.
Systematic Reviews	The primary study states systematic literature review or mapping study explicitly.
Mixed Methods	The primary study follows more than one research methodology.
Not Applicable	The primary study does not mention any research methodology, and it cannot be derived from or interpreted in the paper.

Table A11
Contribution type (DP12) (adapted from Shaw, 2003 and Paternoster et al., 2014).

Title	Description
Framework/Method/Technique	The contribution of the study is a particular framework, method, or technique used to facilitate the construction and management of software-intensive systems.
Guideline	A list of advice or recommendations based on synthesis of the obtained research results.
Lesson Learned	The set of outcomes directly based on the research results obtained from the data analysis.
Model	The representation of an observed reality in concepts or related concepts after a conceptualization process.
Tool	A technology, program, or application that is developed in order to support different aspects of software engineering.
Advice/Implication	A discursive and generic recommendation based on personal opinion.

Appendix B. Overview of research topics

This appendix presents a short overview of recurrent research topics in ASE literature. Given the limited space and large of number of primary studies, the studies cited in this appendix are represented by an ID [e.g. P100, P530], while further information about them is available on the web.[10]

B1. Agreement processes (AGR)

Agreement support group (5 studies). The primary studies on this research topic focused on different aspects of software acquisition and agreement management activities, ranging from acquisition models and contract management to project control and documentation [P87, P245, P259, P267, P447].

B2. Organizational project-enabling processes (ORG)

Organizational project-enabling support group (48 studies). Several primary studies in the literature focused on software development life-cycle models, including agile methodology and the V model [i.e. P141, P53, P11, P354, P146, P292, P210, P262]. Numerous extensions of V model for different purposes were proposed: to improve variability handling and reuse during the development process, to improve management of safety and reliability, or to improve the integration and communication of organizational units. The concept of process assessment and improvement is relevant and central to the automotive industry because of the complex development processes, large supplier chain, domain regulations, and difficulty in managing quality attributes such as reliability and safety. Therefore, a systematic approach to the assessment and improvement of development processes is essential because it provides a means to determine capability and improve organizational processes.

Several primary studies [P338, P417, P418, P592, P677, P653, P267] investigated software process assessment and improvement in the automotive industry. These studies focused on various aspects that are relevant to Automotive SPICE, such as recommendations for improving the assessment process, supplier acquisition, and quality control activities. A few studies also reported its challenges, including the validity and repeatability of assessment results, its applicability in the context of agile methodologies and software product-line, and difficulties associated with the quality control of suppliers. Although the value of software process assessment is highlighted in the existing works, it is also identified as an expensive and time-consuming activity, particularly for small to medium sized enterprises. To deal with this issue, a few lightweight methods [P55, P131] have been proposed to support continuous assessment and improvement of software processes that are tailored for agile methodologies in the automotive industry. A few primary studies [P28, P40, P142, P413] investigated the importance of aligning organizational functions and the difficulty of establishing effective coordination and communication among various units. These studies focused on the integration of the system and software engineering activities, such as interactions between the software development department and manufacturing.

The importance of software reliability and its implications for the effective allocation of resources, product quality improvement, and evaluation of release readiness is highlighted in the literature [i.e. P212, P213, P170, P396, P72, P171, P368, P161]. Many primary studies on this topic focused on software reliability growth models (SRGM), which involve modeling and studying defect inflow profiles and the reliability of software systems. For instance, [P396, P212] investigated common SRGMs to evaluate their ability to support the reliability of embedded software development. The results indicated that the Logistic and Gompertz models were the most accurate and that the expected shape of defect inflow might be a good indication in the selection of appropriate SRGMs. The authors also presented recommendations for successful application of SGRMs in the embedded domain. In another study, [P72] investigated six SRGMs in four industrial automotive projects to assess the relevant criteria for the selection of proper distribution. The results showed that beta distribution provided the best fit for the defect inflow data in the projects with different characteristics. The authors also argued that understanding the distribution of defect inflow not only helped in applying proper statistical methods for analysis but also supported the proper selection of SRGMs for reliability modelling. The findings of another empirical study [P213] showed that certain parameters of SRGMs need to be adjusted in order to provide accurate predictions of reliability. The authors presented recommendations for how and where the adjustment in the models should be made and used during development processes. Furthermore, an overview of software defect prediction methods, their applicability in various stages of software development, and their characteristics, advantages, and limitations was reported in the literature [P171, P139]. The remaining studies focused on various miscellaneous subjects such as software certification and tool qualification [P462, P618, P353, P567, P612], and defect classification based on IEEE standard 1044 [P20].

B3. Project processes (PRO)

Project support group (14 studies). The primary studies on this research topic focused on software project planning, size measurement, and the cost estimation of automotive systems [P32, P214, P244, P358, P406, P407, P78, P538, P645, P644]. A few concrete recommendations and techniques are proposed in the literature that aim to improve size measurement and cost estimation of software-intensive automotive systems. A few other primary studies particularly investigated challenges relevant to cost estimation and size measurement of automotive software product families [P34, P71, P350, P596].

B4. Technical and software implementation processes (ENG/DEV)

System/software architecture and design (131 studies). A large number of studies on this research topic discussed AUTOSAR, demonstrated its use in building various automotive applications [i.e. P136, P29, P30, P422, P144, P188], and explained its limitations and effects on the development process and product quality [P198, P43]. The findings from an existing systematic literature review [P198] show that although AUTOSAR has several benefits, such as efficient development and short lead time, it also has several costs, such as performance risks, a learning curve, and the need to deploy new development processes in the organization. The findings from another survey [P43] indicate that the most popular benefits were standardization, reuse, and interoperability, whereas the most important drawbacks were complexity, the initial investment, and the learning curve. To mitigate these drawbacks, survey respondents suggested a comprehensive tool environment and increasing the stability of releases in order to decrease the cost of updates and migration among versions. Several studies in the literature also investigated the limitations of AUTOSAR architecture [i.e. P330, P405, P309, P198, P43]. For instance, runtime adaptation and dynamic integration of new software components were investigated, and several solutions have been proposed [i.e. P334, P474, P473, P193, P178, P453].

Several primary studies investigated the challenges relevant to architecture modeling, including both logical and technical archi-

[10] Link to the online repository is provided in Findings section.

tecture. The main aim of architecture modeling is to provide a graphical representation of the software system at different abstraction levels to support the design and construction of the system. The particular challenges relevant to the model-driven architecture of automotive software systems was discussed in [P408], which highlighted the central importance of separating logical and technical architecture. This separation further reduces the complexity and dependency of automotive functions to the physical layer and enables conceptual reuse, fast design, modeling, qualification testing, and integration of automotive systems [P408]. Various architecture description and modeling languages, such as EAST-ADL, SysML, AADL, and MARTE, are proposed or mentioned in the literature. Several primary studies reported experiences with and applications of architecture description languages in practice [i.e. P286, P196, P187, P498, P89]. For logical architecture, a domain-specific architecture definition language for supporting abstraction, capturing cross-cutting functionalities, and coordination of services in the context of service-oriented architecture is highlighted as a future research challenge [P408]. Furthermore, the management of the various levels of granularity and detail is required for the effective collaboration of OEMs and suppliers along with tool-supported exchange of interface models [P408]. From a technical architecture viewpoint, integration of models in order to abstract and analyze communication behavior, safety, and reliability analysis as well as software diagnosis is required [P408]. Furthermore, the integration of technical architecture models to support system engineering, such as the evaluation of architectural decisions, in addition to a well-established tool chain and seamless model-based environment, require further investigation [P408].

Several studies also investigated architecture quality of automotive systems, including their evaluation, analysis, and improvement. A wide range of contributions are available in the literature. A few examples of such studies include quality prediction and architectural optimization of automotive systems [P379, P92, P366], safety-oriented reference architecture [P386], and metrics and approaches for quality assessment of automotive architecture that supports architectural decisions and quality improvement with respect to maintainability, reliability, and stability [P194, P291]. Few studies particularly investigated challenges relevant to the architecture of automotive infotainment system [i.e. P641, P658, P662], as well as approaches for the over-the-air updates [i.e. P619, P636, P132]. Several other miscellaneous subjects, such as development and maintenance of automotive reference architecture [P129, P218, P35], architecture for supporting federated embedded systems and ecosystems [P90, P276, P88], and architecture for innovative experimental systems [P211, P91], were also discussed in the literature.

Several primary studies focused on difficulties related to the detailed design of automotive applications. The detailed design of automotive applications is mainly associated with the development of graphical models that specify functionalities at a detailed level [P408]. The main challenge here is precisely defining, capturing, and modeling functional and non-functional requirements to allow effective construction and quality assurance of automotive systems. These graphical models could be classified as existential (model-as-specification) and universal (model-as-implementation) models [P408]. The specification models are mainly used as a communication medium and representations of requirements in various forms. However, implementation models aim to represent the detailed functional and behavioral specifications of a particular function. The specification models are typically developed by using UML, whereas other proprietary notations, such as Matlab Simulink and Stateflow, are commonly used in the design of implementation models [P408]. The particular challenges and open issues relevant to software detailed design of continuous, hybrid, and discrete systems are highlighted in [P408]. In the area of software design research, many primary studies addressed in particular challenges in the specification, modeling, and analysis of the temporal properties of automotive functions [i.e. P232, P319, P327, P454, P456, P455, P137, P421, P290, P453]. Furthermore, various purpose-specific notations (such as CHARON) and UML extensions (such as UML Profile for Human Machine Interface Applications) have been proposed for the design of automotive systems [i.e. P254, P189, P26, P482, P478].

System/software qualification testing (127 studies). The challenges related to system and software qualification testing in the automotive industry arise from the unique characteristics of the domain, the complexity of automotive systems, and the need for strict safety, performance, and reliability. To deal with these challenges, sophisticated testing practices are required at different levels in order to ensure the compliance of the work product with respect to the expected functional and non-functional requirements. System and software qualification testing is the second most frequently addressed research topic in the literature. The test level and the test scope vary among the primary studies, and both functional and non-functional testing are discussed at various levels, including components, integration, and system-level. The majority of studies targeted functional or correctness testing, and non-functional aspects such as reliability, safety, and performance were also discussed.

Model-based testing (MBT), which relies on behavior models for the generation and execution of test cases, was widely discussed among our primary studies. The study [P339] investigated the effectiveness of MBT in practice and finding indicates that automatically and manually derived model-based test suites were much better at detecting requirement errors than hand-crafted test suites. Several primary studies in our review investigated automatic generation, selection, and prioritization of test cases [i.e. P163, P175, P393, P98, P33, P344, P109, P375, P16, P430, P314, P113]. Even though several techniques and approaches are proposed, the common objective was to automatically generate and execute test cases, taking into account the test coverage objective and cost constraints. The challenges relevant to qualification testing of continuous behavior and hybrid systems are particularly highlighted in various studies. Existing works [P359, P431, P201, P429, P393, P294, P151] proposed various approaches for automated testing of software functions developed by Matlab Simulink and Stateflow models.

Several studies also investigated in particular qualification testing of non-functional requirements of automotive applications such as reliability, safety, and performance. For instance, several studies [P411, P265, P224, P195, P500, P263] proposed a fault injection method, which can be used in the assessment of system dependability in the case of faults and in analyzing systems fault-handling capabilities based on a fault model. The difficulties related to the qualification testing of software product lines are also acknowledged in the previous works. Testing each separate product in a software product line is not feasible because there is an enormous range of products and variants of products. Thus, previous studies [P310, P304, P343, P602] proposed approaches to derive small sets of test cases that support the correctness of the entire product family.

Several primary studies focused on aspects related to test specification and notations [P166, P441, P438, P63, P268, P439, P140, P38]. For instance, [P166, P439] presented an extended version of TTCN-3, which is a standardized test technology that originated in the telecommunication industry to support the test specification of continuous and hybrid systems. Similarly, [P438] highlighted the limitation of TTCN-3 with respect to the support of real-time properties and extended it to support the specification of temporal aspects. The challenge with respect to the exchangeability of test specification among different teams and test systems is also highlighted in the literature. Difficulties arise in different test languages where various syntax, semantics, data formats and interface descriptions are used in practice [P441]. To overcome this challenge, several solutions, such as formal and test platform-independent definitions [P38] as well as the test exchange language TestML [P441] are proposed in the literature. In addition, [P268, P140] proposed multi-level test cases and focused on a single test specification and implementation that could be reused throughout the entire test process.

There are several studies in the literature that discuss about challenges relevant to the qualification testing of automotive infotainment systems. These studies particularly investigated usability testing [i.e. P661, P640, P668], as well as test automation and MBT of infotainment systems [i.e. P642, P659, P654, P647, P657]. Findings from an existing systematic literature review [P76] indicate 26 individual challenges and 15 solutions to improve automotive software testing. The challenges related mainly to requirements, test management, and automation and solutions related to requirements management, competence management, quality assurance and standards, test automation and tools, agile incorporation, and test management. Findings from another survey [P440] on the state-of-the-practice show that although automated test execution is commonly used in industry, automated test generation is not prevalent because of the limitations of the tools currently in use. Several other subjects were also covered, such as the human and social dimensions in relation to software testing [P484], parallel execution of loosely coupled segments of test scenarios [P594], and the alignment of requirement specifications with software testing [P428, P143, P283].

Software implementation (62 studies). Contemporary automotive software-intensive systems are implemented based on and in compliance with several standards and technologies. An overview of the commonly used technologies, practices, standards and methodologies in the automotive industry is presented in the literature [P205, P70, P412, P409]. Model-based software engineering (MBSE) is a predominant development methodology used in the design and development of software-intensive automotive systems. MBSE involves the development and exploitation of graphical models used to represent domain and design knowledge at different levels of abstraction throughout the entire development process [P408].

MBSE was recognized as technologically competent for automotive software development because it makes the development process more effective and improves the quality through early simulation, fault detection, and automatic generation of source code [P408, P416]. While MBSE has been identified as a possible solution to many of the challenges of ASE, everyday use of MBSE in the automotive industry remains limited, and its potential has not been fully exploited because of a predefined set of isolated modeling theories and tool environments [P391, P408]. Therefore, a comprehensive development methodology with a well-established modeling theory, common product model, proper process model, and an integrated tool chain are required [P408]. This issue was also acknowledged by several other scholars, and various solutions were proposed [P115, P392, P410, P128, P391, P477, P524, P526]. For instance, [P273] proposed a domain specific language that aimed to resolve inconsistencies with interface specification and integration within heterogeneous and distributed development environments.

Previous studies also reported on building a model-based development platform by integrating various modeling languages, available technologies, and tools [P585, P402, P24, P66, P297, P51, P524]. The common objective was to establish a model-based development flow that incorporated heterogeneous technologies in a complementary way in order to simplify the entire automotive software development process. To shorten the development cycle, existing studies [i.e. P54, P228] investigated challenges in the deployment of agile methodologies in the automotive industry. The results indicate that the application of agile practices shortened the in-house development cycle and, combined with the virtual test environment, relaxed the dependency of software products on mechanical components. One possible future research direction in this area is the investigation and adaptation of contemporary software engineering practices such as continuous integration and continuous deployment in the automotive industry. In addition to these subjects, several studies investigated challenges relevant to the management of safety and reliability [P222, P83, P23, P102], and adoption of MBSE in practice [P177, P67, P624]. The study [P651] proposed a reference model for model-based development of in-vehicle infotainment systems.

System/software integration (44 studies). The main concern in the integration phase is the effective synthesis and deployment of automotive functions without compromising quality attributes, such as performance, resource consumption, and safety. The integration phase is particularly challenging because of the distributed nature of automotive software and the complex interaction and dependency of functions with each other, and with the shared physical layer (i.e., communication channel). Findings from previous studies [P309, P452] indicate that syntactical and technological challenges are typically considered during the design phase by using a standardized architecture (i.e., AUTOSAR), whereas other demanding integration issues are often handled late in the integration phase.

The majority of primary studies under this subject area focused on the component deployment problem [P183, P111, P176, P157, P179, P112, P217, P25, P476, P367, P93, P475, P322, P158, P295, P96, P256, P287, P56], which concerns the correct deployment of software components into the physical layer. The deployment procedure must take into account several aspects, such as physical resources, deployment architecture, quality attributes, and constraints. Thus, the system integrators deal with several challenges and trade-offs with regard to the software component mapping to the hardware resources. This activity is crucial, and due to its complexity, several studies proposed automated and search-based approaches [i.e. P183, P158, P295, P256, P322]. Even though various approaches and mechanisms have been proposed in the literature, the common objective was to identify the near optimal and correct deployment of software components such that they fulfill the expected functional and non-functional requirements. For instance, performance optimization and minimization of computational demands [P217, P295, P256, P96, P452, P56], as well as reliability and safety-driven deployment optimization [P367, P93, P629, P112] were particular focuses of the primary studies.

The challenges relevant to the synthesis of temporal properties and guaranteeing end-to-end deadlines were discussed in several primary studies [i.e. P466, P44, P127, P248, P313, P362, P287]. Furthermore, several other miscellaneous subjects such as overviews of current practices and challenges in the integration phase [P405, P181, P309], certified software integration [P236], and domain-controlled architecture for large-scale integration [P192] were discussed in the literature. It is acknowledged in the literature [P309] that although several approaches have been proposed, the synthesis and deployment of automotive systems remain challenging because of the lack of a comprehensive integration solution that addresses all integration issues.

System/software requirement engineering (35 studies). Requirement engineering (RE) has a significant implication on the entire development process and comprises several activities including elicitation, analysis, documentation and validation. Findings from literature [P250, P408, P670] shows that RE in automotive industry is often inadequate and tends to be ad-hoc in practice. Challenges include the lack of a comprehensive requirement model and deficits related to the systematic capture of requirements, modeling of large amount of requirements, modeling of non-functional characteristics, requirement volatility, and tool support [P408]. Thus, sophisticated approaches are needed to support effective elicitation, specification, analysis, and validation of requirements in the automotive industry. A few recommendations and practical guidelines such as [P250, P670] are available in the literature and practitioners can benefit from these.

Traceability, effective tool support, elucidation of a comprehensive requirements model, systematic capturing requirements, and modelling large numbers of requirements have been identified as relevant research challenges [P408]. To fill these gaps, several methodologies and mechanisms are proposed in the literature that aim to support requirement management, foster automatic analysis, validation, and transformation of requirements. For instance, several studies [i.e. P324; P326; P459] proposed approaches to organize the requirements in a concise and manageable way. These studies attempted to reduce the complexity of the requirements modelling by separating and structuring the requirements into different categories. Previous studies [P231, P504] also investigated inter-dependencies of automotive functions and findings indicate that at least 85% of the automotive functions were interdependent. These studies highlighted the need for new methodologies and approaches that consider automotive requirements dependencies.

Natural-language requirements specification, which is still common in practice, is more prone to misunderstanding and misinterpretation for several reasons [P36, P408]. While structured natural-language specification has some advantages, it does not easily support automated analysis, verification, and transformation of requirements. For that reason, the automotive industry need to migrate from natural language-based to model-based RE [P408]. To address these issues, several studies in the literature investigated systematic transformation and verification of requirements throughout various stages of requirements evolution [i.e. P13, P27, P230, P373, P459]. The literature also emphasized the importance of traceability across requirement artifacts and the rest of development process. For instance, existing studies particularly emphasize the importance of aligning requirement engineering and software testing and propose approaches to link test scenarios and requirement specifications [P428, P143, P283]. A few additional miscellaneous subjects, including systematic elicitation and quality assessment of requirement use-cases [P432, P60, P172], variability analysis and management in requirement phase [P79, P220, P13, P520] and management of non-functional requirements, such as safety and security [P258, P395, P444], were also discussed in our primary studies.

Software construction (22 studies). The primary studies under this research topic focused on approaches to support automated code-generation for automotive systems [i.e. P663, P6, P522, P2, P106, P675]. For instance, [P107] reported experiences at the Ford Motor Company in the automatic code generation of powertrain applications. The study provided a list of requirements as the baseline for automatic source code generation in the automotive industry and elaborated various aspects, such as control flow, data calculation, data variables, standards, legacy and external codes, resource usage, tool requirements and tool characteristics. Furthermore, techniques and mechanisms that increase quality and safety of automatically generated source code were widely discussed in the literature [i.e. P535, P345, P249, P99, P152, P186, P505, P296, P68] and an overview of their contributions is given in software quality assurance and review research topic. Few primary studies particularly focused on development of human-machine interfaces. Example of such contributions include graphical toolkit for adaptive layouts [P671, P648], design patterns for rapid construction of infotainment systems [P434], and automatically generating user interfaces for dynamic services [P663]. Moreover, a few primary studies investigated component-based software engineering, such as the migration of legacy code to the component-based paradigm [P154], as well as a benchmarking framework that allowed the comparison of distinct component implementations [P48].

Software maintenance (18 studies). Because of the long lifetime of vehicle systems, difficulty in recall management, high field-maintenance costs, and other unique characteristics of the automotive domain, the maintenance of automotive systems is demanding. Today, vehicle systems are primarily maintained by vehicle dealers, who follow prescribed semi-automated procedures [P408]; however, advanced approaches to maintenance of automotive systems, such as online-diagnostics and over-the-air updates, are expected to become more prevalent in the near future. To support the maintenance of automotive software, several re-engineering and program comprehension approaches [i.e. P400, P499, P75, P364, P472] are proposed in the literature. These approaches aim to support the maintenance of automotive software artefacts such as Simulink models and source code.

The concept of clone detection and management, particularly in model-based environments, is also investigated in the previous works [i.e. P73, P398, P299]. Findings from the literature show that identification and management of clones not only improves the maintenance and quality of software products but also fosters the identification of reusable components [P299]. However, clone identification and management of large-scale automotive models is not straightforward in practice. For instance, [P299] focused on practical challenges and possible solutions in the application of model clone detection in a large scale industrial project at the BMW Group. Specifically, the study investigated the challenges associated with the improvement of scalability, relevance and tool-support in clone management in the context of model-based development [P299]. In another study, [P398] presented a model-clone detection approach to supporting the identification of semantic clones in which a similar behaviour was implemented with different structures and syntax. The approach was based on the structural normalization of data flows in Simulink models, and it used graph transformation to identify equivalent model fragments with different syntactic structures [P398]. Similarly, [P73] presented a case study of the application of pattern mining and clone detection at General Motors. The study [P73] reported the early experimental use of SIMONE, a model clone detection tool used for the extraction of identical and near-miss clones in Simulink models. The clones were further classified based on their size and similarity. SIMGraph (a SIMONE extension) was used to support visualization and comprehension of Simulink clones and patterns [P73]

The challenges and difficulties associated with software diagnosis in the automotive industry were also investigated. Findings from the literature indicate that the automotive industry lacks a comprehensive error diagnosis model and a systematic error recovery and treatment model [P408]. Failure management, failure logging, and comprehensive error models that ensure fail-safe, graceful degradation and error prevention by supporting redundancy were highlighted as relevant research problems [P408]. A few studies investigated remote recall servicing and the proposed architecture for diagnosis as well as for safe and secure remote code updates [P542, P132, P619, P636]. As acknowledged in the literature, although the proposed approaches might not be applicable to critical issues, it could largely facilitate vehicle recall management of non-critical software issues. The study [P488] focused on the difficulties related to automotive software diagnosis from the point of view of data analytics. The authors argued that software diagnosis was a significant economic factor because legal requirements demanded 15 years of after-sales services and support. The study suggested that a methodology based on multivariate split analysis would support the automatic and effective discovery of symptoms and fault patterns among large-scale diagnosis data. Moreover, several other miscellaneous subjects such as automatic documentation generation based on model-driven techniques [P57], modularity assessment of Simulink models [P401], and measures to support the maintenance of software architecture [P291] were investigated in the literature.

B5. Software support processes (SUP)

Software verification and validation (71 studies). Automotive software-intensive systems are safety critical, required to meet real-time requirements, and distributed over multiple ECUs. Because of the increasing complexity and diversity of automotive software functions, software verification & validation has become increasingly important. The primary studies within this research topic discuss the application of formal analysis and verification methods to these issues, as well as hybrid and simulation-based approaches. A few experience reports and practical guidelines such as [P529, P530, P634, P104, P9, P516, P497, P9] are available in the literature and practitioners can benefit from these.

The majority of primary studies discuss on particular challenges and approaches relevant to the performance verification & validation of automotive functions [i.e. P207, P496, P457, P165, P149, P470, P372, P85, P46, P490, P458, P1]. Several studies focused on the static worst-case execution time (WCET) analysis method [i.e. P679, P84, P426, P149, P506, P630, P620], which is typically used to calculate the maximum execution time of a particular task on a specific hardware platform. Few studies investigated the challenges relevant to the scalability of existing methods, and examined the difficulty of integrating different methods and proposed integrated tool chains. For instance, previous studies [P457, P470] investigated seamless integration of existing timing analysis tools in order to provide a holistic approach for timing validation of automotive systems. The study [P390] argued that neither formal verification nor simulation-based verification approaches were suitable for large-scale complex systems and proposed a hybrid approach that scaled better and supported both bottom-up and top-down verification. There are also few studies [P274, P614, P650] that particularly discussed on performance verification and interface verification of distributed infotainment components. The remaining studies focused on various miscellaneous subjects, such as the verification of interaction behavior in distributed systems [P385], analysis and verification of communication behavior [P19], verification of state reachability [P361], and the formal analysis and verification of Matlab Simulink and Stateflow models [P104, P608, P600].

Software quality assurance and review (48 studies). Following MBSE, graphical models are now a central artifact in embedded software development. These models are used for different purposes and are realized in various modeling environments, such as Matlab Simulink and Stateflow. These models are used as a basis to derive source code using various code-generation tools, such as DSpace TargetLink and Matlab Real-Time Workshop. To support the quality assurance of automatically generated source code, numerous approaches including modeling guidelines, automatic code reviews, static analysis, and tool certification have been suggested in the literature [P345, P249, P99, P152, P186, P505, P296, P68, P569]. The term "safeguarding" implies techniques and mechanisms that increase confidence in automatically derived source code with respect to the expected quality attributes, such as correctness, safety, and reliability [P345]. A comprehensive overview of existing safeguarding techniques for automatically generating source code was presented in several studies [P345, P152, P505]. To improve model comprehension and the quality of derived source code, different modeling guidelines such as MISRA and the control style guideline developed by the MathWorks Automotive Advisory Board (MAAB) were recommended. Our review showed that a few primary studies reported their experiences in static analysis approaches to automatically checking the conformance of models with modeling guidelines [P325, P99, P356, P383, P512, P527].

Furthermore, challenges relevant to safety analysis and safety assurance are investigated in the literature [P389, P202, P4, P121, P120, P616, P610, P589]. To support the safety assessment and certification of automotive systems, several studies proposed the use of safety cases, which is a prevalent approach in the avionic industry. Safety cases are a technique used to systematically extract existing information about a system, its environment, and its context in order to demonstrate system safety. The safety cases can be represented in natural language or in a structured format. They typically constitute the documentation of the facts about the development process or the system safety process, such as compliance with safety standards and results of safety analyses. Moreover, several other miscellaneous subjects, such as quality assessment of requirement use cases [P172, P60], techniques for automated code review and model review [P227, P573], software health management for automatic detection and diagnosis of failures [P603], and automated approaches to identify fault-prone and difficult-to-maintain areas of source code [P257] were discussed among our primary studies.

Software documentation and configuration management (3 studies). The primary studies on this topic focused on configuration and change management in the context of a model-based environment [P160, P206], as well as the automated generation of documentation in model-driven development environments [P57].

B6. Software reuse processes (REU)

Software reuse (72 studies). Software reuse consists of all activities and mechanisms that support an organizations ability to reuse software items across project boundaries. Even though standardized architecture and middleware (i.e., AUTOSAR) promote reuse, the automotive industry still faces several challenges in this area [P408]. At present, while functionality does not differ more than 10% from one vehicle generation to the next, significantly more than 10% of the software is re-written [P408]. Reported issues include imprecise estimation of economic returns, compensation of immediate customer needs with long-term benefits, complexity in variability management, quality concerns, organizational issues, lack of comprehensive reuse strategy, and intensive source-code tailoring and optimization [P94, P173, P408]. Software reuse needs to be managed over the entire development process from requirement engineering onwards. However, the large numbers of variants and configurations prevalent in the automotive industry mean that variability can become extremely complicated, leading to significant costs and risks. Thus a comprehensive reuse methodology is needed, along with supporting mechanisms and tools.

In an effort to establish successful software product lines (SPL) in the automotive domain, previous studies [i.e. P215, P669, P503, P285, P86, P660, P221, P377, P331] have investigated existing challenges and proposed a variety of migration strategies. Previous works particularly investigated the evolution process of SPL and proposed methodologies for its successful establishment and management in the automotive industry. Findings from the literature indicate that variability management [i.e. P468, P493, P607, P492, P119, P5] and feature modeling [i.e. P13, P464, P331] were the main challenges and the most frequently discussed subjects in this area. There are also several studies that investigated the particular challenges relevant to the various phases of SPL, such as requirements [i.e. P79, P220, P13, P520], architecture [i.e. P86, P215, P47, P289, P11], realization [i.e. P669, P503, P299], or testing [i.e. P304, P343, P310, P602, P162]. A few other miscellaneous subjects, such as overviews of best practices and the challenges of software reuse in the automotive domain [P408, P414, P377, P173, P503, P94, P173], automated safety analysis and generation of modular product-line safety cases [P589, P588], automatic diagnosis and debugging of product configurations [P100, P101, P103], and metrics for size measurement and effort estimation of product families [P34, P71, P350], have also been investigated in the literature.

References

Aleti, A., 2015. Designing automotive embedded systems with adaptive genetic algorithms. Automated Softw. Eng. 22 (2), 199–240.
Aleti, A., Meedeniya, I., 2011. Component deployment optimisation with Bayesian learning. In: Proceedings of the 14th international ACM Sigsoft Symposium on Component Based Software Engineering. ACM, pp. 11–20.
Alur, R., Dang, T., Esposito, J., Hur, Y., Ivancic, F., Kumar, V., Mishra, P., Pappas, G., Sokolsky, O., 2003. Hierarchical modeling and analysis of embedded systems. Proc. IEEE 91 (1), 11–28.
Ampatzoglou, A., Stamelos, I., 2010. Software engineering research for computer games: a systematic review. Inf. Softw. Technol. 52 (9), 888–901.
Broy, M., 2005. Automotive software and systems engineering. In: Formal Methods and Models for Co-Design, 2005. MEMOCODE'05. Proceedings. Third ACM and IEEE International Conference on. IEEE, pp. 143–149.
Broy, M., 2006. Challenges in automotive software engineering. In: Proceedings of the 28th International Conference on Software Engineering. ACM, pp. 33–42.
Broy, M., Kruger, I.H., Pretschner, A., Salzmann, C., 2007. Engineering automotive software. Proc. IEEE 95 (2), 356–373.
Bryant, B.R., Gray, J., Mernik, M., 2010. Domain-specific software engineering. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research. ACM, pp. 65–68.
Caliebe, P., Herpel, T., German, R., 2012. Dependency-based test case selection and prioritization in embedded systems. In: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation. IEEE, pp. 731–735.
Choi, Y., Byun, T., 2015. Constraint-based test generation for automotive operating systems. Softw. Syst. Model. 1–18.
Clarke, S., Fitzgerald, B., Nixon, P., Pohl, K., Ryan, K., Sinclair, D., Thiel, S., 2008. The role of software engineering in future automotive systems development.
Dersten, S., Axelsson, J., Fröberg, J., 2011. Effect analysis of the introduction of AUTOSAR: a systematic literature review. In: Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on. IEEE, pp. 239–246.
Di Natale, M., Sangiovanni-Vincentelli, A.L., 2010. Moving from federated to integrated architectures in automotive: the role of standards, methods and tools. Proc. IEEE 98 (4), 603–620.
Fabbrini, F., Fusani, M., Lami, G., 2009. One decade of software process assessments in automotive: a retrospective analysis. In: Computing in the Global Information Technology, 2009. ICCGI'09. Fourth International Multi-Conference on. IEEE, pp. 92–97.
Fabbrini, F., Fusani, M., Lami, G., Sivera, E., 2008. Software engineering in the european automotive industry: Achievements and challenges. In: Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE, pp. 1039–1044.
Feitosa, D., Felizardo, K.R., de Oliveira, L.B.R., Wolf, D., Nakagawa, E.Y., 2010. Software engineering in the embedded software and mobile robot software development: a systematic mapping.. In: SEKE, pp. 738–741.
Fürst, S., 2010. Challenges in the design of automotive software. In: Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, pp. 256–258.
Fürst, S., Mössinger, J., Bunzel, S., Weber, T., Kirschke-Biller, F., Heitkämper, P., Kinkelin, G., Nishikawa, K., Lange, K., 2009. Autosar–a worldwide standard is on the road. 14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden, 62.
Giese, H., Neumann, S., Niggemann, O., Schätz, B., 2010. Model-based integration. In: Model-Based Engineering of Embedded Real-Time Systems. Springer, pp. 17–54.
Glass, R.L., Vessey, I., Ramesh, V., 2002. Research in software engineering: an analysis of the literature. Inf. Softw. Technol. 44 (8), 491–506.
Grimm, K., 2003. Software technology in an automotive company: major challenges. In: Proceedings of the 25th International Conference on Software Engineering. IEEE Computer Society, pp. 498–503.
Gruszczynski, B., 2006. An overview of the current state of software engineering in embedded automotive electronics. In: 2006 IEEE International Conference on Electro/Information Technology.
Harris, I., 2013. Chapter 22 - embedded software for automotive applications. In: Oshana, R., Kraeling, M. (Eds.), Software Engineering for Embedded Systems. Newnes, Oxford, pp. 767–816.
Heinecke, H., Damm, W., Josko, B., Metzner, A., Kopetz, H., Sangiovanni-Vincentelli, A., Di Natale, M., 2008. Software components for reliable automotive systems. In: Proceedings of the Conference on Design, Automation and Test in Europe. ACM, pp. 549–554.
Hryszko, J., Madeyski, L., 2015. Bottlenecks in software defect prediction implementation in industrial projects. Found. Comput. Dec. Sci. 40 (1), 17–33.
International organization for standardization, 2008. ISO/IEC 12207 Systems and Software Engineering - Software Life Cycle Processes. Technical Report.
Kasoju, A., Petersen, K., Mäntylä, M.V., 2013. Analyzing an automotive testing process with evidence-based software engineering. Inf. Softw. Technol. 55 (7), 1237–1259.
Keele, S., 2007. Guidelines for performing systematic literature reviews in software engineering. Technical report, Ver. 2.3 EBSE Technical Report. EBSE.
Kitchenham, B.A., Budgen, D., Brereton, O.P., 2011. Using mapping studies as the basis for further research–a participant-observer case study. Inf. Softw. Technol. 53 (6), 638–651.
Kitchenham, B.A., Dyba, T., Jorgensen, M., 2004. Evidence-based software engineering. In: Proceedings of the 26th International Conference on Software Engineering. IEEE Computer Society, pp. 273–281.
Lakhotia, K., Harman, M., Gross, H., 2013. Austin: an open source tool for search based software testing of c programs. Inf. Softw. Technol. 55 (1), 112–125.
Liggesmeyer, P., Trapp, M., 2009. Trends in embedded software engineering. Software, IEEE 26 (3), 19–25.
Mäntylä, M.V., Adams, B., Khomh, F., Engström, E., Petersen, K., 2015. On rapid releases and software testing: a case study and a semi-systematic literature review. Empirical Softw. Eng. 20 (5), 1384–1425.
Martínez-Fernández, S., Ayala, C.P., Franch, X., Nakagawa, E.Y., 2015. A survey on the benefits and drawbacks of AUTOSAR. In: Proceedings of the First International Workshop on Automotive Software Architecture. ACM, pp. 19–26.
Matinnejad, R., Nejati, S., Briand, L., Bruckmann, T., Poull, C., 2015. Search-based automated testing of continuous controllers: framework, tool support, and case studies. Inf. Softw. Technol. 57, 705–722.
Meedeniya, I., Aleti, A., Avazpour, I., Amin, A., 2012a. Robust archeopterix: Architecture optimization of embedded systems under uncertainty. In: Proceedings of the Second International Workshop on Software Engineering for Embedded Systems. IEEE Press, pp. 23–29.
Meedeniya, I., Aleti, A., Grunske, L., 2012b. Architecture-driven reliability optimization with uncertain model parameters. J. Syst. Softw. 85 (10), 2340–2355.
Meedeniya, I., Buhnova, B., Aleti, A., Grunske, L., 2011a. Reliability-driven deployment optimization for embedded systems. J. Syst. Softw. 84 (5), 835–846.
Meedeniya, I., Moser, I., Aleti, A., Grunske, L., 2011b. Architecture-based reliability evaluation under uncertainty. In: Proceedings of the Joint ACM SIGSOFT Conference–QoSA and ACM SIGSOFT Symposium–ISARCS on Quality of Software Architectures–QoSA and Architecting Critical Systems–ISARCS. ACM, pp. 85–94.
Müller, D., Herbst, J., Hammori, M., Reichert, M., 2006. It support for release management processes in the automotive industry. In: International Conference on Business Process Management. Springer, pp. 368–377.
Nejati, S., Adedjouma, M., Briand, L.C., Hellebaut, J., Begey, J., Clement, Y., 2013. Minimizing CPU time shortage risks in integrated embedded software. In: Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on. IEEE, pp. 529–539.
Nejati, S., Briand, L.C., 2014. Identifying optimal trade-offs between cpu time usage and temporal constraints using search. In: Proceedings of the 2014 International Symposium on Software Testing and Analysis. ACM, pp. 351–361.
Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P., 2014. Software development in startup companies: a systematic mapping study. Inf. Softw. Technol. 56 (10), 1200–1218.
Pernstål, J., Feldt, R., Gorschek, T., 2013. The lean gap: a review of lean approaches to large-scale software systems development. J. Syst. Softw. 86 (11), 2797–2821.
Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, 17. sn, pp. 1–10.
Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: an update. Inf. Softw. Technol. 64, 1–18.
Pfaller, C., Fleischmann, A., Hartmann, J., Rappl, M., Rittmann, S., Wild, D., 2006. On the integration of design and test: a model-based approach for embedded systems. In: Proceedings of the 2006 International Workshop on Automation of Software Test. ACM, pp. 15–21.
Pretschner, A., Broy, M., Kruger, I.H., Stauner, T., 2007. Software engineering for automotive systems: a roadmap. In: 2007 Future of Software Engineering. IEEE Computer Society, pp. 55–71.
Pretschner, A., Prenninger, W., Wagner, S., Kühnel, C., Baumgartner, M., Sostawa, B., Zölch, R., Stauner, T., 2005. One evaluation of model-based testing and its automation. In: Proceedings of the 27th International Conference on Software Engineering. ACM, pp. 392–401.
Rana, R., Staron, M., Berger, C., Hansson, J., Nilsson, M., Törner, F., 2013. Evaluating long-term predictive power of standard reliability growth models on automotive systems. In: 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE). IEEE, pp. 228–237.
Rana, R., Staron, M., Berger, C., Hansson, J., Nilsson, M., Törner, F., Meding, W., Höglund, C., 2014a. Selecting software reliability growth models and improving their predictive accuracy using historical projects data. J. Syst. Softw. 98, 59–78.
Rana, R., Staron, M., Hansson, J., Nilsson, M., 2014b. Defect prediction over software life cycle in automotive domain state of the art and road map for future. In: Software Engineering and Applications (ICSOFT-EA), 2014 9th International Conference on. IEEE, pp. 377–382.
Rodriguez, P., Haghighatkhah, A., Lwakatare, L.E., Teppola, S., Suomalainen, T., Eskeli, J., Karvonen, T., Kuvaja, P., Verner, J.M., Oivo, M., 2017. Continuous deployment of software intensive products and services: a systematic mapping study. J. Syst. Softw. 123, 263–291.

Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. 14 (2), 131–164.
Salzmann, C., Stauner, T., 2004. Automotive software engineering. In: Languages for System Specification. Springer, pp. 333–347.
Shaw, M., 2003. Writing good software engineering research papers: minitutorial. In: Proceedings of the 25th International Conference on Software Engineering. IEEE Computer Society, pp. 726–736.
Weber, M., Weisbrod, J., 2002. Requirements engineering in automotive development-experiences and challenges. In: Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on. IEEE, pp. 331–340.
White, J., Schmidt, D.C., Benavides, D., Trinidad, P., Ruiz-Cortés, A., 2008. Automated diagnosis of product-line configuration errors in feature models. In: Software Product Line Conference, 2008. SPLC'08. 12th International. IEEE, pp. 225–234.
Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Eng. 11 (1), 102–107.
Wohlin, C., Runeson, P., Host, M., Ohlsson, C., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell, MA, USA.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer Science & Business Media.
Wohlin, C., Runeson, P., Neto, P.A.d.M.S., Engström, E., do Carmo Machado, I., De Almeida, E.S., 2013. On the reliability of mapping studies in software engineering. J. Syst. Softw. 86 (10), 2594–2610.
Zhang, H., Babar, M.A., Tell, P., 2011. Identifying relevant studies in software engineering. Inf. Softw. Technol. 53 (6), 625–637.

Alireza Haghighatkhah is a Ph.D candidate and a researcher at M3S Research Unit in the Faculty of Information Technology and Electrical Engineering (ITEE) at the University of Oulu, Finland. He received B.Sc. in Applied Computing from Southern Cross University; Australia in 2010 and M.Sc. in Information Processing Science from University of Oulu, Finland in 2014. He has over four years of industrial experience in J2EE platform, design and development of large scale enterprise solutions. His research interests include empirical software engineering, software quality and process improvement, release engineering, evidence-based software engineering as well as automotive software engineering.

Ahmad Bani Jamali is a Ph.D. candidate in M3S research unit at the Faculty of Information Technology and Electrical Engineering (ITEE) at the University of Oulu, Finland. After his BSc. degree in Industrial Engineering, he has received two master's degrees in "IT Management" from Azad University and "Information Processing Science" from University of Oulu. Moreover, he has more than ten years of experience in a wide range of industries. His current research interests include empirical software engineering with focus on: automotive software ecosystems, automotive software deployment, data analytics, and cloud computing.

Olli-Pekka Pakanen is a Ph.D. student in software engineering at the University of Oulu. His current research interest are in the area of software quality and data mining.

Markku Oivo (PhD, eMBA) is Full professor and Head of M3S and M-Group at University of Oulu, Finland. During 2000–2002 he was Vice President and director at Solid Co. He held several positions at VTT 1986–2000. He had visiting positions at University of Maryland (1990-91), Schlumberger Ltd. (Paris 1994-95), Fraunhofer IESE (1999–2000) University of Bolzano (2014–2015) and Technical University of Madrid (2015). He worked at Kone Co. 1982-86. He has initiated and managed 100+ research projects with tens of millions of euros funding. He has over 100+ publications. His research interests include empirical research, software engineering methods and processes, agile, lean, cloud, startups, and management of technology.

Pasi Kuvaja (Ph.D.) is Vice Head of M3S Research Unit in the Faculty of Information Technology and Electrical Engineering (ITEE) at the University of Oulu, Finland. His research interests are in Software Engineering including: continuous deployment and delivery, DevOps, agile and lean software development, software quality, software process, software process assessment and improvement, embedded systems, and software measurement. He is one of the developers of the BOOTSTRAP methodology and product quality- driven software process improvement approach – PROFES. He has been actively contributing the SPICE project and was co-editor of ISO/IEC 15504 Part 7 – "Process Improvement Guide". He has worked three decades in close research co-operation with European industry. He has over 100 publications and has authored several books.