Information and Software Technology 55 (2013) 320–343

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

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

Variability in quality attributes of service-based software systems: A systematic literature review

Sara Mahdavi-Hezavehi, Matthias Galster *, Paris Avgeriou

University of Groningen, Department of Mathematics and Computing Science, P.O. Box 407, 9700 AK Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 16 February 2012
Received in revised form 9 July 2012
Accepted 26 August 2012
Available online 7 September 2012

Keywords:
Variability
Service-based systems
Quality attributes
Systematic literature review

ABSTRACT

Context: Variability is the ability of a software artifact (e.g., a system, component) to be adapted for a specific context, in a preplanned manner. Variability not only affects functionality, but also quality attributes (e.g., security, performance). Service-based software systems consider variability in functionality implicitly by dynamic service composition. However, variability in quality attributes of service-based systems seems insufficiently addressed in current design practices.
Objective: We aim at (a) assessing methods for handling variability in quality attributes of service-based systems, (b) collecting evidence about current research that suggests implications for practice, and (c) identifying open problems and areas for improvement.
Method: A systematic literature review with an automated search was conducted. The review included studies published between the year 2000 and 2011. We identified 46 relevant studies.
Results: Current methods focus on a few quality attributes, in particular performance and availability. Also, most methods use formal techniques. Furthermore, current studies do not provide enough evidence for practitioners to adopt proposed approaches. So far, variability in quality attributes has mainly been studied in laboratory settings rather than in industrial environments.
Conclusions: The product line domain as the domain that traditionally deals with variability has only little impact on handling variability in quality attributes. The lack of tool support, the lack of practical research and evidence for the applicability of approaches to handle variability are obstacles for practitioners to adopt methods. Therefore, we suggest studies in industry (e.g., surveys) to collect data on how practitioners handle variability of quality attributes in service-based systems. For example, results of our study help formulate hypotheses and questions for such surveys. Based on needs in practice, new approaches can be proposed.

1. Introduction

Variability is the ability of a software system to be adapted for different contexts [1]. Variability affects functionality as well as quality attributes of software systems. Even though variability is primarily studied in the software product line (SPL) domain [2–4], variability is a concern not only in the context of product lines but of many systems, including service-based systems [5]. Service-oriented architecture (SOA), the underlying architecture paradigm of service-based systems, has become a widely used concept in software engineering practice. SOA[1] supports adaptive systems in heterogeneous and changing environments [6]. In service-based systems, variability is usually achieved through flexible service retrieval and binding, mostly focused on functional aspects or business process variability. However, quality attributes (QAs), such as performance or safety, have not received much attention in the context of variability in service-based systems.

The objective of this paper is to describe the state-of-the-art of handling variability in quality attributes of service-based systems (detailed goals are outlined in Sections 1.3 and 2.1). Therefore, we present the results of a systematic literature review (SLR). Although reviews have been presented in similar fields, such as variability management in the product line domain [2], service-based systems [7], variability-intensive SOA systems [8], and service-oriented system engineering [9], the problem of handling variability in quality attributes has not been solved in generic problems. Also, even though heavily used in practice, there is no comprehensive study on variability in service-based systems that focuses on quality aspects; thus, we scoped our review only for the domain of service-based systems. In this article, variability in quality attributes of service-based systems refers to the ability that

* Corresponding author.
E-mail address: mgalster@ieee.org (M. Galster).

[1] We use the terms "service-oriented architecture", "service-oriented/-based software", "service-oriented/-based applications" and "service-oriented/-based systems" interchangeable (each service-oriented/-based software or system has an underlying service-oriented architecture).

0950-5849/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2012.08.010

a service can be delivered with several levels of QAs to fulfill the expectation of different service consumers. These levels of Quality of Service (QoS) requirements are negotiated with the service users and are defined in a Service Level Agreement (SLA). The architecture of a service-based system must be capable of dealing with different levels of QAs and at the same time ensure other QA requirements. For instance, a serviced-based system should be able to provide services with different levels of performance for distinctive consumers (e.g., an online store with a priority queue for premium customers) and at the same time keep the availability of the system at a desired level.

The review follows Kitchenham and Charter's guidelines for systematic literature reviews [10]. Furthermore, the review takes into account insights from practical experiences with systematic reviews [11–16].

1.1. Background

In the following section we briefly describe the definitions of service-based systems, quality attributes, and variability which we use in this paper.

1.1.1. Service-based systems and quality attributes

Service-orientation is a standard-based, technology-independent computing paradigm for distributed systems. As there is no universal definition for service, service-oriented architecture or service-oriented development [17], we utilize a broad definition: We consider service-oriented development as the development of a system which is assembled from individual services that are invoked using standardized communication models [6,18]. The two important principles of an SOA are (a) the identification of services aligned with business drivers, and (b) the separation of a service description (i.e., interface) from its implementation [19].

For quality attributes we adopt the definition from the IEEE Standard Glossary for Software Engineering Terminology [20]: A quality attribute is a characteristic that affects the quality of software systems. Here, quality describes to which degree a system meets specified requirements. Furthermore, we refer to quality attributes as discussed in the SWEBOK guide [21]. This guide integrates other quality frameworks, such as the IEEE Standard for a Software Quality Metrics Methodology [22], or ISO standards [23]. The SWEBOK distinguishes quality attributes discernible at runtime (e.g., performance, security, availability), quality attributes not discernible at runtime (e.g., modifiability, portability, reusability), and quality attributes related to the architecture's intrinsic qualities (e.g., conceptual integrity, correctness). In addition to the mentioned quality attributes which directly apply to a system, there are a number of business quality goals (e.g., time to market, cost and benefit, and targeted market) that shape a system's architecture [24].

As there are many quality attributes that are potentially relevant in the context of service-based systems, we scope our review. Gu and Lago found more than 50 quality-related challenges in service-based systems, including security, reusability, flexibility, interpretability, and performance, which are the most emphasized quality-related issues due to the dynamic nature of service-based systems [9]. Furthermore, O'Brien et al. [25] discussed quality attributes in service-based systems and identified the most significant attributes in the context of SOA. Finally, a quality model for service-based systems has also been proposed in the S-Cube project, in which several QAs (e.g., security, performance, cost, and usability) relevant to service-based applications are identified. [26]. Taking the quality attributes that are considered most important for service-based systems in each of these three sources [9,17,26] we aggregated the following list of quality attributes that we focused on when conducting our systematic review (definitions are taken from [17]):

1. *Reliability*: Reliability is the ability of the system to remain operating over time. Two important aspects of reliability in SOA are the reliability of message passing between services, and the reliability of services.
2. *Availability*: Availability is the degree to which a system or component is operational and accessible when it is needed.
3. *Security*: Security is associated with (a) access to information so that service is granted only to authorized subjects, (b) trust that the indicated author/sender of information is the one responsible for the information, and (c) assurance that information is not corrupted.
4. *Performance*: Performance may have different meanings in different contexts, but it is mainly related to response time and throughput.

We are mainly interested in the variability of aforementioned quality attributes with definitions presented above. However, the search strategy used in our review (Section 2.2) also identified studies addressing other quality attributes.

1.1.2. Variability

Variability is understood as the ability of a software artifact to be adapted (e.g., configured, extended) for a specific context, in a preplanned manner [1]. This means, we interpret variability as planned change, rather than change due to errors, maintenance or new unanticipated customer needs. Variability specifies parts of the system and its architecture which remain variable and are not fully defined during design time. Variability allows the development of different versions of an architecture/system. Variability in the architecture is usually introduced through variation points, i.e., locations where change may occur. Variability occurs in different phases of the software life cycle [27]. Design time variability defines variability of quality attributes at design time of the architecture. Runtime variability defines variability in quality attributes while the system is running, i.e., after design, implementation, etc. This is particularly relevant for service-based systems as these can be adapted and reconfigured at runtime [28].

Handling variability requires explicitly representing variability in software artifacts throughout the lifecycle of a software product. We use the term "handling" variability rather than "managing" variability. As argued by Svahnberg et al. [29], managing variability is only one of several activities in the context of handling variability. Managing variability comprises managing dependencies between variabilities, maintenance and continuous population of variant features with new variants, removing features, the distribution of new variants to the installed customer base, etc. Additional activities involved in handling variability include identifying variability (i.e., determining where variability is needed), reasoning, representing and implementing variability (i.e., use a variability realization technique resolve variability at variation points and to implement a certain variant) [29].

1.2. Lack of existing reviews

We could not identify any systematic reviews which study variability in service-oriented systems focusing on quality aspects. However, Chen et al. reviewed 33 approaches for variability management in the product line domain [2]. The study found that most current work addresses variability in terms of features, assets or decisions. Also, most work has been done on variability modeling; only little work has been presented to resolve variability at any time of the software life-cycle. There are three main differences between Chen et al. [2] and our review: First, we focus on quality as-

pects rather than on variability of features. Second, we study handling variability beyond variability management. Third, we focus on the domain of service-oriented systems instead of product lines. We argue that variability in service-oriented systems differs from variability in product lines:

a. Variability in service-based systems occurs at different levels of abstraction. For example, variability might be provided through parameter values used to invoke a service, or by replacing complete services. Product lines on the other hand usually address variability explicitly, in terms of features, assets or decisions, i.e., on a higher conceptual level.
b. Service-oriented systems face the challenge of meeting requirements for each organization while crossing boundaries between organizations [30]. Such systems run in the context of a volatile, distributed service composition environment in which services can change, fail, become temporarily unavailable, or disappear. This is usually not the case for software product lines which do not rely on the integration of services and third party applications.
c. Dynamic runtime variability and re-binding and re-composition at runtime must be supported. Product lines focus on compile time variability [28]. However, to fully support variability in service-oriented systems, events that occur in such systems must be coupled with rules to reason about execution alternatives [31].
d. Compared to software product lines, service-oriented computing includes a different design paradigm and its own principles, design patterns, a distinct architectural model, and related concepts, technologies, and frameworks [32]. These different principles, technologies, etc. cause the need for different methods to handle the variability issues (e.g., in terms of different model types).

In 2011 Montagud et al. presented a systematic literature review to classify quality attributes and measures for assessing the quality of software product lines [33]. The study found 165 measures related to 97 different quality attributes. Many measures (e.g., reusability, efficiency) were proposed for evaluating maintainability (92%). Additionally, 67% of the measures were used during the design phase of domain engineering, and 56% of the measures were applied to evaluate the product line architecture. Only 25% of previously proposed measures have been empirically validated.

A broad review on service-based systems was carried out by Brereton et al. [7]. This review aimed at (a) identifying main issues that need to be addressed to successfully implement service-based systems, (b) identifying solutions that have been proposed to address issues raised, (c) identifying research methods used to investigate proposed solutions, (d) providing frameworks for positioning new research activities, and (e) identifying gaps in current research. The review concluded that main issues that need to be addressed are managing evolution and change of systems, the selection of the most appropriate services, and service co-ordination. Solutions presented to address these issues focus on technologies. Research methods primarily used are those of concept implementation and conceptual analysis. Even though the goals and the topic area are quite similar to ours, we performed a more specific search by focusing on variability and QAs. Also, our method is different: We searched more than six journals (as done by Brereton et al.), and applied quality criteria to selected studies. We also performed a more formal data analysis. Most importantly, Brereton et al. study focused on the period from 2000 to 2004. However, many publication venues (in particular conferences and workshops targeted by SOA researchers) were established during the last 5 years.

Kontogogos and Avgeriou studied variability-intensive SOA systems [8]. Their review differentiated integrated variability modeling (extending traditional software artifacts with variability) and orthogonal variability modeling (adding new representations of variability separately from existing software). They found that most current approaches that could be applied to variability modeling in SOA are feature-based and stem from the product line domain. However, their study does not focus on quality aspects. Moreover, based on Kitchenham et al., their study cannot be considered as a systematic literature review but as an informal literature survey [13]. Similarly, Kazhamiakin et al. studied adaptation of service-based systems in an informal review [34].

In 2009, Gu and Lago presented a systematic literature review on service-oriented systems engineering [9]. The review explored challenges that have been claimed in studies published between January 2000 and July 2008. In this review, 51 primary studies were selected, from which more than 400 challenges were elicited. The study concluded that challenges can be classified along two dimensions: (a) based on themes (or topics) that they cover (e.g., service composition), and (b) based on characteristics (or types) that they reveal (e.g., technique challenges [9]). The paper pointed out quality as the top challenge.

Endo and Simao presented a systematic review on formal testing for SOA and web services [35]. They studied 37 papers focusing on testing aspects for single services and service compositions. The focus of this review was to identify formal approaches to test service-oriented architectures and web services. Similarly, Palacios et al. performed a mapping study to identify the state of the art of testing in SOA with dynamic binding [36]. The study found that the main objective of current research is to detect faults and to make decisions for dynamic binding based on the information gathered from tests. Furthermore, they discovered that monitoring and test case generation are the most frequently proposed methods to test functional and non-functional properties. Although these works are related to our domain of study in that they took quality attributes into consideration, their main concern is testing, and they do not consider the issue of variability in quality attributes.

1.3. Paper goal and contributions

A first step towards addressing variability in quality attributes of service-based systems is to identify current methods for handling variability in this context. Therefore, we define the goal of our study through Goal-Question-Metric (GQM) perspectives [37]:

Purpose: Analyze and characterize.
Issue: Handling variability in quality attributes.
Object: In service-based systems.
Viewpoint: From the viewpoint of researchers and practitioners.

We are particularly interested in:

a. Assessing the current research on handling variability in quality attributes of service-based systems.
b. Assessing provided evidence about current research regarding how far it can convince practitioners.
c. Identifying open problems and areas for improvement.

The target audience for this review is twofold: First, we aim at researchers who would like to get a systematic overview of the area of variability in quality attributes of service-based systems. Second, we aim at practitioners who would like to find out what methods to apply in what context.

1.4. Paper structure

This paper is organized as follows. Section 2 presents an overview of the systematic literature review method. We introduce

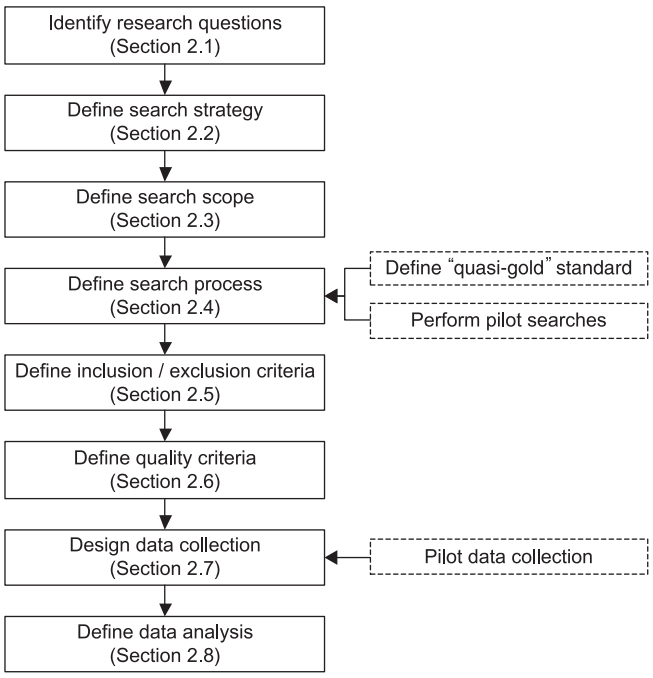

Fig. 1. Steps for developing review protocol (boxes include references to paper sections).

our research questions, discuss our search strategy and method, quality criteria, data extraction and analysis. Section 3 presents the results and how the collected data answers the research questions. Section 4 discusses the results, including main findings and limitations of the review. Finally, Section 5 concludes the paper.

2. Research method

The systematic literature review method is a well-defined method to identify, evaluate and interpret all relevant studies regarding a particular research question or topic area [10]. This method was chosen because we aimed at a credible and fair evaluation of studies on variability in quality attributes of service-based systems. A significant step when performing a systematic literature review is the development of a protocol (Fig. 1). The protocol specifies all steps performed during the review and increases its rigor and repeatability.

The protocol started with defining research questions, identifying the search strategy and search scope. Subsequently we designed a search process. As part of this step, we defined a "quasi-gold" standard for a search string [38]. Then, we developed a number of inclusion and exclusion criteria for studies that were identified in the search phase. Also, we proposed our strategy for assessing the quality of studies that we considered in the review. Next, we decided on the data elements to be extracted from the selected studies to help answer the research questions. As the final step, we designed our strategy to analyze the data extracted from studies. The protocol was reviewed by external reviewers. Moreover, the protocol was validated as follows:

a. We used a subset of resources to pilot the process. Problems we encountered when replicating the process were identified and the process revised accordingly.
b. The reliability of how to extract data from papers was piloted. A researcher was given a set of papers and asked to fill in the data extraction form. The objective of this step was to check whether data can be extracted based on the data extraction form, if the collected data was consistent between reviewers, and if the data allowed addressing the study goal.

2.1. Research questions

We aim at research questions important not only to researchers, but also to practitioners. Therefore, based on the study goal introduced in Section 1.3 our review covers the following research questions:

- *RQ1*: What quality attributes do existing methods for variability in quality attributes of service-based systems handle?
- *RQ2*: What software development activities are addressed by existing methods for handling variability in quality attributes of service-based systems?
- *RQ3*: What solution types are used by methods to handle variability in quality attributes of service-based systems?
- *RQ4*: What evidence is available to adopt proposed methods for handling variability in quality attributes of service-based systems?
- *RQ5*: Are methods only applicable to variability of design-time or run-time quality attributes?
- *RQ6*: Is there support for practitioners concerning how to use current methods?

We pose RQ1 to get an overview of what quality attributes existing methods deal with and if there are quality attributes that are studied more frequently than others. Moreover, in order to identify how variability in quality attributes of service-based systems fits into the software development process, we aim at finding out what software development activities are affected by handling variability in quality attributes (RQ2). In addition, we are interested in the solution types used by methods to handle variability in QA (RQ3). For example, methods could use common techniques from the product line domain, or other methods. Answering RQ1–RQ3 provides us with an overview of existing methods including their aforementioned characteristics. This is mainly of interest for researchers, even though RQ1 and RQ2 could also be relevant for practitioners. We pose RQ4 to help practitioners evaluate current methods for handling variability based on the provided evidence and to decide what methods they might use. Based on the definition used in [39], by "evidence" we mean any indicator that supports a proposed method and helps evaluate its validity and verifies whether a method works to address the problem targeted by this method. Furthermore, RQ4 helps researchers assess the quality of existing research. RQ5 and RQ6 help us outline directions for future research and identify areas that need work in order to make methods more applicable in practice.

With regard to the paper objectives outlined in Section 1.3, RQ1–RQ3 match with objective a. (assessing and reviewing current methods to handle variability issue), RQ4 matches with objective b. (analyzing quality of proposed method by assessing provided evidence and check if the methods can be adopted by practitioners), and RQ5 and RQ6 match with objective c. (RQ5 and RQ6 help discover limitations and liabilities of current methods which lead us to identify open problems and areas for improvement).

2.2. Search strategy

The search strategy is important so that relevant studies are included in the search results, without including too many irrelevant search results. The search strategy was based on

a. Preliminary searches in existing systematic reviews (e.g., variability management in the product line domain [2], service-based systems [7], variability-intensive SOA systems [8], and service-oriented system engineering [9]).
b. Reviews of research results (e.g., papers published at the Software Product Line Conference (SPLC), or the Workshop on Variability Modeling of Software-intensive Systems).
c. Trial searches and piloting using various combinations of search terms derived from the research questions.
d. Consultation with experts in the field through e-mail and personal conversations.

2.2.1. Search method

We used an automatic search by executing search strings on search engines of electronic data sources. In this review we aim for a broad search, rather than a focused search on key venues, therefore, we dealt with a huge number of studies. Manual search is not feasible for databases where the number of published papers can be over several thousand [40]. Moreover, manually searching journals and conferences might not cover all relevant venues (e.g., venues from other relevant domains, such as the business domain which publishes research on business processes, a discipline related to SOA and service-based systems). However, we also conducted a partial manual search to establish a "quasi-gold" standard.

2.2.2. Search terms for automatic search

We used an eight-step strategy to obtain our search strategy:

1. Derive major terms from the research questions and the topics being researched.
2. Identify alternative spellings, plurals, related terms and synonyms for major terms.
3. Check keywords in any relevant paper included in a "quasi-gold" standard.
4. When database allows, use Boolean "or" to incorporate alternative spellings and synonyms.
5. When database allows, use Boolean "and" to link the major terms from population, intervention and outcome.
6. Discuss between researchers.
7. Pilot different combinations of search terms in test executions and reviews.
8. Check pilot results with "quasi-gold" standard.

To create a good search string we established a "quasi-gold" standard, as proposed by Zhang and Babar [38]. For that reason, we manually searched a small number of venues. Results from these manual searches can be treated as a "quasi-gold" standard by cross-checking the result we obtain from the automatic search. Venues for the limited manual search were determined based on their significance for publishing research on service-based computing (see Table 24 in the Appendix A). We also limited the manual search to a time interval between January 2000 and February 2011 as the first papers on service-oriented computing started to appear around the year 2000. When manually searching the venues, we considered title, keywords, and abstract. The result of our manual search included 20 papers at first. However, when we started reading the papers, we excluded 17 papers based on inclusion and exclusion criteria (Section 2.5) and we ended up with three papers. Table 25 in the appendix presents the results to form the "quasi-gold" standard. The "quasi-gold" standard was expected to be a subset of the results obtained through automatic searches using a search string. This helped us get an idea if we were missing any papers in the automatic search.

Since we were particularly interested in performance, security, reliability and availability, we included these quality attributes in our search string. The search string consisted of three parts: Service-orientation AND variability AND quality attributes. The alternate keywords are connected through logical OR to form a reference search string for automatic search of databases:

```
(service OR services OR service-oriented OR
  service oriented OR service-based OR service
  based OR SOA OR software as service OR software as
  a service OR SaS OR SaaS)
AND
(change OR changes OR modification OR
  modifications OR modify OR adaptive OR adapt OR
  adaptation OR aware OR flexibility OR
  flexibilities OR product line OR product lines OR
  product family OR product families OR variability
  OR variabilities OR variant OR variants OR
  variation OR variations OR variation point OR
  variation points)
AND
(aspect OR aspects OR cross-cutting OR non-
  functional OR quality OR qualities OR quality
  attribute OR quality attributes OR quality factor
  OR quality factors OR system quality OR system
  qualities OR QoS OR quality of service OR service
  level OR service-level OR SLA OR performance OR
  security OR reliability OR availability)
```

Our reference search string went through modifications based on search features of electronic sources (e.g., different field codes, case sensitivity, syntax of search strings, and inclusion and exclusion criteria like language and domain of the study). Consequently, we used different search strings for different sources [10]. However, for each source a semantically and logically equivalent search string was created.

2.3. Search scope and sources to be searched

The scope of our search is defined in two dimensions: publication period (time) and source. In terms of publication period, we limited our search to January 2000 to February 2011. This is because the first papers on service-based systems appeared around ten years ago [9]. Furthermore, the first version of SOAP, a protocol for web services (the most popular technology for implementing service-based systems) was submitted to the World Wide Web Consortium (W3C) in 2000. Please note that even though major conferences on service-based computing started to emerge in 2004 (e.g., ICSOC), we chose to start the search in the year 2000 to avoid missing studies that were not published at a service-specific venue. Moreover, events on variability started to emerge in the year 2000 with the first product line conference.

In terms of source we identified six electronic data sources (Table 1). For each data source, we documented the number of papers that was returned per search (i.e., hits per search), and the number of selected results per search. In Table 1, the number of hits per search and the number of selected results per search differs for some of the electronic sources (e.g., for SpringerLink). The reason is that some electronic sources do not allow an automatic import of search results into the reference manager tool that we used (Mendeley). Thus, we manually imported groups of papers into the reference manager tool. To avoid manually importing huge number of irrelevant papers into the reference manager tool, we already filtered the results of automatic search at this stage, and excluded some of the clearly irrelevant papers (based on titles and abstracts) before importing them into reference manager tool.

Table 1
Electronic sources searched.

Electronic sources	Number of hits per search	Number of selected results per search
ScienceDirect	2237	2237
SpringerLink	952	50
Scopus	8904	8904
ACM Digital Library	3052	24
Web of Science	65	25
IEEE Xplore	2554	106
Total number of hits	15,210	N/A
Total number of imported papers	N/A	11,240

2.4. Search process

We used a staged study selection process (Fig. 2). At Stage 1 we searched databases listed in Section 2.3. The search string searched title, abstract and keywords. As explained above, before importing papers into the reference manager tool, we already excluded clearly irrelevant papers at this stage (based on title and abstract). At Stage 2, inclusion and exclusion criteria were applied (see Section 2.5). Initially, inclusion and exclusion criteria were interpreted liberally, i.e., if there was any doubt if a study should be included based on title, abstract and keywords it was included. As abstracts might be too poor to rely on when selecting primary studies [12] we also decided based on the conclusions of studies. Full copies of studies were obtained for the remaining studies. Final inclusion/exclusion decisions were made after full texts had been retrieved (Stage 3). For excluded studies, we documented reasons for exclusion. In case of multiple studies referring to the same method, only the most recent was included in the final review. Consequently, if a paper was the improved version of a previous paper, only the newer (improved) version was included in our study. Note that Fig. 2 also shows how the "quasi-gold" standard is related to the search results. In Fig. 2 duplicate papers refer to multiple occurrences of the same paper in different data sources which were imported into the reference manager tool.

2.5. Inclusion and exclusion criteria

To be included in the final review, a paper needed to meet all of the following inclusion criteria:

- *I1*: Study is internal to the service-oriented domain. We are interested in variability of quality attributes in service-based systems. This implies that studies are about service-based systems.

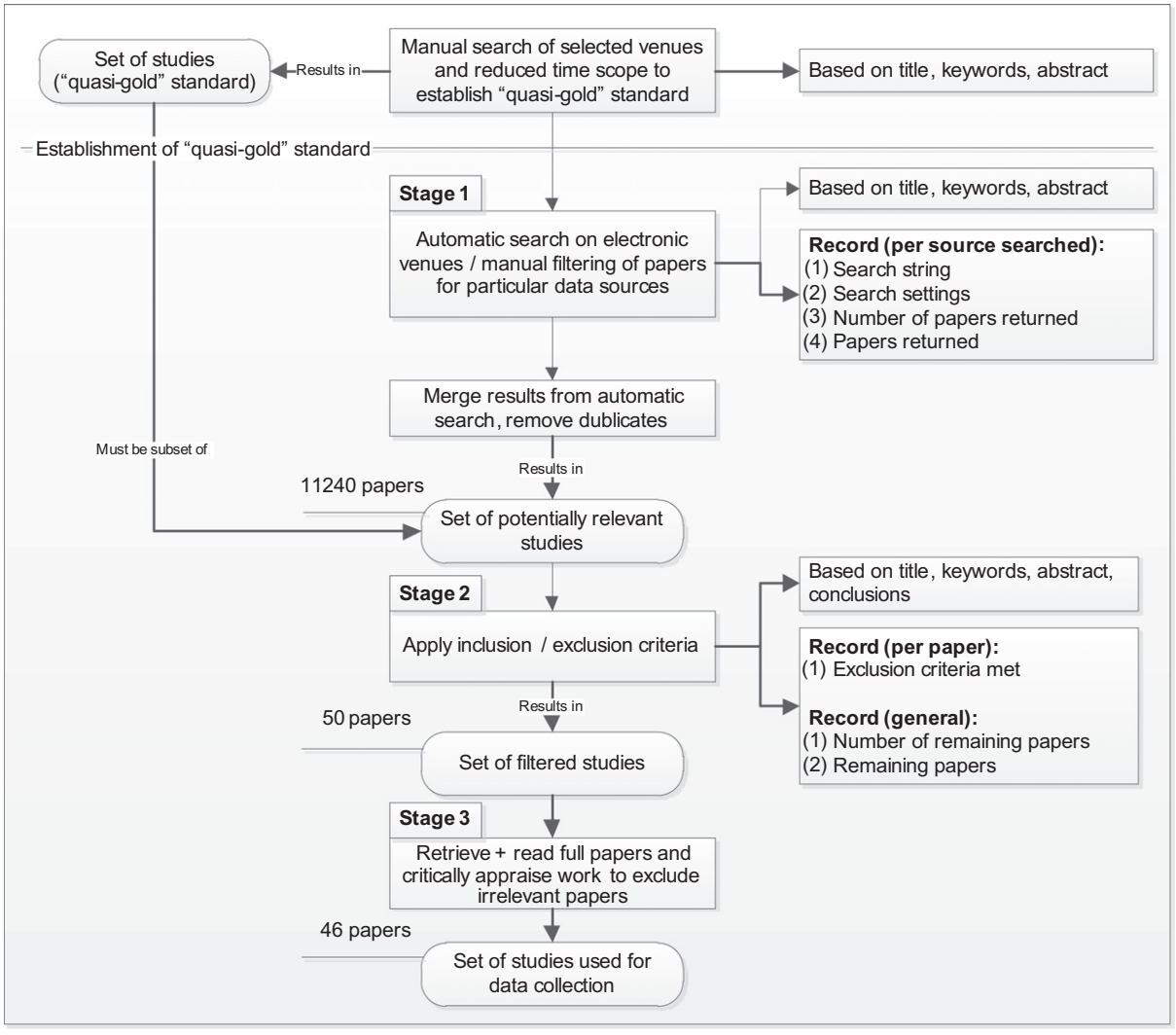

Fig. 2. Search process.

- *I2*: Study describes a method to handle variability in quality attributes.

Moreover, papers should not meet any exclusion criterion:

- *E1*: Study is marginally related to service-based systems. If the focus of a paper was about a field other than service-based systems and only marginally related to service-oriented systems, the paper was excluded. For example, a study that is mainly about how to design and develop health care information systems (which uses some external software services) should be excluded.
- *E2*: Study is in the domain of variability, but does not consider quality attributes. A paper that does not address quality attributes together with variability has no value to answer our research questions.
- *E3*: Study is editorial, position paper, abstract, keynote, opinion, tutorial summary, panel discussion, technical report, or a book chapter. As Kitchenham et al. [41] argue, grey literature are of lower quality than papers published in journals and conferences as they usually are not subject to a thorough peer-review. Books/book chapters were only included if they were conference/workshop proceedings (e.g., as part of the LNCS or LNBIP series) and are available through data sources are included in our study. Other workshops papers which are not available through the electronic data sources were not found in the automatic search.

Each study was reviewed by one researcher (based on title, keywords, and abstract) to determine a paper's relevance according to each criterion. When necessary, the content of the paper was also examined. For each reviewer result, another researcher independently performed sanity checks. Differences were reconciled collaboratively.

2.6. Quality criteria

All papers were evaluated against a set of quality criteria. Similar as Ali et al. [40], we adopted the quality assessment used by Dyba and Dingsoyr [42]. This instrument uses a three point scale to answer each question, either as "yes", "to some extent" or "no". By including "to some extend" we did not neglect statements where authors provided only limited information to answer the assessment questions. Each quality assessment question was answered by assigning a numerical value (1 = "yes", 0 = "no", and 0.5 = "to some extend"). Then, a quality assessment score was given to a study by summing up the scores for all the questions for a study (quality assessment score of a study). Quality criteria are:

- *Q1*: Is there a rationale for why the study was undertaken?
- *Q2*: Is there an adequate description of the context (industry, laboratory setting, products used, etc.) in which the research was carried out?
- *Q3*: Is there a justification and description for the research design?
- *Q4*: Does the study provide description and justification of the data analysis approaches?
- *Q5*: Is there a clear statement of findings and has sufficient data been presented to support them?
- *Q6*: Did the authors critically examine their own role, potential bias and influence during the formulation of research questions and evaluation?
- *Q7*: Do the authors discuss the credibility and limitations of their findings explicitly?

We used quality assessment criteria for synthesis purposes and not for filtering papers. The calculated quality scores are used as one of the factors to validate all reviewed papers. This assessment is used to answer RQ4 which might be useful for practitioners or researchers who are interested in the validity of studies. The results of the quality assessment are provided in Section 3.5.

2.7. Data collection

The 46 selected primary studies have been read in detail to extract the data needed in order to answer the research questions. Data was extracted using a data extraction form (Table 2).

Details about fields F10, F15, F18 and F19 are provided in the following. Adapting types of solutions from [43], we utilize the types of solutions as indicated in Table 3 for F10 ("Nature of solution").

Adopting architecture activities from [44], we used the development activities (F15) as indicated in Table 4. As quality attributes play a significant role during software architecting, we emphasized architecture activities.

The evidence level (F18) evaluates the evidence level of the proposed method. The results are critical for researchers to identify new topics for empirical studies, and for practitioners to assess the maturity of a particular method or tool. We adopted the classification proposed by Alves et al. [45] to make the assessment more practical. From weakest to strongest, our classification is as follows:

1. No evidence.
2. Evidence obtained from demonstration or working out toy examples.
3. Evidence obtained from expert opinions or observations.
4. Evidence obtained from academic studies (e.g., controlled lab experiments).
5. Evidence obtained from industrial studies (i.e., studies are done in industrial environments, e.g., causal case studies).
6. Evidence obtained from industrial application (i.e., actual use of a method in industry).

Category 5 includes studies done in industrial environments for the purpose of the research and not for using the method to achieve an operational goal. On the other hand, evidence level 6 means that the method has been used in practice, beyond evaluating it. According to Alves et al., industrial practice indicates that a

Table 2
Data extraction form.

#	Field	Concern/research question
F1	Author(s)	Documentation
F2	Year	Documentation
F3	Title	Documentation
F4	Source	Reliability of review
F5	Keywords	Documentation
F6	Abstract	Documentation
F7	Citation count (Google scholar as of mid 2011)	RQ4
F8	Quality score (according to schema introduced in Section 2.6)	RQ4
F9	Method proposed (brief description as free text)	RQ1–RQ3
F10	Nature of solution (see Table 3)	RQ3
F11	Domain (application domain of approach)	RQ1, RQ3, RQ5
F12	Runtime QAs	RQ1, RQ5
F13	Design time QAs	RQ1, RQ5
F14	Tool support	RQ6
F15	Development activities addressed (see Table 4)	RQ2
F16	Limitations (time, cost, learning curve, others)	RQ5, RQ6
F17	Research/practice/both	RQ6
F18	Evidence level	RQ4
F19	Evaluation approach (see Table 5)	RQ4

Table 3
Solution types as options for F10 (“Nature of solution”).

Abbreviation	Type of solution
MF	Feature model
UM	Using UML and its extensibility
AR	Express variability as part of a technique that models the architecture of the system
NL	Using natural language
SV	Expressed variability as part of a technique that models services of the system
FM	Formal techniques based on mathematics
DS	Domain-specific language
ON	Ontology based techniques
OR	Orthogonal variability management
Other	Other used solutions

Table 4
Development activities as options for F15 (“Development activities addressed”).

Abbreviation	Activity
AA	Architecture analysis
AS	Architecture synthesis
AE	Architecture evaluation
AM	Architecture maintenance
AI	Architecture implementation
ADs	Architecture design
AR	Architecture recovery
ADp	Architecture documentation and description
AIA	Architecture impact analysis
II	Implementation and Integration
R	Requirements
T	Testing
M	Maintenance

method has already been approved and adopted by industrial organizations [45]. Thus, practice shows a convincing proof that something works and is therefore ranked strongest in the hierarchy.

Based on [43], we used the categorization for evaluation approaches (F19) as shown in Table 5.

The difference between evidence level (F18) and evaluation approach (F19) is that F18 is more about the type of evidence which authors used to present their methods, whereas F19 is about the type of approaches they used to evaluate their proposed methods.

A record of extracted data was kept in Mendeley file and Excel spreadsheets for analysis. Data was collected by one researcher. Another researcher independently performed sanity checks. Differences were reconciled collaboratively.

2.8. Data analysis

Data from primary studies were summarized to answer the research questions. Most of the selected studies were grounded in qualitative research. As argued by Dyba and Dingsoyr, meta-analysis might not be suitable for synthesizing qualitative data [52]. Therefore, the data was manually reviewed. We performed descriptive synthesis to represent the results in tabular form. As found by other researchers, tabulating the data was useful during aggregation [53]. We used descriptive statistics for analyzing the data. As noted by Chen and Babar, frequency analysis has been used by other systematic reviews, which primarily deal with qualitative data [53].

3. Results and analysis

We used the extracted data to answer our research questions. In the following, first we give an overview of the identified studies and extracted information. Then, we answer the research questions by analyzing the data relevant to each question.

3.1. Results overview and demographics

After performing the filtering phases described in Section 2.5, we obtained 50 papers to be included in the data analysis. When studying these 50 papers, we found two more duplicated papers, and one paper being only an abstract (missed during the filtering based on inclusion and exclusion criteria). Therefore, we excluded these three papers from our study. Moreover, we could not access one particular paper [57]. We directly contacted the author to get the paper, but did not succeed. Finally, we ended up with 46 papers to review (see Table 26 in the Appendix A).

Fig. 3 shows the number of papers per year between January 2000 and February 2011. Furthermore, Fig. 3 shows how many papers were found in journals or conferences. According to Fig. 3, the first papers started to appear in 2004 and the highest number of studies has been published in 2009. Comparing to 2009, in 2010 the number of published papers decreased. As our search stopped in February 2011, there are no papers for 2011.

Fig. 3 also shows that only eight papers out of 46 were found in journals. These were mostly published in 2008 and 2009. Furthermore, Fig. 3 shows an interesting trend of published papers: Compared to studies on variability management in general and publications related to software product line engineering that started to emerge around the year 2000 [2], there seems to be a delay of 4–5 years before researchers started to investigate variability in quality attributes of service-based systems.

3.2. RQ1: What quality attributes do existing methods for variability in quality attributes of service-based systems handle?

To answer this question we analyzed the data of F11 (domain), F12 (runtime quality attributes), and F13 (design time quality attributes) from the data extraction form. Table 6 contains all

Table 5
Evaluation approaches as options for F19 (“Evaluation approach”).

Abbreviation	Evaluation approaches
RA	*Rigorous analysis*: Rigorous derivation of results and proof [46]
CS	*Case study*: Empirical inquiry that investigates a contemporary phenomenon within its real-life context, when boundaries between phenomenon and context are not clearly evident, and in which multiple sources of evidence are used [47]
DC	*Discussion*: Qualitative, textual, opinion-oriented evaluation. E.g., compare and contrast, oral discussion of advantages and disadvantages [48]
EA	*Example application*: Describing an application including an example to assist in the description, but the example is “used to validate” as far as the authors suggest [46]
EP	*Experience*: Result has been used on real examples, but not in the form of case studies or controlled experiments, the evidence of its use is collected informally or formally [46]
FE	*Field experiment*: Controlled experiment performed in industry settings [49]
LH	*Laboratory experiment with human subjects*: Identification of precise relationships between variables in a designed controlled environment using human subjects and quantitative techniques
LS	*Laboratory experiment with software subjects*: Laboratory experiment to compare the performance of newly proposed solution with existing solution [50]
SI	*Simulation*: Execution of a system with artificial data, using a model of the real word [51]

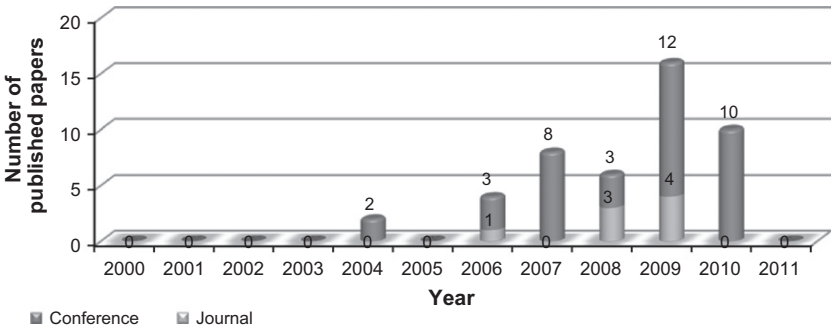

Fig. 3. Papers per year.

Table 6
Runtime quality attributes addressed by assessed studies.

QA	Number of papers	Study identifiers
Performance	34	S1, S3, S4, S5, S6, S7, S9, S10, S12, S13, S14, S16, S17, S18, S19, S20, S21, S22, S23, S24, S25, S26, S28, S29, S30, S31, S32, S33, S34, S35, S42, S43, S44, S45
Availability	18	S1, S3, S5, S6, S7, S10, S12, S16, S17, S19, S22, S25, S28, S29, S31, S38, S41, S44
Reliability	14	S1, S3, S6, S7, S15,S16, S17, S18, S28, S29, S30, S31, S33, S35
Security	5	S21, S23, S35, S36, S43

QAs defined in Section 1.1.1, number of studies addressing these QAs, and identifiers of studies. This table shows what studies addressed what runtime quality attributes.

Performance is the most addressed QA (34 papers). On the other hand, security is the least addressed QA (five papers). To further analyze how quality attributes are addressed, we investigated if studies consider only single quality attributes or specific sets of quality attributes. Eleven papers (24%) only address performance, two papers only address availability (4.4%), and one paper addresses reliability and security, respectively (2.1%). We list all possible sets of quality attributes, the number of papers that include these sets, and study identifiers in Table 7. Amongst the analyzed studies, 17.4% do not explicitly specify which runtime quality attributes are addressed. These studies (i.e., S2, S8, S11, S27, S37, S39, S40, and S46) mention "Quality of Service" (QoS) without specifying particular quality attributes.

A significant amount of proposed methods (19.5%) handle variability of performance and availability and reliability. From Table 7 it is obvious that certain QA sets are not taken into account by any study. These sets are specified by 0 under "Number of papers".

The only design time attribute which is addressed by some of studies is cost. Note that, although many studies consider cost as constraint, we consider cost as a business QA as defined in the S-Cube quality model [26]. In 41.3% of our assessed studies (i.e., 19 papers) cost is taken into account as a design time QA. All of the studies which address cost are listed in Table 8. The rest of the studies (58.7%) do not consider any design time QA.

In Table 9, we list the extracted study domains, the number of papers addressing these domains, study identifiers, and the QAs which are addressed by these studies. Most proposed methods are useful for the domains of web services and enterprise applications. Some of the domains in Table 9 may overlap but this categorization considers the closest domain of studies. We also added quality attributes from Table 6 to show what domains are concerned with what QAs. From this we can see that there are no quality attributes that only occur in one domain.

Table 9 does not cover all papers and their domains. Since certain studies belong to more than one domain, we listed the remaining of papers, their identifiers, and domains in Table 10.

3.2.1. Summary to answer RQ1

With regard to quality attributes handled by methods, most reviewed studies focus on the performance. Cost is the only design time QA which is addressed by several studies. Also, most of the studies were conducted in generic domains and do not define an application domain for which they can handle variability issues. We compared domains and QAs to see if there is a correlation between the domains of the studies and the QAs addressed by the methods. This was to find out if in certain domains only variability of particular QAs is addressed, or if variability of specific sets of QAs are more likely to be handled in particular domains. However, there seems to be no relation between the domain of the study and the QAs, and we did not find any quality attributes that would only occur in particular domains.

3.3. RQ2: What software development activities are addressed by existing methods for handling variability in quality attributes of service-based systems?

To answer this question we analyzed the data of F15 (development activities addressed) from the data extraction form. Table 11 presents software development activities, number of studies that address these activities, and study identifiers. Architecture design (ADs) has been addressed by 20 studies, while Implementation and Integration (II) has been addressed by 19 studies. On the other hand, several activities including architecture analysis (AA), architecture synthesis (AS), architecture evaluation (AE), architecture maintenance (AM), architecture recovery (AR), architecture documentation and description (ADp), and testing (T) have not been addressed by any study. Also, six studies (i.e., S1, S2, S9, S23, S24, and S33) do not explicitly address any software development activity.

Based on the fact that some studies take more than one activity into consideration, the total number of papers in Table 11 is 61 (several papers are listed more than once). Moreover, we analyzed how activities were addressed in isolation, and if certain software development activities were addressed together in sets. In Table 12 we only listed those sets of activities which were explicitly addressed by studies rather than listing all possible combinations of activities, number of papers addressing those sets of activities, and study identifiers.

As a result, six different sets of activities are provided in Table 12. As mentioned before, out of all 46 selected studies six of them did not explicitly consider any development activity. As indicated in Table 12, (II, ADs) is the most common set and has been considered by 10.9% of the papers, and (ADs, AA, AIA) and (AA, ADs) have been addressed least. Fifty-six percent of the studies

Table 7
Sets of runtime quality attributes and the number of papers addressing these sets.

QA combinations	Number of papers (percentage)	Study identifiers
Performance and availability	7 (15.2%)	S5, S10, S12, S19, S22, S25, S44
Performance and reliability	3 (6.5%)	S18, S30, S33
Performance and security	2 (4.4%)	S21, S23
Availability and reliability	0 (0.0%)	None
Availability and security	0 (0.0%)	None
Reliability and security	0 (0.0%)	None
Performance and availability and reliability	9 (19.5%)	S1, S3, S6, S7, S16, S17, S28, S29, S31
Performance and security and reliability	2 (4.4%)	S35, S43
Availability and reliability and security	0 (0.0%)	None
Performance and availability and reliability and security	0 (0.0%)	None

Table 8
List of design time QAs addressed by papers.

Design time QA	Study identifiers
Cost	S1, S3, S6, S7, S12, S13, S14, S22, S24, S25, S28, S29, S30, S31, S37, S38, S40, S42, S43

(i.e., 26 papers) address one activity, 30.4% of the studies (i.e., 14 papers) address multiple software development activities, and finally 13.1% of the studies (i.e., six papers) address no activity.

3.3.1. Summary to answer RQ2

The majority of the studies (i.e., 26 papers) only address a single software development activity. Architecture design, and implementation and integration are the most addressed activities. Fourteen papers address multiple software development activities and implementation and integration together with architecture design is the most frequently addressed combination of activities. For six studies (S1, S2, S9, S23, S24, and S33), we could not identify any development activity based on the information provided in these papers. Since most of the approaches deal with systems design and implementation, and integration of services in their proposed methods, it seems that current research regarding variability in QAs of service-based software systems is more focused on design and implementation phases of software systems and evaluation, testing or maintenance are not considered when dealing with variability in QAs.

3.4. RQ3: What solution types are used by methods to handle variability in quality attributes of service-based systems?

To answer this question we analyzed the data of F10 (nature of solution), and F11 (domain) from the data extraction form. In Table 13, we list all types of solutions and the papers that used those solution types. As we can see in Table 13, formal techniques (FM) is the most common solution type, as 13 of 46 papers use FM as their single solution type. On the other hand, service variability (SV), ontologies (ON) and domain-specific languages (DS) are used in one paper, and are the most uncommon solution types. In Table 13 we only listed papers (i.e., 27 papers) where one solution type is used by their proposed methods, number of papers addressing those solution types, and study identifiers.

We analyze data of F10 (nature of solution) from another point of view as well. Since some of the studies used more than one solution, we also list all those papers and their study identifiers (i.e., 15 papers) and assign them to solution type sets (Table 14). Among solution type sets, AR, FM is the most common set which is used by three papers, and (ON, FM), (FM, UM), and (SV, ON) are used only in one study.

Note that since the analysis presented in Tables 13 and 14 are from two different angles, the papers in these two tables do not overlap. Therefore, the sum of papers listed in these tables equals to 40 papers. The rest of the papers (i.e., four papers) presented new solution types not covered by our classification schema (S21, S23, S24, and S30). Table 15 presents studies that do not address any of our particular solution types, their study identifiers, and the extracted solution types from the papers.

3.4.1. Summary to answer RQ3

Most of the papers (i.e., 27 papers) only use one specific solution type to present their methods. Although it is not common, several studies (i.e., 15 papers) use two different solution types in their proposed methods. Formal techniques are the most common techniques used. Together with the fact that feature modeling is almost non-existent in our identified solution approaches, this shows that product line engineering does not have an impact on handling variability of quality attributes in SOA. This means that, although variability issue is widely studies in software product lines, researchers have not attempted to adapt approaches from product line engineering (e.g. modeling of features), to model variability of non-functional requirements.

3.5. RQ4: What evidence is available to adopt existing methods?

Several factors were used to evaluate the trustworthiness of a study: citation count (F7), quality score (F8), evidence level (F18), and evaluation approach (F19).

3.5.1. Citation count

The first factor is the citation count of the publication. By counting the number of times a study has been cited, we can estimate the impact of that study. For instance, if a study has a high number of citations, we can conclude that the study has been the subject of discussion in other published studies. Table 16 shows papers, citation counts, citation counts excluding self-citations, and aver-

Table 9
Studies belonging to single domain.

Domain	Number of papers	Study identifiers	QAs
Web services	14	S2, S3, S7, S10, S19, S21, S25, S28, S31, S34, S36, S43, S45, S46	Performance, availability, reliability, security
Enterprise and business applications, and e-commerce	13	S5, S11, S12, S15, S16, S20, S29, S35, S38, S40, S41, S42, S44	Performance, availability, reliability, security
Telecommunication	2	S8, S32	Performance, availability, reliability
Distributed computing	5	S4, S6, S22, S27, S33	Performance, availability, reliability
Cloud computing	1	S1	Performance, availability, reliability
Grid computing	1	S14	Performance
Network-accessible services	1	S18	Performance, reliability
Service-oriented computing	4	S17, S24, S30, S37	Performance, availability, reliability

Table 10
Studies belonging to multiple domains.

Domain	Study identifiers	QAs
Web services, telecommunication	S9, S23	Performance, security
Web services, enterprise and business applications	S13, S26	Performance
Telecommunication, enterprise and business applications	S39	None

age citation counts per year (i.e., citation count excluding self-citation/[2011-publication year]).

Table 17 shows citation counts excluding self-citations, number of papers related to citation counts, and study identifiers.

As we can see, the lowest citation count and the highest citation counts are 0 and 63, respectively. Forty papers (around 87%) have a citation count in range of 0–20, and only 6 papers (13%) have high citation counts in range of 21–63.

3.5.2. Quality score

The second factor which we used to validate the studies was their quality score. Based on the description we provided in Section 2.6, each study received a quality score between 0 and 7, having intervals of 0.5. The list of studies and their related scores for each of the quality assessment questions are shown in Table 27 in Appendix A. Fig. 4 shows quality scores and the number of papers with those quality scores. As we can see, there are certain quality scores which were not assigned to any of the papers: less than 1.5 and greater than 5.5. This means that, we did not have any papers fulfilling none or all of the quality criteria. The most common score is 3, which was achieved by 24% of the papers, and the rarest score is 5 (one paper). The highest score is 5.5 (four papers) and the lowest score is 1.5 (two papers).

To analyze the data based on the quality questions, we summarized our data in Table 18. The first column of this table shows quality assessment questions as provided in Section 2.6. The other columns show the number of papers assigned to each score per question.

In the following we provide an analysis for the answers to each quality question:

- *Q1: Is there a rationale for why the study was undertaken?* Forty-five of 46 of assessed papers scored 1 answering this question. This means that almost all of the assessed papers include a rationale. Although the quality and the level of details for their rationale might be different, 98% include a rationale.
- *Q2: Is there an adequate description of the context (e.g., industry, laboratory setting, products used, etc.) in which the research was carried out?* Nine percent of the assessed studies (i.e., four papers) do not provide any description about the context of the research, and 34.8% (i.e., 16 papers) of the studies address this issue to some extent. However, most studies (56.5%, 26 papers) provide an adequate description of the context, and whenever it is applicable, the research setting is explained.
- *Q3: Is there a justification and description for the research design?* Seventy-six percent of the studies (i.e., 35 papers) scored zero answering this question, and 15.3% of the studies (i.e., seven papers) scored 0.5, which means these studies explained the research design to some extent. Only 8.7% of the studies (i.e., four papers) provided a full justification and description of the research design.
- *Q5: Is there a clear statement of findings and has sufficient data been presented to support them?* Although 6.5% of the studies (i.e., 3 papers) do not clearly explain findings and include supporting data, 30.5% of the studies (i.e., 14 papers) explain findings to some extent and 63% of the studies (i.e., 29 papers) entirely state their findings and offer adequate data to support them.

Table 11
Development activities addressed in studies.

Activity	Number of papers	Study identifiers
Architecture analysis (AA)	3	S30, S37, S39
Architecture synthesis (AS)	0	None
Architecture evaluation (AE)	0	None
Architecture maintenance (AM)	0	None
Architecture implementation (AI)	4	S10, S18, S27, S36
Architecture design (ADs)	20	S4, S8, S10, S12, S16, S19, S20, S25, S27, S29, S32, S34, S35, S36, S37, S38, S39, S41, S44, S46
Architecture recovery (AR)	0	None
Architecture documentation and description (ADp)	0	None
Architecture impact analysis (AIA)	1	S37
Implementation and integration (II)	19	S3, S7, S11, S13, S14, S17, S19, S22, S25, S26, S28, S29, S31, S40, S42, S43, S44, S45, S46
Requirements (R)	3	S6, S7, S13
Testing (T)	0	None
Maintenance (M)	5	S5, S13, S15, S21, S42

Table 12
Sets of development activities addressed in assessed studies.

Activities	Number of papers (percentage)	Study identifiers
Implementation and integration (II), requirements (R)	2 (4.3%)	S7, S13
Architecture design (AD), architecture implementation (AI)	3 (6.5%)	S10, S27, S36
Architecture design (AD), architecture analysis (AA), architecture impact analysis (AIA)	1 (2.2%)	S37
Architecture analysis (AA), architecture design (AD)	1 (2.2%)	S39
Implementation and integration (AI), maintenance (M)	2 (4.3%)	S14, S42
Implementation and integration (II), Architecture design (AD)	5 (10.9%)	S19, S25, S29, S44, S46

Table 13
Nature of proposed solutions and papers.

Nature of solution	Number of papers	Study identifiers
Natural language (NL)	5	S1, S11, S16, S33, S45
Formal techniques based on mathematics (FM)	13	S3, S5, S9, S13, S14, S17, S19, S22, S26, S28, S31, S40, S42
Variability as part of a technique that models services of the system (SV)	1	S2
Variability as part of a technique that models the architecture of the system (AR)	6	S8, S12, S27, S32, S34
Ontology based techniques (ON)	1	S29
Domain-specific language (DS)	1	S6

- *Q6: Did the researcher critically examine their own role, potential bias and influence during the formulation of research questions and evaluation?* In most of the studies (i.e., 36 papers) researchers do not examine their own role and their possible influence during the formulation of research questions and evaluation. In 19.5% of the studies (i.e., 9 papers) researchers do pay attention to this issue, but their main concern is their role and potential influence on the evaluation and not the formulation of the research questions; only one of 46 papers fully addressed this issue.
- *Q7: Do the authors discuss the credibility and limitations of their findings explicitly?* Twenty-four percent of the studies (i.e., 11 papers) do not discuss the credibility and limitations of their findings at all; 52% of the studies (i.e., 24 papers) discuss credibility and limitations of the findings to some extent, and the remaining studies (24%) discuss this issue explicitly. However, the studies that explicitly discuss credibility and limitations usually focus the limitations rather than credibility of their findings.

The only remaining question which needs to be analyzed is Q4. Since the answer to this question also includes a "Not applicable" option, we analyze this question separately: *Q4: Does the study provide description and justification of the data analysis approaches?* This question referred to the existence of a discussion of the data analysis (as we would expect from an empirical paper), rather than to the existence of data as presented in a study. This means, even though we previously showed that some papers present experiments and case studies, data analysis might only be weakly discussed in these papers. Ninety-one percent of the approaches (i.e., 42 papers) do not include any data analysis; thus, we do not assign them any score but marked them as not applicable (N/A). Five percent of the studies (i.e., 2 papers) include data analysis approaches, but do not fully describe and justify the approaches, and 4.3% of the studies (i.e., 2 papers) completely describe and justify their offered data analysis approaches.

3.5.3. Evidence level

The third factor we used to check the credibility of the studies was evidence level which is described in Section 2.7. Table 19 relates evidence levels to papers; all the evidence levels, numbers of papers assigned to each evidence levels and their identifiers are listed.

Twenty-six papers (56.5%) obtained their evidence from demonstration or working toy examples (evidence level 2). We studied identified toy examples to see whether we can find any particular reoccurring example used in more than one study. However, we could not find such example. As can be seen in Table 19 few studies use expert opinions or observations, and industrial studies.

Table 14
Solution type sets and papers using them.

Nature of solution (sets)	Number of papers	Study identifiers
SV, FM	2	S7, S25
AR, FM	3	S15, S38, S46
SV, ON	1	S44
SV, NL	1	S4
AR, NL	3	S10, S20, S41
ON, FM	1	S43
FM, UM	1	S18
AR, UM	1	S37
NL, UM	1	S39
AR, DS	1	S35

3.5.4. Evaluation approach

The last factor we used to check the validity of the studies was whether the studies have provided an evaluation of their proposed variability approaches. Therefore, we mapped all evaluation methods (as presented in Table 5), to papers, in Table 20. Table 20 presents all evaluation approaches, the number of papers used each of these approaches, and their identifiers. Since some studies used more than one evaluation approach, we also listed all those sets of approaches used by these studies. Those studies (i.e., nine papers) which did not include any evaluation are listed in the last row labeled as "None". Field experiments and laboratory experiments with human subjects were not use by any study.

From Table 20 we conclude that 80.3% of the studies (i.e., 37 papers) use one or more evaluation approaches to evaluate the credibility of their proposed methods. Thirty-three of these 37 papers only use one evaluation approach and four papers use two approaches (S7, S5, S28 and S42). Among those papers that use one evaluation approach, simulation is the most used. Nine studies (19.7%) do not use any type of the evaluation approaches.

3.5.5. Summary to answer RQ4

Although most studies provide the reason for being conducted and also an adequate description of the context, not many studies critically examine the role of researchers and their potential influence on the study. In addition, only a few papers present a justification and description for their research designs, and perform a rigorous data analysis. However, most of the studies (i.e., 80.3%) tend to use one or several evaluation approaches to evaluate the credibility of their method. By comparing quality scores and citation counts assigned to each study, we found that the majority of studies with citation counts over 10 got a quality score over 3.5. Although this means that studies with higher citation counts are often more valuable, it does not work the other way around, as we can find studies with high quality scores and no citation counts (such as S4). We also conclude that since most studies use toy examples (weakest evidence level), the majority of the studies fail to provide trustworthy evidence to adopt their proposed variability methods. We also examined if studies with a high citation counts (higher that 20) and high quality scores (higher or equal to 3.5) used strong evidence levels. Therefore, we compared data from Tables 16 and 21 to data of Table 19 and found that only three studies (S5, S42, and S43) used convincing evidence (evidence from academic studies) for their methods. Finally, we compared data from Tables 19 and 20 to see whether there is a connection between the evaluation approaches used and evidence levels provided by the studies. This could help us to define if methods

Table 15
Studies not addressing one particular solution type.

Nature of solution (sets)	Number of papers	Study identifiers
A combination of UML modeling and graph transformation as a visual approach	1	S21
Controlled experiments to design model	1	S23
Simulation-based method	1	S24
Analytic Hierarchy Process based tool	1	S30

with higher evidence levels have been evaluated by using specific evaluation approaches. However, we could not associate any particular evaluation approach to the studies with higher evidence levels.

3.6. RQ5: Are methods only applicable to variability of design-time or run-time quality attributes?

To answer this research question we can use the analyses from previous sections (i.e., based on F11, F12, and F13) as follows: When we analyzed runtime quality attributes in Section 3.2, we already saw that studies address only performance, availability, reliability, and security. When considering the S-Cube quality model [26], we notice that although more than 60 QAs in the context of service-based systems exist, only few studies take some of these QAs into account. In Section 3.2, we also saw that only 19 studies address design time quality attributes, and the only design-time QA which is addressed is cost. This leads us to conclude that design time quality attributes are almost non-existent when it comes to handling variability.

3.6.1. Summary to answer RQ5

Our results indicates that although many different QAs exist in the domain of service-based systems, current methods can be used to handle variability of a limited number of run-time QAs (e.g., performance, availability, reliability, etc.). From another angle, we can conclude that the main concern of current studies are run-time QAs, and design-time QAs are not the main focus of methods for handling variability. Cost is the only design-time QA which is addressed by certain methods.

3.7. RQ6: Is there support for practitioners concerning how to use current methods?

To answer this question we analyzed the data of F14 (tool support), and F17 (research/practice/both) from the data extraction form. Table 21 indicates how many of our assessed studies include pure research work, practical work, or both, by relating the number of papers and their identifiers to "Research", "Practice", or "Both" categories. By practical work we mean implementation of the proposed method in an industrial setting, or an industrial context in which the study was conducted.

There is no study that presents pure practical work, and only four studies include both research and practice. Most of the studies only include research; this indicates that researchers have been focusing on the academic and theoretical aspects, and not much effort has been put on the implementation and use of proposed methods.

Table 22 lists all studies which provided tool support. Overall, 34.8% of the studies have tool support and the rest of the studies do not provide any tool support, neither for implementation of the method nor for evaluation.

Although all studies in Table 22 have tool support, there are some differences among them. First, unlike most of the studies (e.g., S21 and S24) that specify the supporting tool, some studies (such as S27 and S39) do not elaborate on the tool. These studies (S27 and S39) just mention that visualization, modeling and simulation software, and statistical analysis systems could be used while implementing the methods. Second, studies like S10 and S18 have developed the tools to support their proposed methods, but most of the other studies, such as S40 and S8 use tools and software which are already available. The last difference is related to the tools themselves. Each tool has been used for a particular purpose. To give an example, in S42 and S44, tools are used to implement certain parts of the system, in S41 the tool is used to measure the performance of the system, and in S7 the tool is used to administrate and monitor the implemented system. By comparing Tables 21 and 22 we can see that three out of four studies that include both research and practice (S6, S10, and S27), also provide tool support, and only one (S16) does not offer tool support.

3.7.1. Summary to answer RQ6

The fact that only two of our reviewed studies include practical work shows that current research fails to provide enough evidence for practitioners to adopt their methods. Practitioners value the studies which provide real life implementations of proposed methods. This would allow practitioners to recognize if the proposed methods are relevant and applicable to their environment. Also our results indicate that a limited number of studies provide tool support for their proposed methods. Thus, we conclude that there is a lack of support for practitioners. Also, this could mean that concerns of practitioners are not taken into consideration sufficiently.

4. Discussion of results

In the following we provide a summary of the main findings, limitations to the review, and threats to validity.

4.1. Main findings

4.1.1. Focus on certain quality attributes

Results indicate that the main concern of current approaches is to fulfill runtime QAs, and the main focus of the reviewed studies lies on certain types of quality attributes, especially on performance. Design-time QAs are almost neglected, and cost is the only design-time QA which is addressed by less than half of our selected studies (i.e., 19 papers). Furthermore, several studies use reliability and availability terms interchangeably and it is difficult to distinguish them from each other if no clear definition is presented. Although several types of relevant quality attributes are key drivers in the domain of service-oriented computing, most variability methods emphasize performance, availability, and reliability. Quality models that contain an extensive list of QAs, such as the S-Cube quality model, which targets quality attributes for service-based systems, are not covered by current approaches for variability in quality attributes of service-based systems.

4.1.2. Impact of product line engineering

Based on the fact that the product line domain is the domain that focuses on variability, and feature modeling is one of the well-known used methods in product line engineering [3], we were expecting to find some studies which use feature modeling in their proposed solutions. However, none of the studies uses feature modeling. Instead, most studies use formal techniques. This is an indicator that product line engineering and related paradigms have only little impact on variability in quality attributes of service-based systems. This is different to managing variability in product lines, where most approaches to manage variability are based on feature modeling and use UML or its extensions [43].

Table 16
Citation counts and average citation counts per paper.

Study identifiers	Citation counts	Citation counts excluding self-citations	Average citation count per year	Study identifiers	Citation counts	Citation counts excluding self-citations	Average citation count per year
S1	3	2	1	S24	4	3	1.5
S2	1	1	0.25	S25	2	1	0.5
S3	5	4	1	S26	2	1	0.5
S4	0	0	0	S27	0	0	0
S5	33	29	9.6	S28	0	0	0
S6	23	14	7	S29	12	6	3
S7	0	0	0	S30	3	2	2
S8	4	0	0	S31	3	3	3
S9	1	0	0	S32	8	5	2.5
S10	6	4	4	S33	6	3	3
S11	0	0	0	S34	7	3	1
S12	1	1	1	S35	55	48	6.8
S13	0	0	0	S36	2	2	0.5
S14	3	3	1.5	S37	14	8	2
S15	9	2	0.6	S38	5	4	0.8
S16	23	22	4.4	S39	21	20	2.8
S17	0	0	0	S40	0	0	0
S18	26	19	9.5	S41	1	1	1
S19	48	41	10.25	S42	61	58	19.3
S20	4	3	0.6	S43	71	63	21
S21	1	1	0.5	S44	9	4	1
S22	0	0	0	S45	2	0	0
S23	14	7	2.3	S46	5	4	0.8

Table 17
Papers with citation counts excluding self-citations.

Citation counts	Number of papers	Study identifiers
0	12	S4, S7, S8, S9, S11, S13, S17, S22, S27, S28, S40, S45
1	6	S2, S12, S21, S25, S26, S41
2	4	S1, S15, S30, S36
3	6	S14, S20, S24, S31, S33, S34
4	5	S3, S10, S38, S44, S46
5	1	S32
6	1	S29
7	1	S23
8	1	S37
1	1	S6
19	1	S18
20	1	S39
22	1	S16
29	1	S5
41	1	S19
48	1	S35
58	1	S42
63	1	S43

Table 18
Number of papers assigned to each score per question.

	0	0.5	1
Q1 (rationale for study)	0	1	45
Q2 (description of research context)	4	16	26
Q3 (justification of research design)	35	7	4
Q5 (clear statement of findings)	3	14	29
Q6 (critical examination of researchers' role)	36	9	1
Q7 (credibility and limitations)	11	24	11

On the other hand, Chen et al. found that formal techniques are only rarely used to manage variability in product lines. This might be because variability in product line engineering focuses on the functional requirements and variability in terms of features, assets and decisions, instead of non-functional requirements, as shown in a recent systematic literature review [2]. Therefore, popular methods used in the product line domain (such as feature modeling) might be of no help for variability in quality attributes of service-based systems.

4.1.3. Poor evidence of proposed methods

Similar to results of Chen and Babar [53], most studies that do provide evidence for their offered methods, get their evidence from demonstrating toy examples, which is the weakest evidence level in our hierarchy of evidence levels. Chen and Babar also noticed that only little experimental or elaborated comparative analysis is available to show the relative advantages and disadvantages of different variability management approaches in software product lines. Our study shows that this also applies with regard to variability in QA's of service-based systems. Although toy examples help illustrate the methods, the lack of industrial evidence is an indication that the method has not been adopted by any industrial organizations yet, therefore, it is hard to build evidence-based guidance for practitioners to select approaches for specific context.

4.1.4. Implications for practitioners and researchers

We list general limitations of proposed studies in Table 28 in Appendix A. However, to discuss the implications for researchers and practitioners, we evaluate the relevance of proposed methods to handle variability in QAs of service-based systems. For evaluating relevance, we adopted the model presented by Ivarsson and Gorschek [54]. According to this model, relevance refers to the potential impact that the research has on both academia and industry. To evaluate the relevance of existing work, Ivarsson and Gorschek address two different issues:

- First, the realism of the environment in which studies are conducted. To evaluate the realism, three aspects are considered: (1) subjects involved in a study and which should be representatives of the intended users of a proposed approach, (2) the scale at which a study is conducted, and (3) the context in which a study is performed. The first aspect is not applicable in our case as not all studies involve subjects. Furthermore, we merged the second and third aspect, i.e. scale and context to one factor, since they are somehow overlapping as we are using the same data to address them.

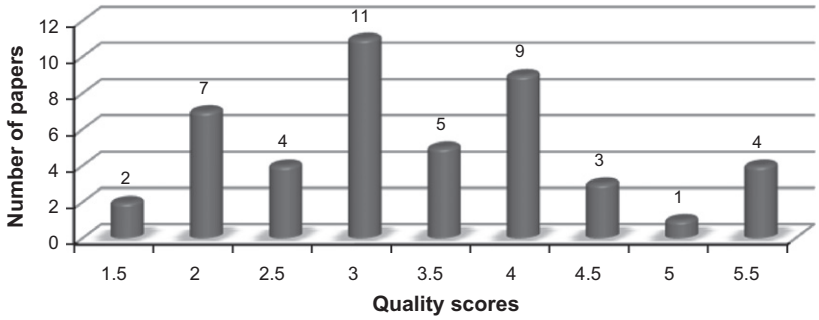

Fig. 4. Quality scores of papers.

- Second, the research method used to produce the results influence the relevance of the evaluation.

Therefore, we only use two aspects to evaluate the relevance of reviewed works: scale/context, and research methods. To use Ivarsson and Gorschek's model for evaluating relevance, we mapped evidence level and evaluation level from the data extraction form (see Table 2) to scale/context, and research methods aspects, respectively. We assigned 0 and 1 scores to the evidence levels (see Section 2.7) and evaluation approaches (Table 5) in Table 23.

The relevance score for a study was determined by summing up the scores for context/scale and research method. Thus, the maximum value for relevance was 2.

To further explore implications for practitioners and researchers, we studied rigor of current research, based on Ivarsson and Gorschek [54]. Rigor refers to both how an evaluation is performed and how it is reported. If a study is poorly described, neither reviewers nor researchers can evaluate the rigor of the evaluation. In their model, Ivarsson and Gorschek consider three aspects to score rigor: the extent to which context, study design and validity are described. We mapped our related quality criteria to their proposed factors as following: "Q2: Is there an adequate description of the context in which the research was carried out? " was mapped to the "Context" aspect, "Q3: Is there a justification and description for the research design?" was mapped to "Study design", and "Q6: Did the authors critically examine their own role, potential bias and influence during the formulation of research questions and evaluation? " was mapped to "Validity". We used a similar numerical scoring system as introduced in Section 2.6 to calculate the rigor value. Quality assessment questions were answered by assigning a numerical value (1 = "Strong", 0 = "Weak", and 0.5 = "Medium"). Then the rigor score was given to each study by summing up the scores for all the questions (i.e., Rigor = Q2 (context) + Q3 (Study design) + Q6 (Validity)). The maximum value for rigor was 3. The scores to each quality assessment questions are presented in Table 18. By summing up scores assigned to Q2, Q3, and Q6 provided in Table 18, and based on the scheme presented in Table 23, we obtained rigor and relevance values for each of the reviewed studies. Table 29 listed in Appendix A shows papers according to their rigor and relevance. These scores are represented in a bubble chart in Fig. 5. The size of bubbles shows the number of papers.

Fig. 5 shows that majority of the papers (i.e., 37 papers) are located in the lower part of the chart. This means that the majority of papers lack relevance. Also, three papers scored zero and 10 papers scored 0.5 for rigor, which means that these papers are poorly presented and are difficult to comprehend by researchers and practitioners and no strong evaluation is provided to indicate the relevance of the proposed methods. These papers do not present industrial evidence, real life experiments, and casual studies; therefore, it is difficult for practitioners to investigate the relevance and usefulness of proposed methods regarding to their own environments and industrial situations. Fourteen papers gained a score of 1 for rigor and 0 for relevance. This set of papers still lacks strong evaluation approaches, however, the context of these studies are described adequately and are easier to understand by researchers and practitioners. Generally, Fig. 5 indicates that there is substantial space for improvement of both rigor and relevance in this study domain.

Fig. 6 shows how rigor and relevance of studies have changed over period of 2000–2011. To get a better insight into rigor and relevance averages fluctuations over time, we performed regression analysis for each of the two data sets (i.e., rigor and relevance average values) in Fig. 6. A regression line or a trend line is a graphical representation of trends in data sets which assists to interpret the behavior of data over a specific period of time.

The rigor average graph which is located in the upper section of Fig. 6 shows that papers published in 2004 have the highest rigor average and after 2004 the rigor average starts to decrease significantly. Although in 2008 the rigor average of published papers increases comparing to 2006 and 2007, based on regression line of

Table 19
Papers assigned to evidence levels.

Evidence levels	Number of papers	Study identifiers
1 (No evidence)	10	S3, S13, S17, S21, S25, S28, S30, S33, S34, S41,
2 (Demonstrations, toy example)	26	S1, S2, S6, S8, S9, S10, S11, S12, S14, S16, S18, S19, S20, S22, S23, S26, S31, S32, S35, S36, S37, S39, S40, S44, S45, S46,
3 (Expert opinions, observations)	2	S5, S29
4 (Academic studies)	6	S4, S7, S15, S24, S42, S43
5 (Industrial studies)	1	S27
6 (Industrial evidence)	1	S38

Table 20
Papers assigned to evaluation approaches.

Evaluation approach	Number of papers	Study identifiers
Discussion (DC)	4	S1, S2, S6, S8
Rigorous analysis (RA)	2	S3, S19
Simulation (SI)	9	S4, S11, S14, S18, S23, S26, S31, S32, S35
Example application (EA)	6	S9, S13, S20, S34, S44, S45
Laboratory experiment with human subjects (LH)	8	S10, S15, S22, S29, S33, S41, S43, S46
Experience (EP)	2	S16, S24
Case study (CS)	2	S17, S21
Case study, simulation (CS, SI)	1	S7
Laboratory experiment with software subjects, discussion (LH, DC)	1	S5
Laboratory experiment with software subjects, rigorous analysis (LH, RA)	1	S28
Rigorous analysis, example application (RA, EA)	1	S42
None	9	S12, S25, S27, S30, S36, S37, S38, S39, S40

Table 21
Papers assigned to research/practice/both.

	Number of papers	Study identifiers
Research	42	S1, S2, S3, S4, S6, S7, S8, S9, S11, S12, S13, S14, S15, S17, S18, S19, S20, S21, S22, S23, S24, S25, S26, S28, S29, S30, S31, S32, S33, S34, S35, S36, S37, S38, S39, S40, S41, S42, S43, S44, S45, S46
Practice	0	–
Both	4	S5, S10, S16, S27

rigor average data set, we can see that the rigor average of published papers per years has decreased over time. For the relevance average graph, which is located on the lower section of Fig. 6, we can see that the relevance average of published papers in 2004 was zero and even though it remarkably improves in 2006, from the regression line of relevance average data set we can see that the relevance average of our 46 selected papers slightly decreases over the period of 2000–2011.

In the following we briefly discuss the common characteristics of studies with lowest rigor and relevance, and review methods proposed in studies with the highest rigor and relevance evaluations. Table 30 in the Appendix A lists papers with lowest rigor and relevance evaluations, QAs and development activities addressed by each of them, nature of solution of proposed methods, tool support, and evidence levels. Six papers (i.e., S11, S22, S20, S25, S28, and S37) deal with variability of QAs as a part of service discovery mechanisms and web service composition frameworks. Three papers (i.e., S1, S2, and S45) address QoS issue by proposing architectural solutions for system design. The most common solution types used by these papers are natural language (NL) and formal techniques based on mathematics (FM). Eight of these papers use toy examples, and one paper refers to academic studies as evidence for their proposed methods. Both of these evidence levels are considered to be weak according to our classification presented in section 2.7, and finally, four papers do not present any type of evidence to support their methods.

S24 with rigor and relevance (2.5,1) presents a simulation based approach to develop a general SOA simulation framework. The proposed methodology includes a tool to generate a simulator based on the Web Services Description Language. In S7, with rigor and relevance (1.5,1), Sui et al. introduce a dependable service-oriented middleware which supports QoS monitoring, configuration and runtime management functionalities to meet users' QoS requests. In their methodology the QoS of the composite service is promised by fulfilling the expected QoS for each of the atomic services with adaptive service scheduling mechanisms. In S16, with rigor and relevance (1.5,1), Garcia and Toldedo propose a web service architecture to support QoS management for web services. Their architecture includes brokers to assist service selection based on users' expected functional and non-functional requirements. In S19, with rigor and relevance (1,1), Cardellini et al. present a broker architecture which offers a composite service model with multiple QoS classes to different users to manage all the incoming flows of user requests. S38 addresses the issue of SOA adaptation, non-functional requirements management, and policy reconciliation between service providers and service requesters. In this work, Padmanabhuni et al. propose a constraint satisfaction based framework to represent, model and deal with policy based non-functional requirements in adaptive web services. In Table 31 (Appendix A), we list these studies, QAs and development activities addressed by each of them, nature of solution of proposed methods, tool support, and evidence levels.

From Table 31 (Appendix A) we can see that the most common development activity discussed in studies with high rigor and relevance is architecture design. Moreover, S16 and S19 introduce different broker architectures to handle QoS issue in their approaches. This indicates that in most of the papers with high rigor and relevance considering architectural aspects of the service based systems is a part of proposed methods.

Out of the five studies with highest relevance and rigor (i.e., S7, S16, S19, S24, and S38) only S38 obtained evidence from industry. The rest of the papers use weak evidence levels (e.g., toy examples) which does not help practitioners to evaluate the maturity and relevance of the proposed method. In addition, formal techniques based on mathematics (FM) are the most common solution types used by these studies (i.e., S7, S19, and S38). Although formal techniques help to build a mathematically rigorous model of complex systems and increase the reliability of system designs, they are more difficult to learn [55]. Therefore, great effort might be required to comprehend formal techniques, and transform the mathematically presented methods into methods which can be used by practitioners. Also, only S7 and S24 present tool supports for their proposed methods. All the aforementioned issues are obstacles for practitioners interested in adopting proposed methods.

4.1.5. Conceptual adaptation model

Adaptation can be described as a process to modify a service-based system in order to satisfy new requirements and to adjust to changes in the environment [54]. To further analyze our results, we adopted the generic conceptual adaptation model introduced

Table 22
Papers that provide tool support.

Study identifiers	Tool support
S5	ServiceGlobe (AutoGlobe component)
S6	Microsoft Oslo Toolkit
S7	Graphic Process Manager (Register Admin console, and Monitor console)
S8	UML, Eclipse Modeling Framework
S10	MOSES version 1 and 2 developed as part of this study
S18	MOSES prototype tool
S20	Application-specific middleware can be created using the ROAD (Role-Oriented Adaptive Design) framework
S21	Visual tools such as FUJABA or Murφ Model Checker can take the model as input and implement the method for tracing quality attributes
S24	Text analysis toolkit (TAPoRware)
S27	Visualization, modeling and simulation software
S30	AHP Wizard
S39	Statistical Analysis System
S40	IBM's Rational Software Architect modeling tool
S41	httperf (can be used for measuring the performance of Web servers for experiment)
S42	Discovery tool (for service search), Proxy Generator tool (for creation and deployment of the Proxy service)
S44	WSDL2JAVA tool (to automatically create a service stub for the discovered Web Service)

Table 23
Mapping of evaluation approaches and evidence levels to aspects for evaluating relevance.

Aspect	Scores	
	0	1
Context/scale	Evidence level 1, 2, 3 and 4	Evidence level 5 and 6
Evaluation	DC, EA, LH, LS, SI	RA, CS, EP, FE

by Kazhamiakin et al. and mapped certain data fields (i.e., F12, F13, F14, and F15) of Table 2 to their model [54] of Fig. 7.

In this model, adaptation actors are mapped to F15 (i.e., development activities) as several system users may deal with these activities during the adaptation process. By investigating the mapped development activities, we find the most and least addressed actors. Based on Table 11 the activities of adaptation initiator and adaptation designer actors are addressed by 24 and 23 papers respectively, which makes them the most addressed actors. Activities of adaptation requestor and adaptation executor actors

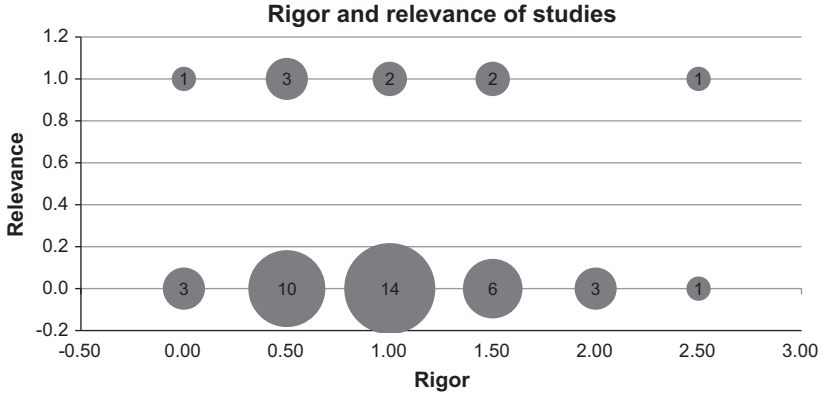

Fig. 5. Relevance and rigor of our 46 selected studies.

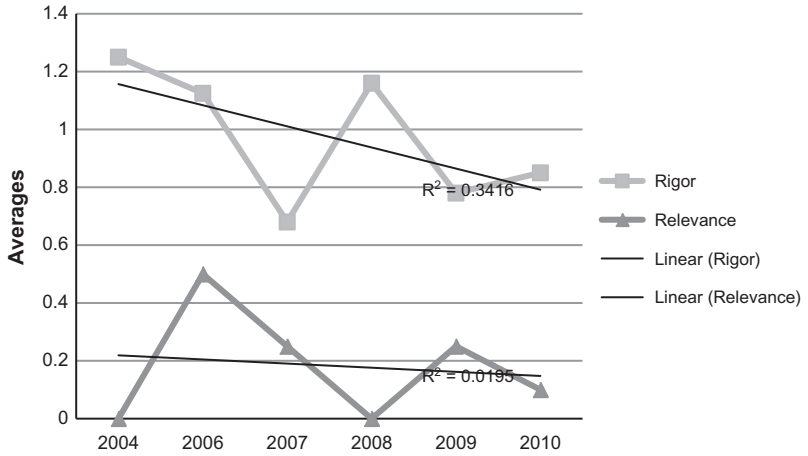

Fig. 6. Variations of rigor and relevance averages over time.

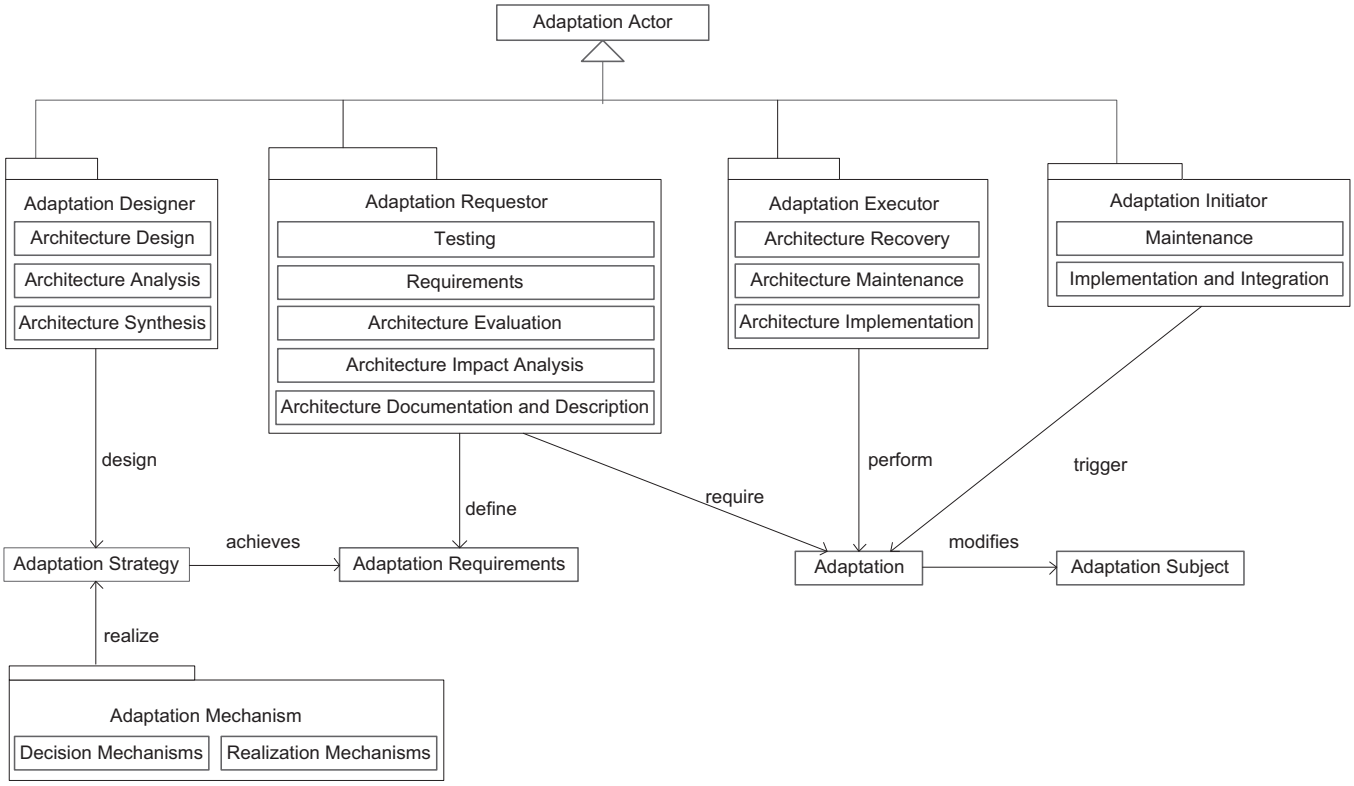

Fig. 7. Mapping of data fields to generic conceptual adaptation model.

are addressed by four papers which makes them the least addressed actors by the reviewed papers.

Adaptation strategies are the methods through which the adaptation requirements are fulfilled. Adaptation requirements are the service-based system model characteristics which are subject to variation (e.g., functionality of the system). Adaptation mechanisms may include tools, which could be mapped to F10 (i.e., nature of solution), and F14 (i.e., tool support) of our data extraction form in Table 2, and is used for performing adaptation actions (i.e., realization mechanisms) and tools for decision making (e.g., selecting an appropriate adaptation strategy among various alternatives) regarding adaption process (i.e., decision mechanisms). Adaptation subjects are different features and elements of service-based system, which could be mapped to F12 (i.e., runtime QAs), and F13 (i.e., design time QAs) of our data extraction form in Table 2, and might be modified during the adaptation process [54].

4.1.6. Research direction for future work

Since only a few of the quality attributes introduced in S-cube reference model are addressed in current studies, our first suggestion for researchers is to develop a better and more applicable quality attribute reference model for the service-based domain. Second, we suggest that researchers focus on enhancing the robustness of their methods instead of inventing new methods to handle variability. For instance, they may try to implement their methods in industrial environments to evaluate their method practically. This also provides guidelines for practitioners, and motivates them to start using the methods. Furthermore, we see a need for more empirical studies.

4.2. Limitations of the review and threats to validity

4.2.1. Inaccuracy and bias in selected papers for review

During the automatic search, our main goal was to ensure the completeness of selected papers. As mentioned before, we manually searched a limited number of venues and determined a "quasi-gold" standard as proposed in [38]. This helped us to make sure that the search string for the automatic search resulted in all the relevant papers. In the next phase, when we tried to exclude irrelevant papers, we wanted to reduce researcher's bias affecting the process of paper selection. Therefore, to mitigate this problem, a second researcher was checking excluded and included papers at each iteration of paper filtering.

4.2.2. Inaccuracy and bias in data extraction

As with any systematic review, one of the main limitations of our review is inaccuracy in data extraction. We had some difficulties to extract relevant information from our selected papers. For instance, several papers do not explicitly mention in which domain, the proposed methods can be used; several papers do not explicitly refer to any specific type of development activity in their methods, and some of the studies do not provide clear definitions for each of the quality attributes they considered in their methods. In situations like these, interpretation of information was needed. Therefore, the researcher's bias could affect the final extracted data. To mitigate this problem, in the case of domains, we tried to assign the papers to generic domains, while in the case of development activities we did not assign any activity to any method unless we were sure the paper was addressing the activity. One problem we encountered while analyzing quality attributes was the absence of definitions or poor definitions for quality attributes. We tried to check the meaning of the quality attributes in their context for each of the studies, but in some cases, studies do not provide a clear, or any definition for their discussed quality attributes. For instance, certain studies do not clearly define the meaning of availability and reliability, and because many researchers use these two terms interchangeably, or count them as one concept, we could not realize to which one of them they were actually referring. The same issues occurred when answering quality criteria questions, and assigning quality scores to the papers. Since it was a very subjective matter to

decide which quality score best suits each of the studies, the final score assigned to each paper can be inaccurate. Generally, to mitigate the inaccuracy of data extraction and quality scores, the researchers conducted discussions.

4.2.3. Deviations from the procedures for systematic reviews

Although we were determined to use the guidelines provided in [10] to perform our systematic review, we had deviations from their procedures. For instance, in our research a single researcher extracted the data rather than a group of researchers. Although this practice has been suggested in [14], this means that some of the data that we collected may be erroneous. Furthermore, we did not completely follow the guidelines of Kitchenham to use population, intervention, and outcomes to construct our search string. This is because using population, intervention, etc. only apply to empirical studies. We also found several published systematic reviews, such as [53,45,56], in requirement engineering and product line domains which do not use this method to create their search string.

4.2.4. Evaluation of review

Kitchenham et al. proposed four quality questions for systematic reviews[13]:

1. Are inclusion and exclusion criteria described and appropriate? Our review meets this criterion as we explicitly defined and explained inclusion and exclusion criteria.
2. Is the literature search likely to have covered all relevant studies? This criterion is met if either four or more digital libraries and additional search strategies are identified, or if all journals addressing the topic of interest are identified. We included more than four digital libraries in our search, so the criterion is met.
3. Did the reviewers assess the quality/validity of the included studies? We consider this criterion as met as we have explicitly defined quality criteria. We extracted quality criteria from each primary study.
4. Were the basic data/studies adequately described? We consider that this criterion is met as we used a detailed data collection form for each study. This data collection form was reviewed and piloted.

5. Conclusions

The goal of this paper is to systematically study variability of quality attributes in service-based systems. Our aim was assessing the quality of current research on variability in quality attributes of service-based systems, collecting evidence about current research that suggests implications for practice, and identifying open problems and areas for potential improvement. Our results suggest that design-time quality attributes are almost non-existent in current approaches available for practitioners, and product line engineering as the traditional discipline for variability management has almost no influence how we deal with variability in quality attributes of service-based systems. Also, variability at runtime in service-based systems is one if the main focuses of researchers in recent years.

Results of section four show that majority of papers do not present industrial evidence, real life experiments, and casual studies to support their proposed methods, thus, they fail to indicate the relevance of their proposed methods to the industrial environment. Furthermore, most of the researchers use formal techniques to present their methods. However, formal methods are difficult to learn and time consuming to apply in the real environment. Therefore, we suggest that researchers use more tangible and relatable evidence (e.g., experiments, casual case studies) to present their variability methods in the future. We also suggest using more comprehensible solution types, such as feature modeling, to help to reduce the effort needed to understand and apply the methods in the industrial levels.

Our results also show that the rigor of the papers has diminished remarkably over the past ten years. We suggest that researchers should be more meticulous about the reporting of their methods. This can be done by presenting adequate and perspicuous description of their method and the context, providing justification for the research design, and discussing the limitations, advantages, and disadvantages of their approaches. This will help both researchers and practitioners understand and evaluate the maturity of the methods and to decide if the method could be applied in specific environments.

Although Brereton et al. [7] stated that the selection of appropriate services need to be addressed, our results indicate that this issue has been fulfilled by most of the recent researches: Implementation and integration of services (which includes selection of services) are the most addressed development activities in proposed variability methods.

Our suggestion for performing systematic reviews in the future is to focus on variability of one specific quality attributes in service-based systems. Our systematic review showed that performance is the most addressed, and probably most important, quality attribute in this domain. Therefore, performing a review on handling variability of performance in service-based systems can provide useful information on how this QA is treated. Another option to investigate variability in quality attributes of service-based systems can be performing industrial surveys and collecting data on how practitioners actually handle the issue of variability of quality attributes in service-based systems.

Acknowledgements

We thank the reviewers for their valuable comments that helped improve the paper. This research has been partially sponsored by NWO SaS-LeG, Contract No. 638.000.000.07N07.

Appendix A

Tables 24–31.

Table 24
List of venues searched manually to establish "quasi-gold" standard.

Venues
IEEE Transactions on Services Computing
Journal of Service Oriented Computing and Applications
International Conference on Service Oriented Computing
International Conference on Services Computing
International Conference on Web Services
ServiceWave (2008, 2009, 2010)

Table 25
Results of manual search used to form "quasi-gold" standard.

Authors	Title	Venue
Narendra, N., Ponnalagu, K.	Towards a Variability Model for SOA-Based Solutions	IEEE International Conference on Services Computing (2010)
Narendra, N., Ponnalagu, K., Gomadam, K., Sheth, A.	Variation Oriented Service Composition and Adaptation (VOSCA): A Work in Progress	IEEE International Conference on Services Computing (2007)
Zhang, L., Arsanjani, A., Lu, D., Chee, Y.	Variation-Oriented Analysis for SOA Solution Design	IEEE International Conference on Services Computing (2007)

Table 26
List of reviewed studies in systematic literature review.

Study identifier	Authors(s)	Year	Title	Source
S1	Nallur, V., Bahsoon, R., Yao, X.	2009	Self-optimizing architecture for ensuring quality attributes in the cloud	Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture
S2	Narendra, N.C., Ponnalagu, K., Gomadam, K., Sheth, A.	2007	Variation Oriented Service Composition and Adaptation (VOSCA): A Work in Progress	IEEE International Conference on Services Computing
S3	Kim, Y., Doh, K.	2007	A trust type based model for managing QoS in Web services composition	International Conference on Convergence Information Technology
S4	Jiang, C., Hu, H., Cai, K., Huang, D., Yau, S.	2009	An intelligent control architecture for adaptive service-based software systems	International Journal of Software Engineering and Knowledge Engineering
S5	Gmach, D., Krompass, S., Scholz, A., Wimmer, M., Kemper, A.	2008	Adaptive Quality of Service Management for Enterprise Services	ACM Transactions on the Web
S6	Rosenberg, F., Leitner, P., Michlmayr, A., Celikovic, P., Dustdar, S.	2009	Towards Composition as a Service – A Quality of Service Driven Approach	IEEE 25th International Conference on Data Engineering
S7	Sui, Y., Zhou, X., Yang, G.	2009	QoS Decomposition for Dependable Service-Oriented Middleware	ISECS International Colloquium on Computing, Communication, Control, and Management
S8	Briones, J., De Miguel, M., Alonso, A., Silva, J.	2009	Quality of Service Composition and Adaptability of Software Architectures	IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing
S9	Wan, C., Wang, H.	2007	Uncertainty-aware QoS Description and Selection Model for Web Services	IEEE International Conference on Services Computing
S10	Caredellini, V., Iannucci, S.	2010	Designing a broker for QoS-driven runtime adaptation of SOA applications	IEEE 8th International Conference on Web Services
S11	Rajaram, K., Babu, C.	2010	Template based SOA framework for dynamic and adaptive composition of Web Services	International Conference on Networking and Information Technology
S12	Rajendran, T., Balasubramanie, P.	2010	An OptimalAgent-Based Architecture for Dynamic Web Service Discovery with QoS	2nd International Conference on Computing, Communication and Networking Technologies
S13	Liu, B., Shi, Y., Wang, H.	2009	QoS Oriented Web Service Composition and Optimization in SOA	Joint Conferences on Pervasive Computing
S14	Jun-Zhou Luo, J., Zhou, J., Wu, Z.	2009	An adaptive algorithm for QoS-aware service composition in grid environments	Service Oriented Computing and Applications
S15	Zheng, Z., Lyu, M.	2008	A QoS-Aware Middleware for Fault TolerantWeb Services	19th International Symposium on Software Reliability Engineering
S16	Garcia, D., de Toledo, M.	2006	A Web Service Architecture Providing QoS Management	4th Latin American Web Congress
S17	Peng, D., Chen, Q.	2009	QoS-aware Selection of Web Services Based on Fuzzy Partial Ordering	International Conference on E-Business and Information System Security
S18	Cardellini, V., Casalicchio, E., Grassi, V., Lo P., Mirandola, R.	2009	QoS-driven Runtime Adaptation of Service Oriented Architectures	Joint 12th European Software Engineering Conference and 17th ACM SIGSOFT Symposium on the Foundations
S19	Cardellini, V., Casalicchio, E., Grassi, V., Lo, F.	2007	Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes	IEEE International Conference on Web Services
S20	Colman, A., Pham, L., Han, J., Schneider, J.	2006	Adaptive Application-Specific Middleware	1st workshop on Middleware for Service Oriented Computing
S21	Golshan, F., Barforoush, A.	2009	A New Approach for Tracing Quality Attributes in Service Oriented Architecture Using Graph Transformation Systems	14th International CSI Computer Conference
S22	Li, M., Deng, T., Sun, H., Guo, H., Liu, X.	2010	GOS: A Global Optimal Selection Approach for QoS-Aware Web Services Composition	5th IEEE International Symposium on Service Oriented System Engineering
S23	Yau, S., Ye, N., Sarjoughian, H., Huang, D.	2008	Developing Service-based Software Systems with QoS Monitoring and Adaptation	IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems
S24	Smit, M., Nisbet, A., Stroulia, E., Iszlai, G., Edgar, A.	2009	Toward a Simulation-generated Knowledge Base of Service Performance	4th Workshop on Middleware for Service Oriented Computing
S25	Ye, G., Wu, C., Yue, J., Cheng, S.	2009	A QoS-aware Model for Web Services Discovery	First International Workshop on Education Technology and Computer Science
S26	Xu, B., Yan, Y.	2009	An Efficient QoS-driven Service Composition Approach for Large-scale Service Oriented Systems	IEEE International Conference on Service-Oriented Computing and Applications
S27	Bhakti, M., Abdullah, A.	2010	Towards an autonomic service-oriented architecture in computational engineering framework	10th International Conference on Information Science, Signal Processing and their Applications
S28	Jafarpour, N., Khayyambashi, M.	2009	A new approach for QoS-aware web service composition based on Harmony Search algorithm	11th IEEE International Symposium on Web Systems Evolution
S29	Kritikos, K., Plexousakis, D.	2009	Requirements for QoS-Based Web Service Description and Discovery	IEEE Transactions on Services Computing
S30	Hatvani, L., Jansen, A., Seceleanu, C., Pettersson, P.	2010	An Integrated Tool for Trade-off Analysis of Quality-of-Service Attributes	2nd International Workshop on the Quality of Service-Oriented Software Systems
S31	Zhang, W., Chang, C., Feng, T., Jiang, H.	2010	QoS-based Dynamic Web Service Composition with Ant Colony Optimization	IEEE 34th Annual Computer Software and Applications Conference
S32	Yau, S., Ye, N., Sarjoughian, H., Huang, D., Roontiva, A., Baydogan, M., Muqsith, M.	2009	Toward Development of Adaptive Service-Based Software Systems	IEEE Transactions on Services Computing
S33	Loyall, J., Gillen, M., Paulos, A., Edmondson, J., Varshneya, P., Schmidt, D., Bunch, L., Carvalho, M., Martignoni, A.	2010	Dynamic Policy-Driven Quality of Service in Service-Oriented Systems	13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing

(*continued on next page*)

Table 26 (*continued*)

Study identifier	Authors(s)	Year	Title	Source
S34	Litoiu, M., Mihaescu, M., Solomon, B. Ionescu, D.	2008	Scalable Adaptive Web Services	International Conference on Software Engineering
S35	Wang, G., Chen, A., Wang, C., Fung, C., Uczekaj, S.	2004	Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architectures	Eighth IEEE International Enterprise Distributed Object Computing Conference
S36	Ponnalagu, K., Krishnamurthy, J.	2007	Aspect-oriented Approach for Non-functional Adaptation of Composite Web Services	IEEE Congress on Services
S37	Zhang, L., Arsanjani, A., Allam, A., Lu, D., Chee, Y.	2007	Variation-Oriented Analysis for SOA Solution Design	IEEE International Conference on Services Computing
S38	Padmanabbuni, S., Majumdar, B., Chawla, M., Mysore, U.	2006	A Constraint Satisfaction Approach to Non-functional Requirements in Adaptive Web Services	International Conference on Next Generation Web Services Practices
S39	Topaloglu, N., Capilla, R.	2004	Modeling the Variability of Web Services from a Pattern Point of View	European Conference on Web Services
S40	Nanjangud C., Ponnalagu, K.	2010	Towards a Variability Model for SOA-Based Solutions	IEEE International Conference on Services Computing
S41	Alessandro, B., Cardellini, V, di Valerio, V., Iannucci, S.	2010	A Scalable and Highly Available Brokering Service for SLA-Based Composite Services	8th International Conference on Service-oriented Computing
S42	Canfora, G., di Penta, M., Esposito, R., Villani, M.	2008	A framework for QoS-aware binding and re-binding of composite web services	The Journal of Systems and Software
S43	Mokhtar, S., Preuveneers, D., Georgantas, N., Berbers, V.	2008	EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support	The Journal of Systems and Software
S44	Bleul, S., Zapf, M., Geihs, K.	2007	Flexible Automatic Service Brokering for SOAs	10th IFIP/IEEE International Symposium on Integrated Network Management
S45	Furtado, P., Santos, C.	2007	Extensible Contract Broker for Performance Differentiation	International Workshop on Software Engineering for Adaptive and Self-Managing Systems
S46	Wang, X., Huang, S., Zhou, A.	2006	QoS-Aware Composite Services Retrieval	Journal of Computer Science and Technology

Table 27
Quality scores per study.

Study identifier	Quality score							
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total
S1	1	0	0	N/A	0.5	0	0	1.5
S2	1	0.5	0	N/A	0.5	0	0	2
S3	1	0.5	0	N/A	0	0	0	1.5
S4	1	1	0	0.5	1	0.5	0.5	4.5
S5	1	1	0	0.5	1	0	0.5	4
S6	1	1	0	N/A	0.5	1	1	4.5
S7	1	1	0.5	N/A	1	0	0	3.5
S8	0.5	0.5	0	N/A	0	0.5	0.5	2
S9	1	0.5	0	N/A	1	0	1	3.5
S10	1	1	0	N/A	0.5	0	0.5	3
S11	1	0.5	0	N/A	0.5	0	0.5	2.5
S12	1	1	0	N/A	0.5	0	0	2.5
S13	1	1	0	N/A	1	0	1	4
S14	1	1	0	N/A	1	0	1	4
S15	1	0	0–0.5	N/A	1	0	1	3
S16	1	1	0.5	N/A	1	0	1	4.5
S17	1	0.5	0	N/A	1	0	0.5	3
S18	1	1	0	N/A	1	0.5	0.5	4
S19	1	1	0	N/A	1	0	0.5	3.5
S20	1	0.5	0	N/A	1	0	0.5	3
S21	1	0	0	N/A	0.5	0	0.5	2
S22	1	0.5	0	N/A	1	0	0.5	3
S23	1	0.5	1	1	1	0.5	0.5	5.5
S24	1	1	1	N/A	1	0.5	1	5.5
S25	1	0	0	N/A	0.5	0	0.5	2
S26	1	0.5	0	N/A	0.5	0.5	0.5	3
S27	1	0.5	0	N/A	0.5	0	0	2
S28	1	0.5	0	N/A	0.5	0	0.5	2.5
S29	1	1	0	N/A	1	0.5	0.5	4
S30	1	1	0	N/A	0	0	0	2
S31	1	1	0	N/A	1	0	0.5	3.5
S32	1	1	1	1	1	0	0.5	5.5
S33	1	0.5	0	N/A	1	0	0.5	3
S34	1	0.5	0	N/A	0.5	0	1	3
S35	1	1	0.5	N/A	1	0	0.5	4
S36	1	1	0	N/A	1	0	0	3
S37	1	0.5	0	N/A	0.5	0	0	2
S38	1	1	0	N/A	1	0	0	3
S39	1	1	0	N/A	1	0	0	3
S40	1	1	0	N/A	1	0	0.5	3.5
S41	1	1	0.5	N/A	1	0	0.5	4
S42	1	1	1	N/A	1	0.5	1	5.5
S43	1	1	0.5	N/A	1	0.5	1	5
S44	1	1	0	N/A	1	0	1	4
S45	1	0.5	0	N/A	0.5	0	0.5	2.5
S46	1	1	0.5	N/A	1	0	0.5	4

Table 28
Limitations of studies.

Study identifier	Limitations
S1	Applications built with this method should be resilient to changes in demand, and changes in types of supporting services and even organizational objectives
S2	Maximum potential of the method is only achievable during runtime adaptation
S3	Model includes four generic quality criteria to evaluate the QoS of web services, but it is possible to add new criteria
S6	Developed with a special focus on QoS- aware service compositions; not targeted to solve large- scale multi-party workflows with several interactions
S8	QoS composition problem is formulated based on quality levels, and therefore some analyses cannot be solved
S13	Algorithm is more applicable to the service composition which has adequate web services to select and complex process structure
S14	Only takes into account additive QoS parameters
S15	Only the most important QoS properties (e.g., failure-rate) are considered. Also, fault tolerance middleware can only work on stateless web services
S16	Extra cost
S18	Cost; only takes into account fail-stop failure model; only manages composite services whose orchestration pattern matches with predefined patterns
S20	Limited to domains that do not involve high loads or require rapid response times; only deals with "internal" contracts between roles within the organizational boundary
S24	Only simple message types are supported; only useful for one-to-one simulation (where every operation in the web service has a corresponding object in the simulation)
S29	Web service providers and requesters should use a set of prescribed tools
S34	Limited number of workloads are considered by the control scheme; all control schemes have addressed individual web services; there is no general theory on how to combine multiple autonomic loops; limited accessibility to metrics
S40	Service variant should possess the same set of "minimum required" inputs and outputs as the original service; service variant should retain the "integrity" of the original service
S42	For QoS variability further analyses are needed to assess whether the proposed binding/re-binding approach is robust enough under different network and server configurations; limited analyses of risks concerned with the increase of QoS variability for individual services

Table 29
List of papers and their rigor and relevance.

(Rigor, Relevance)	Number of papers	Study identifiers
(0,0)	3	S1, S15, S25
(0.5,0)	10	S2, S9, S11, S20, S22, S28, S33, S34, S37, S45
(1,0)	14	S5, S8, S10, S12, S13, S14, S23, S26, S36, S30, S31, S39, S40, S44
(2,0)	3	S6, S32, S43
(1.5,1)	2	S7, S16
(0.5,1)	3	S3, S17, S27
(1.5,0)	6	S4, S18, S29, S35, S41, S46
(1,1)	2	S19, S38,
(0,1)	1	S21
(2.5,1)	1	S24
(2.5,0)	1	S42

Table 30
Papers with lowest rigor and relevance evaluations.

Study identifier	QAs	Development activities	Nature of solution	Tool support	Evidence level
S1	Performance and availability and reliability	NA	Natural language (NL)	Yes	2
S2	N/A	N/A	Variability as part of a technique that models services of the system (SV)	Yes	2
S9	Performance	N/A	Formal techniques based on mathematics (FM)	–	2
S11	N/A	Implementation and Integration (II)	Natural language (NL)	–	2
S15	Reliability	Maintenance (M)	Variability as part of a technique that models the architecture of the system (AR), Formal techniques based on mathematics (FM)	Yes	4
S20	Performance	Architecture Design (ADs)	Variability as part of a technique that models the architecture of the system (AR), Natural language (NL)	Yes	2
S22	Performance and availability	Implementation and Integration (II)	Formal techniques based on mathematics (FM)	–	2
S25	Performance and availability	Architecture Design (ADs)	Variability as part of a technique that models services of the system (SV)	–	1
S28	Performance and availability and reliability	Implementation and Integration (II)	Formal techniques based on mathematics (FM),	–	1
S33	Performance and reliability	N/A	Natural language (NL)	–	1
S34	Performance and security and reliability	Architecture Design (ADs)	Variability as part of a technique that models the architecture of the system (AR)	Yes	1
S37	N/A	Architecture Design (ADs), Architecture Analysis (AA)	Variability as part of a technique that models the architecture of the system (AR), Using UML and its extensibility (UM)	–	2
S45	Performance	Implementation and Integration (II)	Natural language (NL)	–	2

Table 31
Papers with highest rigor and relevance evaluations.

	QAs	Development activities	Nature of solution	Tool support	Evidence level
S7	Performance, availability, reliability, cost	Implementation and integration, requirements	SV, FM	Yes	4
S16	Performance, availability, reliability	Architecture design	NL	–	2
S19	Performance, availability	Architecture design, implementation and integration	FM	–	2
S24	Performance, cost	Not addressed explicitly.	Simulation-based method	Yes	4
S38	Availability, cost	Architecture design	AR, FM	–	6

References

[1] F. Bachmann, P.C. Clements, Variability in Software Product Lines, SEI CMU, Pittsburgh, PA, 2005.
[2] L. Chen, M.A. Babar, N. Ali, Variability management in software product lines: a systematic review, in: 13th International Software Product Line Conference (SPLC), Carnegie Mellon University, San Francisco, CA, 2009, pp. 81–90.
[3] J. Bosch, G. Florijn, D. Greefhorst, J. Kuusela, J.H. Obbink, K. Pohl, Variability Issues in Software Product Lines, in: 4th International Workshop on Software Product Family Engineering, Springer Verlag, Bilbao, Spain, 2002, pp. 303–338.
[4] L. Chen, M.A. Babar, Variability management in software product lines: an investigation of contemporary industrial challenges, in: 14th International Software Product Line Conference, Springer Verlag, Jeju Island, South Korea, 2010, pp. 1–15.
[5] R. Hilliard, On representing variation, in: Workshop on Variability in Software Product Line Architectures, ACM, Copenhagen, Denmark, 2010, pp. 312–315.
[6] A. Kontogogos, P. Avgeriou, An overview of software engineering approaches to service oriented architectures in various fields, in: 18th International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), IEEE Computer Society, Groningen, The Netherlands, 2009, pp. 254–259.
[7] P. Brereton, N. Gold, D. Budgen, K. Bennett, N. Mehandjiev, Service-Based SYSTEMS: A Systematic Literature Review of Issues, Keele University, Staffordshire, 2005.
[8] A. Kontogogos, P. Avgeriou, Towards Modelling Variability-Intensive SOA Systems, Technical Report, University of Groningen, The Netherlands, 2009, p. 9.
[9] Q. Gu, P. Lago, Exploring service-oriented system engineering challenges: a systematic literature review, Service Oriented Computing and Applications 3 (2009) 171–188.
[10] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Keele University, 2007.

[11] J. Biolchini, P. Mian, A. Natali, G. Travassos, Systematic Review in Software Engineering, Programa de Engenharia de Sistemas e Computacao, Rio de Janeiro, Brazil, 2005.
[12] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (2007) 571–583.
[13] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering – a systematic literature review, Information and Software Technology 51 (2009) 7–15.
[14] B. Kitchenham, R. Pretorius, D. Budgen, P. Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering – a tertiary study, Information and Software Technology 52 (2010) 792–805.
[15] M. Riaz, M. Sulayman, N. Salleh, E. Mendes, Experiences conducting systematic reviews from novices' perspective, in: Evaluation and Assessment in Software Engineering (EASE 10), BCS, Keele University, UK, 2010, pp. 1–10.
[16] M. Staples, M. Niazi, Experiences using systematic review guidelines, Journal of Systems and Software 80 (2007) 1425–1437.
[17] L. O'Brien, L. Bass, P. Merson, Quality Attributes and Service-Oriented Architectures, CMU SEI, Pittsburgh, PA, 2005. p. 39.
[18] OASIS, Reference Model for Service Oriented Architecture 1.0, 2006.
[19] S. Cohen, R. Krut, Managing Variation in Services in a Software Product Line Context, CMU SEI, Pittsburgh, PA, 2010.
[20] IEEE Computer Society Software Engineering Standards Committee, IEEE Standard Glossary of Software Engineering Terminology, 1990.
[21] A. Abran, J.W. Moore, in: P. Bourque, R. Dupuis (Eds.), Guide to the Software Engineering Body of Knowledge – 2004 Version, IEEE Computer Society, Los Alamitos, CA, 2004.
[22] IEEE Computer Society Software Engineering Standards Committee, IEEE Standard for a Software Quality Metrics, Methodology, 1998.
[23] ISO/IEC, Software engineering – Product quality – Part 1: Quality model, Geneva, Switzerland, 2001.
[24] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, Addison-Wesley, Boston, MA, 2003.
[25] L. O'Brien, P. Merson, L. Bass, Quality attributes for service-oriented architectures, in: International Workshop on Systems Development in SOA Environments, IEEE Computer Society, Minneapolis, MN, 2007, pp. 1–7.
[26] A. Gehlert, A. Metzger, Quality Reference Model for SBA, Deliverable #CD-JRA-1.3.2, S-Cube, 2009, p. 64.
[27] M. Aiello, P. Bulanov, H. Groefsema, Requirements and tools for variability management, in: 4th IEEE Workshop on Requirement Engineering for Services (REFS 2010), IEEE Computer Society, Seoul, South Korea, 2010, pp. 245–250.
[28] K. Kontogiannis, G.A. Lewis, D.B. Smith, M. Litoiu, The landscape of service-oriented systems: a research perspective, in: International Workshop on Systems Development in SOA Environments, IEEE Computer Society, Minneapolis, MN, 2007, pp. 1–6.
[29] M. Svahnberg, J. van Grup, J. Bosch, A taxonomy of variability realization techniques, Software – Practice and Experience 35 (2005) 705–754.
[30] W. Anderson, What COTS and software reuse teach us about SOA, in: 6th International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, IEEE Computer Society, Banff, AB, 2007, pp. 141–149.
[31] G.C. Gannod, J.E. Burge, S.D. Urban, Issues in the design of flexible and dynamic service-oriented systems, in: International Workshop on Systems Development in SOA Environments, IEEE Computer Society, Minneapolis, MN, 2007, pp. 118–123.
[32] T. Erl, SOA Design Patterns, Prentice Hall, Upper Saddle River, NJ, 2009.
[33] S. Montagud, S. Abrahao, E. Insfran, A systematic review of quality attributes and measures for software product lines, Software Quality Journal (2011).
[34] R. Kazhamiakin, S. Benbernou, L. Baresi, P. Plebani, M. Uhlig, O. Barais, in: M.P. Papazoglou, K. Pohl, M. Parkin, A. Metzger (Eds.), Adaptation of Service-Based Systems, Springer Verlag, Berlin/Heidelberg, 2010, pp. 117–156.
[35] A.T. Endo, A. Simao, A systematic review on formal testing approaches for web services, in: 4th Brazilian Workshop on Systematic and Automated Software Testing, DIMAp, Natal, Brazil, 2010, pp. 89–98.
[36] M. Palacios, J. Garcia-Fanjul, J. Tuya, Testing in service-oriented architectures with dynamic binding: a mapping study, Information and Software Technology 53 (2011) 171–189.
[37] V. Basili, G. Caldiera, D. Rombach, The goal question metric approach, in: J.J. Marciniak (Ed.), Encyclopedia of Software Engineering, John Wiley & Sons, New York, NY, 1994, pp. 528–532.
[38] H. Zhang, M.A. Babar, On searching relevant studies in software engineering, in: Evaluation and Assessment in Software Engineering (EASE 10), BCS, Keele University, UK, 2010, pp. 1–10.
[39] A. Rainer, S. Beecham, Supplementary Guidelines, Assessment Scheme and Evidence-based Evaluations of the Use of Evidence Based Software Engineering, University of Hertfordshire, UK, 2009.
[40] M.S. Ali, M.A. Babar, L. Chen, K.-J. Stol, A systematic review of comparative evidence of aspect-oriented programming, Information and Software Technology 52 (2010) 871–887.
[41] B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius, D. Budgen, Refining the systematic literature review process – two participant-observer case studies, Empirical Software Engineering 15 (2010) 618–653.
[42] T. Dyba, T. Dingsoyr, Empirical studies of agile software development: a systematic review, Information and Software Technology 50 (2008) 833–859.
[43] L. Chen, M.A. Babar, C. Cawley, A status report on the evaluation of variability management approaches, in: 13th International Conference on Evaluation and Assessment in Software Engineering (EASE), BCS, Durham, UK, 2009, pp. 1–10.
[44] C. Hofmeister, P. Kruchten, R.L. Nord, H. Obbink, A. Ran, P. America, Generalizing a model of software architecture design from five industrial approaches, in: 5th Working IEEE/IFIP Conference on Software Architecture, IEEE Computer Society, Pittsburgh, PA, 2005, pp. 77–88.
[45] V. Alves, N. Niu, C. Alves, G. Valenca, Requirements engineering for software product lines: a systematic literature review, Information and Software Technology 52 (2010) 806–820.
[46] M. Shaw, Writing good software engineering research papers, in: 25th International Conference on Software Engineering, IEEE Computer Society, Portland, Oregon, 2003, pp. 726–736.
[47] R.K. Yin, Case Study Research – Design and Methods, Sage Publications, London, UK, 2009.
[48] C. Zannier, G. Melnik, F. Maurer, On the success of empirical studies in the international conference on software engineering, in: 28th International Conference on Software Engineering, ACM, Shanghai, China, 2006, pp. 341–350.
[49] V. Basili, R.W. Selby, D.H. Hutchens, Experimentation in software engineering, IEEE Transactions on Software Engineering 12 (1986) 733–743.
[50] R. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software Technology 44 (2002) 491–506.
[51] M. Zelkowitz, D.R. Wallace, Experimental models for validating technology, IEEE Computer (1998) 23–31.
[52] T. Dyba, T. Dingsoyr, G.K. Hanssen, Applying systematic reviews to diverse study types: an experience report, in: International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, Madrid, Spain, 2007, pp. 225–234.
[53] L. Chen, M.A. Babar, A systematic review of evaluation of variability management approaches in software product lines, Information and Software Technology 53 (2011) 344–362.
[54] M. Ivarsson, T. Gorschek, A method for evaluating rigor and industrial relevance of technology evaluations, Empirical Software Engineering 16 (2011) 365–395.
[55] K. Finney, mathematical notation in formal specification: too difficult for the masses?, IEEE Transactions on Software Engineering 22 (1996) 158–159
[56] G. Holl, P. Gruenbacher, R. Rabiser, A Systematic Review and an Expert Survey on Capabilities Supporting Multi Product Lines, Information and Software Technology 54 (2012) 828–852.
[57] R. Varadan, K. Channabasavaiah, S. Simpson, K. Holley, A. Allam, Increasing business flexibility and SOA adaption through effective SOA governance, IBM Systems Journal 47 (2008) 473–488.