

Systematic analysis of software development in cloud computing perceptions

Habib Ullah Khan¹  | Farhad Ali¹ | Shah Nazir² 

¹Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar

²Department of Computer Science, University of Swabi, Swabi, Pakistan

Correspondence

Habib Ullah Khan and Farhad Ali, Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar.

Email: habib.khan@qu.edu.qa; farhadamk@gmail.com

Funding information

Qatar University, Grant/Award Number: IRCC-2021-010

Abstract

Cloud computing is characterized as a shared computing and communication infrastructure. It encourages the efficient and effective developmental processes that are carried out in various organizations. Cloud computing offers both possibilities and solutions of problems for outsourcing and management of software developmental operations across distinct geography. Cloud computing is adopted by organizations and application developers for developing quality software. The cloud has the significant impact on utilizing the artificial complexity required in developing and designing quality software. Software developmental organization prefers cloud computing for outsourcing tasks because of its available and scalable nature. Cloud computing is the ideal choice utilized for development modern software as they have provided a completely new way of developing real-time cost-effective, efficient, and quality software. Tenants (providers, developers, and consumers) are provided with platforms, software services, and infrastructure based on pay per use phenomenon. Cloud-based software services are becoming increasingly popular, as observed by their widespread use. Cloud computing approach has drawn the interest of researchers and business because of its ability to provide a flexible and resourceful platform for development and deployment. To determine a cohesive understanding of the analyzed problems and solutions to improve the quality of software, the existing literature resources on cloud-based software development should be analyzed and synthesized systematically. Keyword strings were formulated for analyzing relevant research articles from journals, book chapters, and conference papers. The research articles published in (2011–2021) various scientific databases were extracted and analyzed for retrieval of relevant research articles. A total of 97 research publications are examined in this SLR and are evaluated to be appropriate studies in explaining and discussing the proposed topic. The major emphasis of the presented systematic literature review (SLR) is to identify the participating entities of cloud-based software development, challenges associated with adopting cloud for software developmental processes, and its significance to software industries and developers. This SLR will assist organizations, designers,

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. Journal of Software: Evolution and Process published by John Wiley & Sons Ltd.

and developers to develop and deploy user-friendly, efficient, effective, and real time software applications.

KEYWORDS

CC, cloud-based software, cloud computing, cloud deployment, cloud development, software engineering

1 | INTRODUCTION

Software engineering is concerned with applying the principles and technological knowledge for designing and developing quality software. Engineers are now capable of producing scalable, complex, distributed, and sophisticated applications due to technological improvements.¹ The development of modern software nowadays is not carried out by isolated team of engineers but generally takes place in organizations and communities that comprise massive group of crowd.^{2,3} To develop cloud-based applications, a major paradigm alteration from traditional software development is required. The prime objective of cloud computing is not just only the provision of basic functionality, that is, “only development and deployment,” but also the overall management of various infrastructure.⁴ Organizations are utilizing Web 2.0⁵ technology to employ cloud-connected workforce. With the global use of internet and communication technology, different applications are being designed and developed to enable the provision of various computing resources.⁶ Cloud computing is a “type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers.”⁷

Cloud computing enables users of networked computer systems to lease data, supply resources, and process data from a “cloud” that is managed and shared by providers organizations. This technology is gradually becoming an important asset for software developmental model.⁸ Business, information technology, and software engineering are revolutionized due to emergence in cloud computing technologies.⁹ Cloud computing merges various devices like smart gadgets, cell phones, computers, sensors, and storage devices for carrying out cost efficient operations.^{6,10} Cloud software operates in shared environments, where computer resources, performance, and accessibility might alter the real-time process due to issues among other users.¹¹ Scalable access, elasticity, and shared computing capability are the key services provided by cloud-based system. The solutions and services provided by cloud are attractive to businesses and organizations because they promise benefits like efficiency and reduction of costs.¹² Cloud computing delivers the utilization of IT resources on a little payment scheme whenever required regardless of owning them permanently. Developers and software organizations are quickly adopting this new paradigm to produce cloud-based applications.¹³ The resources on cloud are provided as a service to people and organizations.¹⁴

Cloud-based software development is an innovative field in computer science in which low cost, scalable, reliable, efficient, and effective software are developed. Cloud software is distributed on cloud and is used as internet service at possibly cheaper cost. Many cloud computing services are available, including infrastructure as a service, platform as a service, and software as a service. The rising demand for processing data, storing information, and unlimited computing infrastructure at less cost has motivated organization to shift their operations to the cloud.¹⁵ IaaS, PaaS, SaaS infrastructures, or third-party platforms can be used by cloud providers to deliver these services.¹⁴ Software deployment and provisioning through the cloud are gaining popularity. While there is a substantial amount of study on how to manage a cloud system, little is known about how software developers actually utilize clouds as well as how the cloud influences development methods.¹⁶ Cloud-based software development is a challenging and prolonged process in which various procedures, methods, and tools are adopted. Various entities are required to collaborate and communicate. Solving a complex software problem may generate another problem and, as a result, reduce the quality of software. A comprehensive literary review over the last 10 years (i.e., 2011–2021) was conducted to analyze cloud-based software engineering and its relevance in the modern corporate world. The suggested study consists of fourfold contributions represented in bullets.

- To identify and analyze the participating entities in cloud-based software.
- To identify the significant feature of cloud-based software development.
- To identify various challenges of cloud-based software development.
- To explore various existing methods or frameworks exploited for the development of cloud-based software.

The article is sequentially structured and divided into sub-sections. The general research approach conducted for this study is briefly described in Section 2. Section 3 provides outcomes of the proposed SLR. Findings and discussions of the proposed research are represented in Section 4. Section 5 deals with the research limitations and threats to validity. The conclusion and future suggestions of this research are provided in Section 6.

2 | RESEARCH METHODOLOGY

To deeply analyze the participating entities, significance, and challenges of cloud-based software and to exploit the existing research work carried on cloud-based software development, SLRs (systematic literature reviews) were conducted. SLR is defined as “a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.”¹⁷ The primary objectives for an SLR rather than an ad hoc or “at-will” literature review are to restrict the scope of the study and to assure that relevant, sufficient, and quality studies are retrieved. SLR approach is intended to prevent the partial evaluation of findings. But, however, it cannot be mitigated against biasness in primary research work. SLR includes planning, conducting, and evaluating phases. In the process of planning, a review paradigm is implemented that guides all future steps to reduce the prejudices of researchers and to structure the ambiguous data. The paradigm specifies the diverse search problems. The search strategy includes sources; the conducting period of the proposed research; rationalization for selecting specific sources; the criteria of search and other constraints; criteria for quality assurance that specifies that the research is to be included or excluded; and the process of extraction and retrieval of data along with the practices for storage of the search files and extraction of data. The proposed SLR is carried out by the collaborative activity of the authors. Various studies were analyzed, reviewed, and evaluated based on authors' judgments. Relevant studies are sorted out and evaluated based on specific criteria, in appropriate studies that do not validate the assessment criteria are excluded.

A review paradigm was implemented using the idea presented in Kitchenham and Brereton.¹⁸ The review paradigm consists of an overview of the topic selection, the research question(s), a search method, scrutiny and retrieval of articles, quality assessment, and data synthesis. The prime purpose of this SLR is to get in-depth knowledge of collaborating entities, significance, and challenges faced in developing software and to analyze the existing research techniques implemented by the engineers for cloud-based software development.

2.1 | Selecting research domain

A comprehensive analysis of the topic cloud-based software development was conducted to gain a better insight on software development via cloud. Research articles from various digital information sources were extensively studied in order to understand the idea of cloud-based software development, to identify issues in the area, and to learn what experts have done so far to tackle the identified issues arising during certain times.

2.2 | Research questions formulation

The aim of this research is fourfold, that is, to identify various cloud participating entities, significance, and challenges of cloud-based software development and to analyze the existing literature for cloud-based software development. For developing a better insight on software development phenomenon and to identify diverse research topics for researchers, various studies were examined. In the initial phase of this review, relevant articles, conference proceedings, book chapters, and journals that explicitly defined cloud-based software development activities were researched. Our pilot findings showed that cloud-based software development is an extensive activity used by many software companies to complete various software tasks and to improve software quality by leveraging the collective energy of developers. To conduct this research, the suggested research focused on answering certain research questions presented in (“research question”) Table 1 that were designed from the evaluation of numerous articles/papers in order to make this review more concise.

TABLE 1 Formulated research questions

Research questions	Description
RQ1: What are the participating entities of cloud-based software?	Focuses on the various pioneer entities of cloud-based software development. These entities are identified for the purpose to observe the overall developmental process carried out on cloud for development of quality software.
RQ2: What is the significance of cloud-based software development?	Highlights why to use cloud-based software development. The need for and importance of software development to be carried out via cloud will be briefly described
RQ3: What are various challenges of cloud-based software development?	As software development is carried out via cloud there may be various challenges faced by engineers. We will identify numerous challenges that will be point out ideas for other researchers for carrying out their research activities.
RQ4: What are the existing methods or frameworks exploited for the development of cloud-based software?	Many researchers have carried out their research on cloud-based software development; these will be explored to make the proposed study an effective one.

2.3 | Search strategy

The search strategy is composed of search keywords and the searching methods. In successive phases, the descriptions are presented.

2.3.1 | KEYWORDS STRINGS FORMULATION

The following steps are carried out for keyword formulation

- Major key terms were retrieved from proposed research questions.
- For effectiveness, the synonyms of main terms were sorted out.
- Books and articles were analyzed for keyword's formulation.
- The synonyms were lined by Boolean OR.
- Boolean AND is used for linking main terms.

By concatenation the keywords, a general query (“Cloud Based Software Engineering” OR “Software Development” AND “Importance” OR “Applications” OR “Advantages” AND “Challenges” OR “Limitations” AND “Methods” OR “Techniques” OR “Approaches”) is formed. As the string formation rules of various libraries are different, therefore, we have formulated specific string for each digital library. These are represented in Table 2.

2.4 | The searching processes

A systemic and complete evaluation of the suggested study was conducted on five digital libraries to collect data from different researchers' work in the field of cloud-based software development for synchronization this step is carried out by the second author of this paper. Wiley, IEEE Xplore, ScienceDirect, ACM Digital Library, and Springer Link are online digital libraries that were searched for extracting relevant research articles. Figure 1 shows the overall reviewing process. The title, abstract, index words of published research papers such as journals article, conference paper, and books chapters were examined for completing the proposed research. Articles are downloaded and searched from relevant libraries in order to overcome duplication of articles. The search procedure in this study comprises of various phases.

- **Phase 1:** Five digital libraries are carefully searched to obtain papers appropriate to the research being suggested. The outcome of the search was grouped as perspective studies.
- **Phase 2:** Papers are retrieved from these libraries based on keyword string.
- **Phase 3:** Appropriate studies are extracted from the digital repositories.

TABLE 2 Formulated searching string

Libraries	Search strings	Web links
ACM	“Cloud based software engineering” / “Cloud softwares” && “Applications”/ “Advantages” AND “Challenges” / “Limitations” && “Techniques/Methods/Approaches”	https://dl.acm.org/
Springer Link	“Cloud based software” OR “Cloud software development” AND “Advantages” AND “Challenges” OR “Limitations” AND “Techniques” OR “Approaches”	https://link.springer.com/
Wiley	“Cloud based software” OR “Cloud software development” AND “Advantages” AND “Limitations” AND “Techniques”	https://onlinelibrary.wiley.com/
IEEE Xplore	“Cloud based software” OR “Cloud software development” AND “Advantages” AND “Limitations” AND “Techniques” AND “Methods”	https://ieeexplore.ieee.org/Xplore/home.jsp
Science Direct	“Cloud based software” OR “Cloud software development” AND “Advantages” AND “Limitations” AND “Techniques”	https://www.sciencedirect.com/

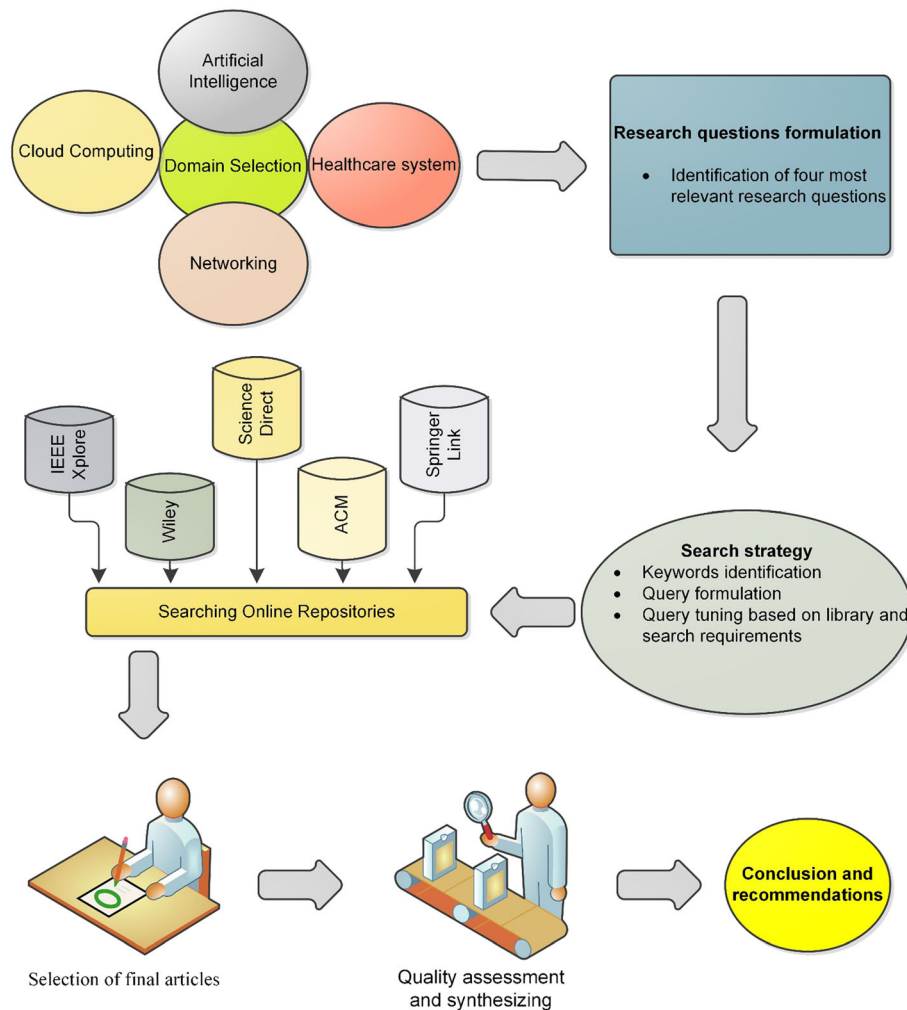


FIGURE 1 Proposed research method

TABLE 3 Number of article retrieved from digital libraries

Libraries	Total papers	Selection based on			Access date
		Title	Content	QoA	
IEEE Xplore	104	44	40	16	June 5, 2021
Wiley	247	8	7	6	June 6, 2021
ACM	290	77	49	46	June 6, 2021
Springer Link	405	36	34	19	June 7, 2021
Elsevier	452	22	20	10	June 8, 2021
Total	1,498	187	150	97	

2.5 | Scrutinization and retrieval of relevant articles

In the searching process, 1,498 research studies were retrieved from analyzing five digital databases represented in Table 3, and the percentage wise studies obtained are dissipated in Figure 2. The Initial search process retrieved meta data (title, abstract, and the contents). Figure 3 represents the year wise article distribution (2011–2021) of various libraries.

To find appropriate research articles, scrutinization (Table 4) was required because the first search yielded 1,498 randomized research. After collecting meta data, the titles of relevant studies were carefully analyzed and checked by one of the authors of this paper (third author) that will help in minimizing the articles; that is, articles that are not relevant to our research area are skipped out and appropriate ones are considered for

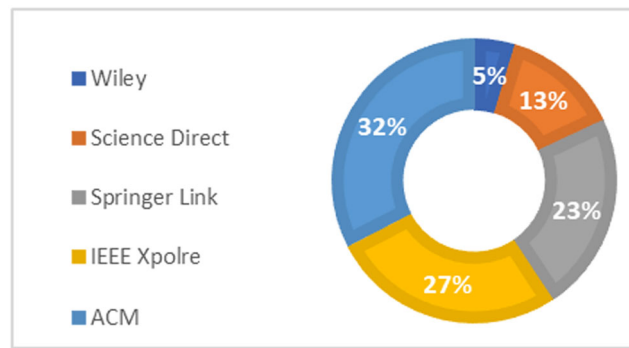


FIGURE 2 Percentage wise retrieved articles from digital libraries

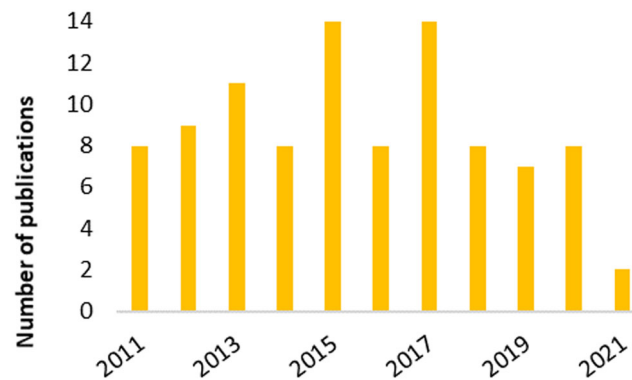


FIGURE 3 Annual distribution of the selected research studies

TABLE 4 Scrutinization criteria

Inclusion process	Exclusion process
English research studies	Articles of another languages
Paper that are relevant to cloud-based software development	Articles not relevant with research topic
Range of years (2011–2021)	Articles reported before 2011 are excluded
Papers that answer at least two proposed questions	Paper that does not answer at least two proposed questions

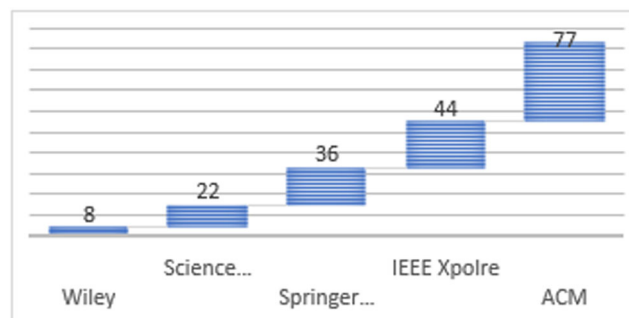


FIGURE 4 Title-based articles selection from digital libraries

further analysis. This procedure is necessary for elimination of inappropriate and identical research title-based paper selection retrieved 187 articles represented in Figure 4.

The relevant articles are then assessed accordingly with article abstract and contents that extracted 150 research articles.

2.6 | Quality of assessment and data synthesis

Based on the filtered studies obtained after scrutinization and retrieval of relevant articles, quality evaluation criteria was applied; that is, each article has been examined and thoroughly evaluated to verify if these studies answer at least two formulated questions. All the three authors of this paper equally contributed in this phase.

This phase is intended to consolidate and analyze the information to respond to the questions presented in Table 5,

Figure 5 depicts the graph of selected articles based on QoA criteria.

For the evaluation of appropriate research, a manual scoring technique was used to thoroughly select and validate discovered data. It was assured that these studies can answer specified research questions or not. Table 6 addresses such issues (i.e., answering at least two RQs). Every question is answered by only two options, that is, *yes* = 1 and *no* = 0. Each study's reliability value is computed as the total answer to the questions addressed. The validity of the suggested study was achieved through the assessment of relevant studies that have an intensity of answering two or four research questions. Accordingly, to the article score, 53 articles were eliminated after applying QoA test on abstract and content-based retrieved papers, leaving 97 publications capable of addressing at least two predefined questions. The scores representing the quality of selected articles are provided in Table 6. The RQ1 retrieves entities of cloud-based software from selected studies, while the RQ2 provides a thorough description of the significance features of cloud-based software technology. RQ3 focuses on various challenges in the development of cloud-based software. RQ4 summarizes and discusses current cloud-based software development methodologies. The evaluation of relevant articles based on assessment criteria is represented in Table 6.

3 | OUTCOMES AND DISCUSSIONS

The results of the proposed study are summarized. First, we start with reviewing the selected articles. Following that, we gave a description of the review process's results in separate subsections, in line with the questions proposed.

3.1 | Overview of selected research articles

For carrying out this research study, 97 relevant research studies are selected based on QoA criteria. This study comprised of journal articles, book chapters, and conference papers. Figure 2 shows the percentage wise retrieved articles from various digital libraries, while Figure 3 shows the annual distribution of selected research articles.

TABLE 5 Quality assessment and selection

No	Questions
1	Are the articles presents various cloud participating entities?
2	Are the studies highlights the significant features of cloud-based software development
3	Does the research studies identify the challenges of cloud-based software development
4	Does the selected studies point out the methods, approaches exploited for the development of cloud-based Software

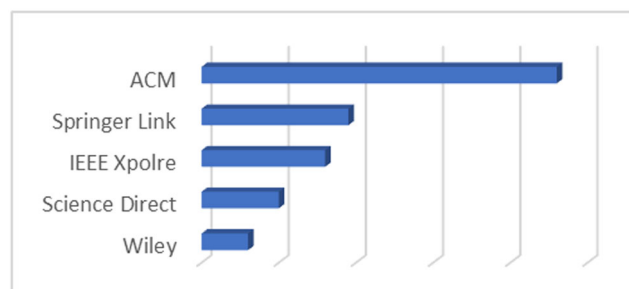


FIGURE 5 QoA-based articles selection from digital libraries

TABLE 6 Evaluation of relevant articles using assessment and selection criteria

Year	RQ1	RQ2	RQ3	RQ4	Score	References
2011	1	1	1	1	4	10
2011	1	0	1	1	3	19
2012	0	1	1	1	3	6
2012	1	1	0	1	3	20
2012	1	1	0	1	3	21
2012	0	0	1	1	2	22
2013	1	0	1	0	2	9
2013	0	0	1	1	2	11
2013	0	1	1	0	2	23
2013	0	1	1	1	3	7
2014	0	1	1	1	3	15
2015	1	0	0	1	2	24
2014	1	0	1	0	3	25
2015	1	1	1	1	4	26
2015	0	0	1	1	2	27
2015	1	1	1	0	3	28
2015	1	1	0	0	2	16
2015	1	1	1	0	3	12
2015	0	1	0	1	2	29
2018	0	1	1	0	2	30
2016	0	1	1	1	3	31
2015	1	1	1	1	4	14
2016	1	1	0	1	3	32
2017	1	0	1	0	2	33
2018	0	1	1	0	2	30
2017	1	1	0	0	2	34
2017	0	1	1	1	3	4
2017	0	1	1	0	2	35
2016	0	1	1	0	2	36
2018	0	1	1	0	2	37
2018	0	1	1	0	2	38
2018	0	1	0	1	2	39
2018	0	1	0	1	2	40
2019	0	1	1	0	2	41
2019	0	1	0	1	2	42
2019	1	1	0	1	3	43
2019	1	0	0	1	2	44
2019	0	1	0	1	2	45
2020	0	1	0	1	2	46
2020	1	1	0	1	3	47
2021	0	1	0	1	2	48
2011	1	1	0	1	3	49
2012	0	1	1	0	2	50
2014	0	1	0	1	2	51
2017	0	1	0	1	2	52
2014	0	1	0	1	2	53

TABLE 6 (Continued)

Year	RQ1	RQ2	RQ3	RQ4	Score	References
2021	1	1	1	0	3	54
2019	0	1	0	1	0	55
2017	0	1	1	0	2	56
2019	0	1	0	1	2	57
2016	0	1	1	0	2	58
2017	0	1	1	0	2	59
2020	0	1	0	1	2	60
2020	0	1	0	1	2	61
2017	0	1	1	0	2	62
2016	0	1	0	1	2	63
2013	1	1	0	0	2	64
2015	1	1	0	1	3	65
2017	0	1	1	1	3	66
2015	1	1	0	0	2	67
2011	0	1	0	1	2	68
2014	0	1	1	0	2	69
2020	1	1	0	0	2	1
2020	0	1	0	1	2	70
2017	0	1	0	1	2	71
2017	0	0	1	1	2	72
2013	0	1	0	1	2	73
2014	0	1	0	1	2	74
2016	0	1	1	0	2	75
2018	1	1	0	0	2	76
2016	1	1	1	0	3	77
2015	1	1	1	1	4	13
2013	0	1	1	1	3	78
2018	0	1	1	0	2	79
2013	0	1	1	0	2	80
2015	1	0	0	1	2	81
2015	0	1	1	0	2	82
2011	0	1	0	1	2	83
2011	0	1	1	0	2	84
2015	1	1	0	0	2	85
2015	0	1	0	1	2	86
2016	0	1	0	1	2	87
2020	0	1	0	1	2	88
2011	0	1	1	0	2	89
2014	0	1	0	1	2	90
2020	0	1	1	1	3	91
2012	0	1	0	1	2	92
2017	1	1	0	0	2	93
2012	0	1	1	0	2	8
2012	0	1	0	1	2	94
2012	1	1	0	0	2	95
2014	0	1	1	0	2	96

(Continues)

TABLE 6 (Continued)

Year	RQ1	RQ2	RQ3	RQ4	Score	References
2013	0	1	1	1	3	97
2013	0	1	0	1	2	98
2017	0	1	0	1	2	99
2013	0	1	0	1	2	100
2017	1	1	1	0	3	101

TABLE 7 Participating entities in cloud-based software

Entities	Description	References
Platform	Platform serves as a service provider and is committed to providing an optimal development environment for small and medium-sized businesses on a cloud. Security is also provided to small firms that are unable to maintain security as constrains due to low budget. Scalable and flexible cloud applications are developed on platform. Platforms are also named as cloud marketplace by researchers. Market places are promoting the competency and creativeness of cloud services.	10,14,20,85
Servers	Server has maximum resources; it accepts request generated from clients and provides a service to clients.	21
Cloud users	cloud users are people or organization who buy cloud apps that meet their requirements in terms of capability, affordability, easy management, etc. The client makes use of the cloud supplier's particular tools to create its own model or program on top of the platform.	24,49,77
Cloud providers	Cloud providers are organizations who offer various services to the platforms.	1,24,44,47
Cloud	Applications are deployed in cloud. Cloud also allows building new models for software development	25,26,34,95
Tenants	customers, end users, or service providers are known as tenants in cloud computing setup.	28,33,43
Brokers	Any individual or corporation that serves as a middleman between a cloud computing service purchaser and its vendors is a "cloud broker."	12,14
Cloud burners	Burner are hijackers who produce artificial load for starvation of resources	32
Internet	The communication and collaboration of heterogeneous entities for resource sharing, computing and storage is carried out with internet.	13,54,76,81,101
Cloud creator	They are responsible for establishing and operating a cloud service that can be managed by cloud service providers or directly exposed to cloud service consumers, depending on the circumstances.	64
Devices	Smart phones, laptops, and tablets paved the way for IT agility that led to the formation of cloud computing.	65,67
Service providers	Organization that provides on demand services to cloud entities	75
Crowd	These include people who complete various tasks. These include designers who design cloud Application. Developers who utilize their skills to develop and testers who test bugs and perform various test to ensure quality application.	9,16,19,24,85,92

3.2 | Participating entities in cloud-based software

Cloud-based development is a collaborative activity by nature, involving the combined efforts of many entities to achieve a shared goal. Collaboration and coordination between these entities are essential. This is reinforced by the decentralization and concurrency brought by new technologies and paradigms. As a result of collaboration and coordination between peoples and organizations, numerous components of the software process are developed efficiently. Various entities of cloud-based software were analyzed that are presented in Table 7.

3.3 | Significant feature of cloud-based software development

Cloud computing provides services to consumers depending on their needs and requirements. Cloud services are intended to give convenient and scalable access to cloud apps and resources and are entirely managed by a service provider. Global software system can be carried out using CC services. Most businesses organizations have shifted from collocated software development environments to global software development environments because of efficient and effective economic factors.⁵³ Cloud computing paradigm allows users to access shared configurable computing

resources. Organizations are adopting cloud-based system due to their multi characteristics. Five important characteristics explained in Hanna et al.⁵⁰ of cloud-based system are as follows:

- i. Service providers do not need to be contacted in order for users to take advantage of the cloud.
- ii. Over the internet, computing resources are accessed using standard protocols.
- iii. In cloud services' provision, a multi-tenant's paradigm can be utilized for resources sharing and distribution between users.
- iv. Computer capabilities may be easily increased or decreased based on the changing needs of users.
- v. Pay-per-use approach for computer resources is utilized by user.

Study like Cocco et al.⁹² highlighted other characteristics of cloud-based software development that are as follows:

- a. Infrastructure investment is not required immediately.
- b. For cloud services provisions of software and hardware are not necessary.
- c. Development apps operate on infrastructure that does not need to be maintained.
- d. Maximizing the usage of internal IT resources.
- e. Eliminating everything that does not provide value and just developing what is necessary.
- f. Streamlining the process of setting up and deploying apps by storing them on the cloud.
- g. Pay per use.
- h. Organizational solutions are simple to be incorporated.
- i. During all phases of software development, teams work together effectively.
- j. Faster time to market.

To identify the significance features of cloud-based software, various research articles are thoroughly studied. The features are extracted to identify the most appropriate feature so that organization or developers can enhance their capabilities for making cloud-based development effective one. Table 8 represents various significant features that are extracted from these published articles.

3.4 | Challenges of cloud-based software development

The paradigm changes of software from conventional to cloud in software development present numerous challenges when the quantity and quality were analyzed. The risk of failures affecting applications running at the cloud will increase with the wide usage of cloud computing resources. A thorough examination of the challenges faced by developers and software firms in designing and implementing cloud-based applications was carried out to assist practitioners to carefully identify and synthesize issues while developing quality software. The presented SLR has revealed that security, privacy, and other significant issues represented in Table 9 have not received much attention but are becoming increasingly important for cloud-based systems. If these challenges are properly tackled, the cloud computing will offer radical alternatives, and the potential to transform the way software is developed and managed.

3.5 | Existing methods exploited for the development of cloud-based software

Various strategies are adopted by researchers and organization to aid in the development of cloud-based software. The existing methods used by organizations and developers are highlighted in Table 10, ranging from model-driven to domain specific methods.

4 | FINDINGS AND DISCUSSIONS OF THE PROPOSED RESEARCH

The research mainly focuses on answering the formulated questions that was posed during question formulation phase. In the first analysis step, various entities of cloud computing are highlighted that make up the cloud infrastructure. These entities include various participating platforms, brokers, tenants, and services providers, and they support the efficient and successful growth procedures of organizations. Cloud computing presents both opportunities and challenges for managing software development outsourcing operations across several geographies. The research also highlights the cloud infrastructure adaptations reasons such as flexibility, scalability, fault tolerance, resources provisioning and management, virtualization, and reliable nature. Real-time, available, and accountability feature of cloud setup also attracts organization to shift towards this paradigm. Software development organizations favor cloud computing for outsourcing tasks to global people. Various challenges are also linked

TABLE 8 Significant feature of cloud-based software

Features	Description	References
Consistency	Consistency relates to how updates are propagated between copies of replicates. It is about the status of data items on multiple sites, whether they are the same. Furthermore, how people see things whether they see the same value or see various values. Consistence model is used in cloud computing for management and control of concurrent system.	10
Failure handling	Error management method is mainly important for distributed systems that are prone to failure. In cloud setup failure handling strategies are used in case of any failure in the system that re-execute the process and for ensure efficiency and effectiveness of cloud systems.	21
Cost effectiveness	Cloud testing utilizes the cloud computing infrastructure to reduce unit costs while boosting test efficiency.	665
Reliability	Software developed via cloud is reliable, and they can easy be used. Reliable nature of cloud software makes them widely accepted for various operations	20,26,46,71,79,99
Expenditure minimization	The cloud has decreased computation and operational costs through sharing resources, virtualization, minimization of maintenance, cheaper IT infrastructure, minor software costs, sharing of expertise etc. Deployment of apps, in cloud environment saves CAPEX and OPEX expenditures, regardless of their dimensions, kind or geographical location. Expenditure minimization attracts organization to use cloud systems and its applications.	15,23
Fast and efficient service	Cloud system provides efficient delivery of services as global crowd participate in carrying out various tasks that reduces tasks burdens.	7,56,64,85,87,97
Flexibility	Flexibility indicates that the system can adapt quickly and economically to any possible direct or indirect changes which influence the delivery value. The rapid evolution of organizational structure presents a range of organizational issues. Flexibility is a vital aspect in preserving the overall organizational framework for change. Cloud infrastructures are flexible in nature as they have to modify themselves according to demands of users.	6,16,26,55,66
Resources sharing	Resource sharing is an important characteristic of cloud computing. Various resources such as storage, server is shared among tenants in an accountable manner that save resources and time.	12,16,28,34,89
Energy efficiency	For cloud service providers, energy consumption has become a key problem, due to both financial and environmental considerations. Consequently, the providers of cloud services are looking for novel solutions to cut energy expenditures. Cloud system stills save energy and are efficient systems as they share various resources.	6,16,48
Fault tolerance	Fault tolerance is unpredictable and leads to unexpected system performance. Cloud systems have increased fault tolerance flexibility to mend themselves from unpredictable faults that will lower their efficiency.	16,30,43,90
Resources management	Cloud computing provide an efficient management system in terms of computing resources (such as storage, networks, and servers). Additional server instances are requested to handle increasing user demands and subsequently launch new servers when running on an elastic Infrastructure such as Windows Azure. If traffic reduces, server instances are released.	10,12,35,53,54,77
Resources availability	In cloud settings the resources are always made available to users which always fulfill the criteria of users and their applications. Appropriate techniques algorithms must be implemented to make availability of resources.	21,29,30
Virtualization	Virtualization of utilized resources in cloud settings must be improved as it enhances adaptability of service-based platforms	30
Robustness	The property to bear unpredictable condition. Cloud-based application must be able to normally operate under such situations.	30,63
Self-serviceability	In cloud system, user must be permitted to select resources according to their demands and utilize them for developing applications this is also good for minimizing various infrastructure costs.	14,30
Distributed storage	Storage must be distributed among distinct servers so that users can access or restore data unintentionally or intentionally. Cloud storage has enabled customers to access distant data repositories that are ubiquitous and easy to access from anywhere.	31,41,51,61,83
Data delivery	Data must be delivered in timely manner, algorithms or processes must be improved for provisioning of data. Instead, in investing in new technology or resources, data is provided to people via Internet in cloud setting	31
Resources	Clouds users must be provided with virtual resource for their needs.	32,38,39,78,90
Resources provision	Resources are provided as services using a distributed architecture of cloud-native services. The provision of resources will fulfill the user's needs.	30,36,74,101
Resource pooling	In cloud scenario, by utilizing a partitioning approach, the provider's computer resources are pooled to serve many clients. According to the consumer's requirements, these resources must be dynamically assigned.	34,55,68
Scalability	Humans have developed scaling, that is, the most sophisticated, efficient, and smart artifacts to cope with the competitive and unpredictable nature of shared cloud infrastructure. Scalability is crucial for applications and	4,10,36,42

TABLE 8 (Continued)

Features	Description	References
	infrastructure to handle demand spikes and minimize downtime. If the rising traffic can be managed by expanding server capacity, a service can be scalable. It is a difficulty to develop a scalable service. It must be built and intended to prevent the bottleneck effect under increasing demand, such as in centralized resources management. Partitioning, asynchrony, and replication in programming models can be utilized for attaining scalability.	
Autonomy	Autonomy is a key concern of cloud eco system. Recovery protocol must be utilized for resuming failed services, allowing the system to return to its correct execution state automatically and without the need for user intervention.	37,40
Automated recovery	It ensures the services are running and functioning correctly. If functional failure occurs, then the cloud system must automatically be recovered	40
Time management	Cloud-based development reduces time to market as developed globally.	6,67,69,102
Utilization of resources	Cloud system utilizes the resources efficiently and effectively to serve many users.	1,45,50,62,80,82
Agility	Cloud platforms must provide agility and flexibility to users.	47
Accountability	Accountability is ensured in cloud environment as resources are shared diversely with various cloud entities.	47
Assurance of services	In cloud environment the services provided must be assured for smooth operations	47
Performance	Performance of task will be increased in cloud environment due to distinct geographical crowd participation.	49,70
Ability	Many companies whose operations include storing huge volumes of data that may be needed for later usage demand cloud computing services that have an archiving function and simple access to stored data. The cloud approach must provide the ability of acquiring various resources to users.	50
Resources allocation	In cloud system resources are virtually allocated to consumers when demanded. Some strategies must be adopted for allocating resources to users.	52,53,60,88,91,100
Service selection	Cost of resource can be reduced by the selection of appropriate services and service providers.	56
Efficacy	Efficacy is a capacity to complete task to a satisfactory or desired level. By utilizing cloud resources, the tasks efficacy can be enhanced.	58
Upgradation	Online software upgradation facility must be provided for ensuring smooth operations. This upgradation takes places when one log-in to the system	58
Collaboration	By using a cloud-based structure, geographically dispersed business units may better collaborate and work together. The collaboration may be carried out by social connected clients for saving time.	58
Configurable resources	Cloud service providers are frequently able to deliver the configuration of hardware and software whenever it is required in a shared infrastructure system.	57,59
Throughput	High throughput will be achieved in cloud setup as multi tenants are involved in carrying out tasks.	60,94,97
Accessibility	Utilizing cloud concept cloud provides are given access to various resources.	62,73,77
Serviceability	Services must be provided when required by clients. User or client only pays for the services that they utilize.	13,95,103
Virtualized resources	Cloud computing is a service provisioning model in which resources are accessed and delivered using dynamically scaled and virtualized resources.	76
Multi-tasking	As cloud consists of multi-tenants so they carry out multiple tasks on cloud platforms	84
Multitude of resources	Cloud consists of multitude resources that may be utilized whenever required	86
Real-time	Cloud-based system operate in Realtime environment as development of application, completion of tasks is carried out virtually.	90
Support	Cloud-based systems can satisfy and support multi-tenants' resources requests at same time.	92
Connectivity	Cloud system provides global connectivity of peoples where organizations and people collaborate and share information with each other.	93
Concurrency	Concurrent processes are carried out on cloud that will enhances the efficiency of processes adopting cloud paradigm.	8
Rapid elasticity	Rapid elasticity permits end users to swiftly add or remove service features as required. They might also charge a fee based on utilization quantity. Cloud-based software and systems must provide elasticity in resources.	95,99
Load balancing	The processes load must be balanced in cloud-based applications for enhancing the overall process.	100

TABLE 9 Challenges of cloud-based software

Challenges	Description	References
Complex programing	Cloud-based software development runs in parallel and in distributed environment. The data are executed on several servers, data centers, and the communication is carried out with diverse end users. Therefore, programing such software is a challenging job and proper check and balance must be ensured to solve various risks.	10
Hardware reliability	Cloud software must recover from a range of hardware problems including machine crashes, disk failures and malfunction on the network. Some appropriate techniques may be utilized to ensure reliability.	19
Planning	As the software is developed virtually it may lack the proper planning process which decreases their performance.	6
Faults and errors	Cloud-based applications are complex and they may face various faults and errors. Fault and tolerance strategy should be adopted to avoid these risks.	22
Compatibility	Web browsers are the primary drivers of cloud-based solutions. In cloud echo system, there may exists device compatibility, browser compatibility, platform compatibility issues between different cloud providers as they may have no or limited interaction with other applications. Software and applications developed must operate on heterogeneous setup in cloud environment.	9,58
Resources variability	Variability in different resources exists in cloud-based applications and heterogeneous devices that impact stochastic performance, and it is difficult to be optimized.	11,25
Security	Security is the most important challenge in the cloud-based software system. The resources and information must be kept secure by utilizing cryptography, digital signatures, etc.	6,12,23,27,77,78,96,97,101
Data protection	In cloud environment the protection of data from vulnerabilities is challenging activity. The data must be protected as it is an important asset of organization.	7
SLAS	Functional or nonfunctional services provided by cloud-based software must be aligned with Service Level Agreements. Cloud software development faced many challenges as there are no proper standard and license agreements between vendors and end users. The Service Level Agreement (SLA) is considered as a key cloud security issue. Cloud providers' inability to comply with service level agreements is also a challenged faced in cloud-based settings.	6,15,27,62
Predictability	Cloud setting may be affected due to various issues such as, characteristics of resources and its visibility, predictability of resources when they are shared among other applications. Proper management of resources is necessity in cloud software systems.	25
Visibility	Cloud computing provide an efficient management system in terms of computing resources (such as storage, networks, servers, etc.). Additional server instances are requested to handle increasing user demands and subsequently launch new servers when running on an elastic Infrastructure as Windows Azure. If traffic reduces, server instances are released. The resources must be made visible for other users.	10,12,35,53,54,77
Vulnerabilities	A vulnerability is "a flaw or defect in an information system, security method, architecture, or control that might be manipulated unintentionally or on intention to break security." Cloud applications are exposed to various kinds of vulnerabilities. Such as spoofing, tampering, etc. Appropriate algorithm or techniques are mandatory for overcome these vulnerabilities.	28,31
Uncertainty	Uncertainty is caused due to heterogeneous devices and multiuser engagement in a cloud setting. Cloud software and system may be provided the ability to handle uncertain situations.	30,89
Service exchange	In cloud setting, many providers and consumers are participating in carrying out tasks; there may exist a deficiency of reasonable cloud platform that can manage a huge number of participating users to exchange services impeccably.	14
Selection issue	A cloud marketplace allows service providers to publicize their services so that users may choose the services they need. Some users can visit the centralized registration server to locate and select various services they need. Service user faces difficulty in the selection and discovery of the essential services from the platforms. Some facilitation like user friendliness must be provided so that users select appropriate service.	14
Self-adaptiveness	It is frequently important to adapt software to changing requirements to ensure proper and satisfactory results. Cloud software however lacks a proper self-adaptive mechanism. They must be provided capabilities in response to the growing number of users and cloud services. Self-adaptiveness includes self-configuring, self-protecting, self-healing, self-optimizing.	14,30,36
Information control	As data is shared with other company as a result there is limited or no control of information. They cloud organization must adopt various strategies for information handling and management.	33

TABLE 9 (Continued)

Challenges	Description	References
Latency	Latency in cloud applications increased when indirect communication between different devices takes place. This must be decreased for enhancing operations.	4
Developmental settings	As cloud-applications will be in various settings and engage with third-party applications, local development activities must be modified.	35
Privacy	Information and its processing are carried out in shared environment in which third parties also participate so there may exist data privacy issues. Some mitigations strategy should be adopted to overcome privacy issues.	37,50,80,103
Unpredictability	As the cloud processes are carried out in heterogeneous environment so the computing and communication infrastructures are unpredictable. The system or applications must be designed in a way so that it works accurately in unpredictable situations.	37
Maintainability	Due to the exponential growth of overall complexity, the quality of software products and preserving their quality are extremely crucial and difficult. A high maintenance cost for modification of cloud-based system is required when a system get damaged or become faulty.	8,37,72,82
Heterogeneity	Heterogeneity between cloud providers, networks, communicating devices, and virtual people may affect the efficiency of cloud-based software as there may exist compatibility issues.	9,38,69,84
Failures of software	Bugs may cause failure of software and Hardware and may crash the system if not analyzed and controlled.	41
Communication	Communication and coordination issues may be raised due to participation of heterogeneous entities. A proper channel may be defined for communication and coordination in cloud setup.	54
Replication	As all the process are carried out on cloud where involvement of heterogeneous entities is present. Cloud federation that is referred to a collection of cloud- enabled resources collaborating with each other may give rise to replication of information. Redundancy check should be used to avoid replications.	56
Modification	The modification in requirements can lead to the evolution of components in cloud software that must be efficiently carried out.	13,59
Fragility	Fragile means the quality of being vulnerable. So, cloud software are fragile due to its global natures.	66
Upgradation of tools	To extend the existing solutions for the future need. Upgradation of cloud software is necessary for smooth operating of various processes. If it is not carried out the system will not maintain his state towards a causing failure.	75
Standards	Cloud computing is still in its development, with a scarcity of comprehensive standards and applications.	79
Irrational resources allocation	As a result of irrational resource allocation, cloud-based software applications may suffer from degraded QoS and excessive resource expenditures.	91

with cloud-based systems such as compatibility issues, vulnerabilities, uncertainty, and privacy that creates some misconceptions between organization in adopting this approach. The proposed SLR will aid corporations, designers, and developers in developing and deploying efficient, effective, and user-friendly real-time software.

Quality software is developed in less time and at a lower cost using cloud. This review has a variety of implications for software developers.

5 | RESEARCH LIMITATIONS AND THREATS TO VALIDITY

Besides many key applications, some of the limitations of this research work are listed below:

- There are several digital libraries that contain research publications; however, in the suggested SLR, we focused on five reputed libraries. It was opted to limit the emphasis on high quality peer-reviewed publications.
- Time frame for this research was decided that ranges from 2011 to 2021 (with a portion of 2021 considered), although the articles are published daily; therefore, some latest research articles may be missed up.
- The selection of papers does not include Google Scholar. This decision was made since it offers access towards every article and journal. To save time, access and to remove redundancy just reputable journal were selected.

TABLE 10 Existing methods utilized for development of cloud-based software

Method	Description	References
Orleans	Orleans is a paradigm for software that can construct dependable, scalable, and flexible cloud applications. Its programming approach emphasizes the adoption of basic, easy to learn and utilize competing patterns. This is built on distributed actor elements, known as grains, which are discrete state units and computation which communicates using asynchronous messages. Promises are the structure for the management of asynchronous communications as well as local task-specific concurrencies. Discrete constraints and a confined model of performance allow Orleans to continue, move, reproduce, and rectify grain.	10
PREFAIL	Testing software for many failures raises the challenge of multiple failures combinatorial explosions. Prefail, a programmed failure injection tool that enables testers to develop a broad spectrum of rules to Control the huge scope of numerous failures, has been developed to address combinatorial challenge. Three cloud systems integrating (HDFS, Cassandra, and zookeeper) are utilized by Prefail. A broad range of trimming policies may be written for the period of the test acceleration. All problems that you can identify using extensive experimental testing while spending 10X-200X less time than complete testing was discovered in an experimental method with the applicable testing rules.	19
Automated testing	Software testing automation is automation of testing process of software. These involve developing and running test scripts, verifying test requirements, and using automated testing tools. Automated testing can be effective if some test process steps are automated and the resources available are allocated towards more tests. Most test cases are predicted to run at least five times in one project. Smoke tests, component tests and integration tests are particularly repetitive, making automation development very vital.	6
Free platform as a Service (FPaaS)	A multiple service platform for the development of a modular platform as a service (PaaS), is developed from free single-service application. The architecture replicates the power of cloud computing through the usage of local client installations of software as a service. The cloud platform is tailored to the user's requirements with small modifications and settings for each software service. This Free Platform as a Service (FPaaS) strategy is an initiative to illustrate the strength of cloud computing without any expense. Free Platform as a Service (FPaaS) matches the PaaS architecture features and offers flexibility, robustness, data adaptability, multi-tenancy, programmable and configurable user interfaces.	20
Software Rejuvenation	Software aging is a phenomenon in which the performance of software deteriorates over time and eventually fails. Software Rejuvenation is a proactive fault management approach to clean up the internal state of the system in order to prevent further catastrophic failures. Rejuvenation software means that the program or system is sometimes terminated, its internal state is cleaned and restarted. Rejuvenation of software is a cost-efficient response to hard failure and deterioration in performance. In an application type client server in which the server is to run permanently to provide its server customers with service, the availability of the services is increased by renewing the server process periodically during the most inactive server time. Rejuvenation and restarting a program at an earlier check point enhances the probability that the application is completed successfully in a long-term, execution of computational application.	21,70
FSaaS paradigm	A novel failure scenario as a service paradigm is introduced that is to be used across the cloud to assess the cloud applications' resilience. The cloud service providers and customers that rely on Hadoop map-reduce clusters can now employ FSaaS paradigm. FSaaS can offer services to a large variety of clients by targeting Hadoop. To enable Hadoop service providers and application vendors to test their applications against the risk of major failures on the Hadoop platform. For specific sorts of tasks, several failure scenarios were produced. Customers are enabled to pick situations in order to assess the systems.	22
Blending algorithm	Blending in the Markov random environments is a novel approach for analyzing queue network models. This family of models is simply applied to describe the performance of cloud software systems, which allows users to easily characterize uncertainties concerning the deployment environment for a random Markov environment.	11
Catbac (category-based access control)	An approach has been provided for the adoption of security cloud services. This technique allows access control models to be specified to secure access to the Cloud data. Cloud providers need to provide security methods for their consumers to safeguard them from	7

TABLE 10 (Continued)

Method	Description	References
	improper access to their sensitive Cloud server data. The approach suggested is based on catbac, a two-stage generic access control metamodel. The metamodel is developed in The first phase into an abstract model, which is completed by the cloud Provider, based on the organization's high-level policies. The second phase enables network managers at the many locations of the company to improve the created abstract model in multiple concrete models, in line with the location limitations and site specifications.	
Analytical hierarchy process	A systematic risk-conscious technique has been developed to identify barriers and to solve them by utilizing the Analytical Hierarchy Process (AHP), in a cloud adoption process. The technique builds on goal-oriented requirements engineering, the handling of barriers and AHP to systemically model and mitigate risks. The steps utilizing a non-trivial instance to show applicability are illustrated. Stability and sensitivity analyses were performed to assess the correctness of the method. The investigation focuses at evaluating the AHP resolution strategies that are capable of mitigating hazards.	15
CAFÉ	A unique system called CAFÉ was developed for cloud architectures where critical software logic is processed, even if the kernel of the operating system is penetrated, with strong confidentiality protection against piracy or reverse engineering attempts. The core mechanism is the end-to-end application execution framework which comprises of the safe encryption and delivery of secret binary software and of runtime procedures for loading, decrypting, and protecting program logic, separating them from hypervisors based virtual machines.	24
Orchestration	The orchestrator is a key framework utilized to the conventional need of cloud applications' reliability. Notably, as an untenable situation, cloud does not provide an availability and reliability of resource; hence, the need for a software solution is mandatory. An orchestration framework with a linked surveillance provides a service to define in its service description a series of alerts and associated calls, therefore allowing it a control loop to guarantee that the service is first operated as planned and scaled to the current user load correctly. In addition, as a risk management, the orchestrator can give the possibility of achieving a dependable continuous support by managing the requested routing path and identifying precisely which components of an application operate in which version of software.	26
Cloud Monatt	Cloud Monatt is built on the paradigm of the property certification and offers numerous new capabilities. First, it offers a framework to monitor several health safeties concerns. Second, it illustrates how measurements gathered may be interpreted and mapped to customer-understood security features. These links the somaticized gap between the VM characteristics sought and platform safety measures. Third, this is the first real estate attest for a VM.	27
Cerebro	A system called Cerebro can forecast the time it will take to respond to a web API when it is deployed on PaaS clouds. By using a combination of static analysis and historical performance measures, it extracts the sequence of cloud SDK calls performed by a particular web API code. Cloud SDK performance data is analyzed using QBETS, a non-parametric time series analysis and forecasting approach. Predicted response times are utilized as statistical "guarantees" with associated guaranteed probabilities. While developing and deploying a web API, Cerebro eliminates the requirement for constant API performance testing. A further benefit is that the software does not interfere with run-time operations and does not require program debugging, making it scalable.	29
Stride-dread	Microsoft's stride-dread methodology is used to analyze and measure cloud security risks and their implications in cloud systems. Threats are ranked according to their severity, and clients' security.	31
Service model	A service publication and subscription model is presented as the basis of an open cloud marketplace middleware system for cloud software services. Traditional content-based publish/subscribe paradigm is extended by an entirely new distributed publish/subscribe paradigm for service publication and subscription. For multidimensional contents, such as functionalities and behaviors matching mechanisms utilized. Using this model Users can subscribe to services in the open platform based on specifications and requirements of the services.	14
Cloud burner	To generate predictions and test them in various experimental circumstances, a mathematical model was devised that explains CPU allocation per tenant based on	32

(Continues)

TABLE 10 (Continued)

Method	Description	References
	interference from other tenants on the same VM. On the other hand, a methodology for evaluating the resource-intensiveness of cloud applications was developed and tested using a slow-down technique when a resource-intensive cloud burner was introduced.	
Pattern catalog	Three patterns are created to help guide software development teams while building applications for the cloud. Messaging System explains how messages may be propagated between services in a cluster by using a communication bus. To accurately debug unforeseen errors, preemptive logging suggests that services and their hosting servers change their log verbosity. Finally, log aggregation takes the logs produced by the previously mentioned patterns and puts them in one place where they are indexed and made available to the development team in a way that simplifies correlating events from different sources.	4
Autoscaling	On a physical machine, auto scaling is a dynamic process that changes software configurations, such as threads, connections, and cache, as well as hardware resources, such as CPU and memory, based on the current demand. The environmental conditions change throughout time. As a result of auto-scaling, cloud-based services' Service Level Agreements and budget requirements may be more easily met. For example, response time and throughput are examples of non-functional Quality of Service (QoS).	36
UML Model	A UML test model is used as a robustness testing approach based on a behavior model of the system under test. Robustness testing based on models is beneficial because it considers both the state of the system under test and its interface. With incorrect inputs and inappropriate inputs, robustness constraints are imposed in a nominal model. The UML state model was not explicitly represented in the test case generation tool. Various erroneous inputs were represented by specific input events to keep the model from becoming too big. Test cases were automatically produced by a program that employs a meta-heuristic search method.	39
Automated recovery	During the design process for cloud-based applications, two patterns are introduced. Automated recovery approaches are integral to infrastructure orchestration, and it constantly checks the execution condition of hosted services. Automatically, one of the various supervision techniques is implemented in the event of a service failure. In addition to making and performing backups, scheduler also enables the team to plan internal health checks.	40
Scale trust	SGX and Hsms are used in a hierarchical paradigm to increase performance, deployment, and usability while increasing security assurances. Our case studies demonstrate that scale trust may be deployed to security services in microservices that regularly require cryptographic operations. Lastly, our early study reveals that scale trust reduces end-to-end latency and delivers cost-effective scalability.	42
Rideaas	Rideaas, a new cloud service, has been suggested and is expected to have a major influence on cloud development. In the cloud, the development environment may be delivered as a service. Rideaas adds to the preceding concepts by combining ideaas and raas, which are complimentary. Thus, cloud end users and developers will be able to identify important components in their program, allowing them to employ fault-tolerant approaches.	43
DECIDE	This paradigm facilitates tight collaboration between software developers and IT operations with the goal of establishing continuous integration, continuous delivery, and continual development. This H2020 activity develops an enhanced DECIDE devops framework and a set of accompanying tools to assist the creation and operation of multi-cloud native applications deployed across heterogeneous cloud resources.	44
RADON	A new devops-based microservices and server less application framework is being created as part of the RADON research project. RADON contributes to performance engineering by including innovative modeling, deployment optimization, testing, and runtime management methods.	102
CANVAS	The technological possibility of building a visual simulation environment in the form of an IDE for designing and executing visual M&S applications "in the cloud" using a web browser is demonstrated. Component-based model composition is used to ease the construction of visual simulation models for traffic networks. An IDE is being developed to give computer-aided help in the composition and visualization of simulation models under a web browser on a client computer, while the simulation model is being performed on a server computer.	45

TABLE 10 (Continued)

Method	Description	References
CSLCP	Any cloud-based software is included in the proposed CSLCP paradigm. An application created from scratch and deployed in the cloud, a SaaS application, a microservice, or a mix of these might be considered cloud-based software. Prototyping takes place in a cyclic and iterative fashion. For continual improvement, the model uses iterations. Because it is cyclical, features are accumulated incrementally throughout the span of the cycle. To ensure early acceptance of the produced solution, prototyping is necessary.	46
Evaluation criteria for cloud infrastructure	Seven group criteria for assessing cloud systems are adopted from Gartner's "Solution criteria for cloud integrated IaaS and PaaS" report. An examination of the real use of Gartner's criteria was adopted that led to the development of a framework for assessing cloud platforms. Moreover, the proposed set of criteria was proven by a comparison of Google, Microsoft Azure, Amazon Web Services, and Cloud Platform solutions.	47
Benchmarking	To minimize the resource requirements of software in data centers, one must be capable of determining the utilization of software resources. For this reason, a benchmark was created to analyze the consumption of resources of data center software and ensure the energy consumption of common application types comparable among providers. The Benchmark addresses three key issues: (a) how to describe and measure software resource consumption in a comparable manner; (b) how to standardize tasks for data center software with identical characteristics; and (c) how to incentivize potential users. Using a benchmark framework and prospective collaborations, various solutions to the issues were suggested.	48
Performance modeling	Cloud capacity planning and design need performance modeling and simulation methodologies. To mimic the hardware and software components of clouds, we have developed a modular method. It allows for the quick building of new cloud models by combining simple or complex simulation modules, adding new cloud modules as needed, and changing the implementation of current modules as required. This technique may be used to create new cloud models in a short period.	49
Branch-and-bound algorithm	Using a Branch-and-Bound (B&B) algorithm, a dynamic adaptation strategy for multi-cloud applications is proposed. This technique optimizes the adaptation process itself by deciding which services to deploy inside the application. When compared to another approach that examines all configurations for customizing a program, the B&B technique was shown to be more efficient. Due to its scalable nature, this new method can handle huge configurations of multi-cloud applications, which include a high number of cloud services, without compromising on performance.	51
AURA	AURA is a cloud deployment solution that is used to deploy applications over providers that experience transitory problems (e.g., network glitches and temporary storage media issues). An application description may be deployed over the most error-prone infrastructures using AURA, and scripts can be re-executed as few times as possible.	52
MADONA	In addition, MADONA, a method for automatically developing cloud-based business applications, and RIVAL, a requirements vocabulary based on the Linked USDL principles, were presented as solutions to the problem of making application development accessible to non-IT users.	53
Cloud-based software testing service	Software unit testing may be made easier by using a CSTS that emphasizes QoS characteristics such as defect detection capabilities, resource consumption and cost. A new Hadoop and cluster customization framework is being developed to offer customers with the most valid and appropriate configuration for improved performance in terms of time and cost in addition, a clear pricing strategy was implemented that meets the expectations of clients while maximizing the net profit of service providers. The efficacy of the proposed framework is clearly demonstrated by experiments done in a cloud environment.	55
Self-learning and self-adaptive approach	Resources for cloud-based software services are allocated in a way that is self-learning and self-adaptive. This model is based on machine learning and can be trained using historical data. It can predict QoS values by utilizing workload and resource allocation information as inputs. To find an acceptable resource allocation plan utilizing the QoS model, genetic algorithms can be used to automate on-line decision-making on resource allocation. QoS model accuracy of over 90% and resource utilization improvement of 10-30% were measured using Rubis benchmark.	57
QoS prediction model and a PSO-based runtime decision algorithm	A self-adaptive resource allocation system for cloud-based software applications was developed using an iterative QoS prediction model and a PSO-based runtime decision algorithm. Using SVM, NLREG, and CART, three iterative QoS models is trained. Using	60

(Continues)

TABLE 10 (Continued)

Method	Description	References
	experimentation, it was proved that iterative QoS model is more accurate than the standard QoS model in terms of predicting service levels. As a second step iterative QoS model, was integrated with the PSO algorithm to perform on-line automated resource allocation, and proved it was proved through experiments that this method substantially surpasses the current state of the art approach for self-adaptive resource allocation	
COSD	Cloud-Based Outsource Software Development is used by Organizations to build software by assembling teams of skilled professionals from across the world and leveraging cloud computing services.	61
Test data generation framework	Test data generation frameworks that cover maximal branches and discover errors in the shortest time feasible have been suggested. It uses the pareto dominance principle to resolve conflicts between numerous objectives and finds the best optimum solution for each situation. To make the framework cloud-ready, Apache Hadoop map-reduce was used. Using Apache Hadoop map-reduce, jobs for Java API test case creation may be distributed among many mappers in a simple and cost-effective manner. Also, existing cloud-based test data creation methods were compared to our suggested framework. It is evident from the experimental results obtained on a Hadoop cluster of ten nodes that our approach is highly effective.	63
CIRANO	CIRANO is a brand-new cloud IDE for building cloud applications. CIRANO's key feature is its ability to bring together developers from various teams while structuring application development into system, package, group, and user layers, allowing for concurrent development. Furthermore, because each new module may be introduced as a new layer, this hierarchical approach makes it easier to design extendable and maintainable programs. The issue of security is addressed by limiting user access to specific layers. Aside from layer constraints, the platform also provides table-level user security via access and change permissions.	65
iobserve	The iobserve technique is presented as a solution to architectural problems in cloud-based software application dev-ops. The iobserve mega model connects architectural models in development and operations at various abstraction levels. In the MAPE control loop, iobserve uses descriptive and prescriptive architectural runtime models. Extending the iobserve mega model to live visualizations allows architectural models used in operator-in-the-loop adaptation to show both static and dynamic content.	66
AHP	A new AHP approach was designed to evaluate information systems in a cloud computing environment. The Analytical Hierarchy Process (AHP) is used to make the best decisions. AHP is utilized for weighting factors.	68
Blockchain security	Blockchain security is used for secure data transmission. Adding BCS to Online Ticketing Booking allows for extensive testing and verification of the entire system. According to the results of the experiments, BCS can provide a high level of security.	103
Software process simulation modeling	SPSM is used in software developmental processes for a variety of purposes. Utilizing SPSM the analyst identifies real-world issues in the field of software, and then uses a simulation method to build as-is/to-be models that are validated using reliable empirical data. Cloud-based project simulation provides a cost-effective approach to forecast consequences of various actions, allowing for effective and smart decision-making throughout the process.	71
Software defect detection model	The model can be used to find problematic components in large data sets of software metrics The proposed method is unique in that it can discover significant metrics by combining several filters and wrapper techniques. One of the major features of the suggested technique is that a parallel framework of a hybrid software defect predictor was built and assessed in order to cope with large amounts of software metric data in a computationally efficient manner in a cloud environment. In the parallel framework, two separate hybrids were built employing Fisher and Maximum Relevance (MR) filters with an Artificial Neural Network (ANN) based wrapper. For all parallel variants, assessments are carried out using real defect-prone software datasets.	72
STAF	Software Testing Automation Framework is a cross, multi-language method predicated on the idea of reusable services that may be utilized to automate important tasks in the testing phase of software development.	73
Model-driven architecture	For the creation of software systems, MDA uses a model-based approach. Computational Independent Model (CIM), Platform Independent Model (PIM), and Platform Specific Model are three types of models used by MDA for three levels of abstraction (PSM).	74

TABLE 10 (Continued)

Method	Description	References
	The first model defines what the system is intended to perform but hides all of the technical specifics connected to its implementation. According to the Platform Independent Model, views of the systems can be mapped to many platforms at the PSM levels. The Platform Certain Model refines the PIM with technical specifics necessary for describing how the system may use specific platform.	
Software development life cycle model for cloud software development	The Software Development Life Cycle (SDLC) for Cloud Computing is provided. The process model describe includes both the activities and the roles of the stakeholders that are involved in them. To create an SDLC for cloud environments, multiple process models were synthesized that are already mature for traditional software development. Water-fall model, prototyping, and incremental models have been merged, and a new model has been created based on their properties.	13
Cloud-based performance testing	Provides faster, efficient, on-demand access to the entire test infrastructure resources. Software, servers, storage system of unit tests and results, bandwidth of network, and security requirements for testing the application can be delivered through networks, decreasing managerial efforts and service provider involvement.	78
Weighted sum model and rule based DSS	The most popular architectural styles for creating cloud-based software-as-a-services are REST and SOAP. Because of the complexity, regular maintenance, and updates of the installed services, selecting the appropriate architectural style based on NFRs, architectural style features, and domain needs is critical. Choosing the proper architectural style, on the other hand, is a multi-criteria decision that requires the software architect to examine all the elements, as well as their relative importance. Using the weighted sum model (WSM) and rule based DSS, the suggested system addresses the problem. To obtain a suggested architectural style that is appropriate for the given domain, define ACS and NFRs criteria in the form of rules and use relative weight of importance (priority) to WSM.	81
Domain-based, decentralized framework	Users and virtualized resources can be offered and managed in IaaS using a domain-based, decentralized paradigm that was presented. Adding a new layer entitled domain to the user-resource direct mapping technique promotes decentralization of both user and resource management tasks. As a result of the architecture, domains can also manage their users and virtualized resources allocated by the IaaS provider. Scalable user and resource management, security and governance policy assistance, and price reductions are some of the advantages of a domain-based approach.	83
MOACS-coper	A unique multi-objective Ant Colony System approach is proposed to enhance cost, performance, and reliability in the cloud. The approach proposes a metaheuristics-based strategy for the multi-objective cloud-based software component deployment problem. Using a number of architectural degrees of freedom, MOACS-coper explores the search space of architecture design options to provide a collection of Pareto optimal deployment configurations. A Java-based implementation of the Non-dominant Sorting Genetic Algorithm II is provided as a comparison (NSGA-II).	86
OVMP	In order to modify the trade-off between on-demand and oversubscribed expenses, OVMP (optimal virtual machine placement) is offered. TPC-W WIPS (web interaction per second) is the primary statistic used by the TPC-W to evaluate performance limitations. Prefrail and D-cloud testing tools are used to detect the abstract information of an I/O call, as well as the injected failure, to identify every failure. D-cloud gives genuine errors in the program.	87
Particle swarm optimization and genetic algorithm	For cloud-based software services with workload windows, a resource allocation technique that adjusts to changing workloads. The proposed technique combines current and future workloads into the process of generating resource allocation plans based on QoS prediction. Next, the PSO-GA is provided for making runtime judgments for examining the goal resource allocation plan. Due to the extensive simulation testing, the recommended approach for improving resource allocation in cloud-based software applications has been proven to be successful.	88
Cloud-based testing	Organizations should carefully test their web-based apps before launching a website. It is common to do functional tests on various platforms, as well as cross-browser reliability and performance assessment. Cloud-based testing offers a bunch of reduced expenditures, on-demand flexibility, and better collaboration above traditional in-house testing. For the most part, it speeds up testing and decreases time-to-market.	90
Reinforcement learning-based resource allocation	Prediction-enabled feedback Control with Reinforcement learning based resource Allocation (PCRA) is presented as a strategy for allocating resources. First, a unique Q-	91

(Continues)

TABLE 10 (Continued)

Method	Description	References
	value prediction model is created to forecast the values of management operations (based on Q-values) in different system states. The Q-learning method integrates several prediction learners to make reliable Q-value predictions. Using a novel feedback-control based decision-making algorithm, the objective resource allocation plans may be found. Rubis assessment simulation results show that the PCRA selects resource allocation control measures with a 93.7% accuracy rate.	
System dynamic simulation model	Cloud and conventional environments methods to Global Software Development were compared using a simple system dynamics model. In order to avoid building a model that is too complex, variables that influence mostly the processes in our opinion were considered in simulator.	93
MSS optimiser	For cloud-based software applications with multi-tenant deployments, MSS optimiser proposes a QoS driven method that allows service selection. Integer Programming is used to help SaaS developers discover the best services for a multi-tenant SaaS that fulfill the QoS needs of diverse stakeholders.	94
Cloud service selection mechanism	Based on the “quality of service” (QoS) of cloud services, a unique service selection mechanism has been developed and tested in the cloud. Aside from multi-objective optimization, the proposed method can explore different cloud services depending on user provided QoS criteria.	97
CMS	For creating online applications, the Content Management System (CMS) has already shown to be a suitable solution, as it provides quick application development and ease of use. Unskilled users may generate, modify, and publish material with CMS. It is a very efficient and controllable application if CMS is used in conjunction with cloud computing technology. It was proposed to use content management systems to build cloud-based applications.	98
BRACE	BRACE, a cloud-based comprehensive, one-stop hub for software reliability tools, is introduced. In addition to an advanced analytics engine, these technologies come with an easy-to-use interface that is available as a service to developers. Because BRACE is cloud-based, it is not only well-suited for VNFs, but also makes it easier for distant teams and numerous projects to share diverse tools. An automated SRGM tool has been developed and is currently being utilized in telecom software projects with impressive accuracy and scalability.	99
shortest-job first and bandwidth awareness algorithms	Simulations are used to evaluate alternative methods under various run-time situations in a cloud environment to improve the scheduling capabilities of cloud computing software systems. a task grouping scheduling method is proposed that is integrated with shortest-job first and bandwidth awareness algorithms to decrease the waiting time and its related processing costs. As a result of the suggested scheduling method, a minimal waiting time is achieved, and deadline restrictions may be computed.	100

6 | CONCLUSION AND FUTURE SUGGESTIONS

Cloud computing is the ideal choice utilized for development modern software as they have provided a completely new way of developing real time cost effective, efficient, and quality software. Resources are provided virtually to tenants in a scalable way. The suggested study has highlighted the important research topics in cloud computing and has provided answers to some of the formulated research questions. A second consequence of the study is that future researchers and practitioners must analyze the proposed area to get relevant insights and information from it. Hence, a thorough examination of cloud-based software development is necessity. The objective of the proposed research is retrieved from different scientific studies. The primary aim of this SLR was to understand the state of the art of research in cloud-based software development and to help developers and academics in determining the literature's achievements for developing quality software. The study assists the organizations involved in the development of cloud software and identifies the significance for the development of software via the cloud. Challenges of cloud-based software were identified to develop and deploy efficient and effective software. For finding the gaps and limitations, the current methods utilized for cloud-based software development were also exploited in this study.

Analysis of significant features of cloud-based software reveals that cloud software technology is necessary for the development of modern software. Quality software are produced in less time and at a little cost using cloud infrastructures. The proposed SLR will assist organization and developer in the development of quality-based software. For the development of modern software, cloud-based software development is a necessity. Various domains such as green computing, mobile cloud computing, and energy efficient systems have a strong potential for carrying out research on cloud-based software development. Carrying out the development via cloud also give rise to various concerns as the tasks are

carried out with global crowd and in heterogeneous environment. In existing literature studies, the software development is carried out via cloud with the help of crowd workers which poses various challenges that must be identified and solved for the purpose to increase the quality of software and to minimize the cost on complex software. As no criteria is present that efficiently tackle these risks and challenges for development of quality software, therefore, a risk mitigation and management system should be adopted in future research work.

ACKNOWLEDGMENT

This work was supported by Qatar University, Doha, Qatar, Internal Grant IRCC-2021-010. This publication is supported by Qatar National Library, Doha.

DATA AVAILABILITY STATEMENT

No primary data were used to support this study. All secondary data is used and included in the reference section.

ORCID

Habib Ullah Khan  <https://orcid.org/0000-0001-8373-2781>

Shah Nazir  <https://orcid.org/0000-0003-2944-146X>

REFERENCES

- Ramachandran M, Mahmood Z. *Software Engineering in the Era of Cloud Computing*. Springer; 2020.
- Begel A, Bosch J, Storey M-A. Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder. *IEEE Software*. 2013;30:52-66.
- Tamburri DA, Lago P, Vliet H v. Organizational social structures for software engineering. *ACM Comput Surv (CSUR)*. 2013;46(1):1-35. doi:10.1145/2522968.2522971
- Sousa TB, Ferreira HS, Correia FF, and Aguiar A, Engineering Software for the Cloud: Messaging Systems and Logging, in Proceedings of the 22Nd European Conference on Pattern Languages of Programs, 2017;1-14.
- Schenk E, Guittard C. Towards a characterization of crowdsourcing practices. *J Innov Econ Manag*. 2011;93-107.
- Chana I, Rana A. Empirical evaluation of cloud-based testing techniques: a systematic review. *ACM SIGSOFT Softw Eng Notes*. 2012;37:1-9.
- Khamadja S, Adi K, and Logrippo L, Designing flexible access control models for the cloud, in Proceedings of the 6th International Conference on Security of Information and Networks, 2013;225-232.
- Grundy J, Kaefer G, Keung J, Liu A. Guest editors' introduction: software engineering for the cloud. *IEEE Software*. 2012;29(2):26-29. doi:10.1109/MS.2012.31
- Baryannis G, Garefalakis P, Kritikos K, Magoutis K, Papaioannou A, Plexousakis D, et al., Lifecycle management of service-based applications on multi-clouds: a research roadmap, in Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds, 2013;13-20.
- Bykov S, Geller A, Kliot G, Larus JR, Pandya R, and Thelin J, Orleans: cloud computing for everyone, in Proceedings of the 2nd ACM Symposium on Cloud Computing, 2011;1-14.
- Casale G, Tribastone M. Modelling exogenous variability in cloud deployments. *ACM SIGMETRICS Perform Eval Review*. 2013;40(4):73-82. doi:10.1145/2479942.2479951
- Lakka S, Michalakelis C, Stamati T, and Anagnostopoulos D, A framework for the classification of cloud computing business models, in Proceedings of the 19th Panhellenic Conference on Informatics, 2015;221-226.
- Hacaloglu T, Eren PE, Mishra D, and Mishra A, A software development process model for cloud by combining traditional approaches, in OTM Confederated International Conferences On the Move to Meaningful Internet Systems, 2015;421-430. doi:10.1007/978-3-319-26138-6_45
- Ye C, Islam A, Wei J, Chen D, Huang M, and Du W, Towards a scalable and efficient open cloud marketplace, in Proceedings of the 7th Asia-Pacific Symposium on Internetware, 2015;102-111.
- Zardari S, Bahsoon R, and Ekárt A, Cloud adoption: Prioritizing obstacles and obstacles resolution tactics using AHP, in Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014;1013-1020.
- Cito J, Leitner P, Fritz T, and Gall HC, The making of cloud applications: an empirical study on software development for the cloud, in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, 2015;393-403.
- Keele S, Guidelines for performing systematic literature reviews in software engineering, Citeseer 2007.
- Kitchenham B, Brereton P. A systematic review of systematic review process research in software engineering. *Inf Softw Technol*. 2013;55(12):2049-2075. doi:10.1016/j.infsof.2013.07.010
- Joshi P, Gunawi HS, and Sen K, PREFAIL: A programmable tool for multiple-failure injection, in Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications, 2011;171-188.
- Garcia YK and Ketel M, An economical approach to PaaS, in Proceedings of the 50th Annual Southeast Regional Conference, 2012;357-358.
- Bruneo D, Longo F, Puliafito A, Scarpa M, and Distefano S, Software rejuvenation in the cloud, in Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques, 2012;8-16.
- Faghri F, Bazarbayev S, Overholt M, Farivar R, Campbell RH, and Sanders WH, Failure scenario as a service (FSaaS) for Hadoop clusters, in Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management, 2012;1-6.
- Gusev M, Ristov S, and Donevski A, Security vulnerabilities from inside and outside the eucalyptus cloud, in Proceedings of the 6th Balkan Conference in Informatics, 2013;95-101.
- Kim CH, Park S, Rhee J, Won J-J, Han T, and Xu D, CAFE: A virtualization-based approach to protecting sensitive cloud application logic confidentiality, in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, 2015;651-656.

25. Litoiu M, Wigglesworth J, and Ng T, 5th workshop on cloud computing, in Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, 2014;288-289.
26. Edmonds A, Dudouet F, and Erne M, Reliable cloud-applications: an implementation through service orchestration, in International Workshop on Automated Incident Management in Cloud (AIMC'15), Bordeaux, France, 21-24 April 2015, 2015.
27. Zhang T and Lee RB, CloudMonatt: an architecture for security health monitoring and attestation of virtual machines in cloud computing, in Proceedings of the 42nd Annual International Symposium on Computer Architecture, 2015;362-374.
28. Ardagna CA, Asal R, Damiani E, Vu QH. From security to assurance in the cloud: A survey. *ACM Comput Surv (CSUR)*. 2015;48(1):1-50. doi:10.1145/2767005
29. Jayathilaka H, Krintz C, and Wolski R, Response time service level agreements for cloud-hosted web applications, in Proceedings of the Sixth ACM Symposium on Cloud Computing, 2015;315-328.
30. Pahl C, Jamshidi P, Zimmermann O. Architectural principles for cloud software. *ACM Trans Internet Technol (TOIT)*. 2018;18(2):1-23. doi:10.1145/3104028
31. Anand P, Ryoo J, Kim H, and Kim E, Threat assessment in the cloud environment: A quantitative approach for security pattern selection, in Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, 2016;1-8.
32. Patros SA, MacKay KB, Kent KB and Dawson M, Investigating resource interference and scaling on multitenant PaaS clouds, in Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering, 2016;166-177.
33. Brenner S, Goltzsche D, and Kapitza R, TrApps: secure compartments in the evil cloud, in Proceedings of the 1st International Workshop on Security and Dependability of Multi-Domain Infrastructures, 2017;1-6.
34. Almeida AvD, Borges MM, and Roque L, The European Open Science Cloud: a new challenge for Europe, in Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, 2017;1-4.
35. van Eyk E, Iosup A, Seif S, and Thömmes M, The SPEC cloud group's research vision on FaaS and serverless architectures, in Proceedings of the 2nd International Workshop on Serverless Computing, 2017;1-4.
36. Chen T, Bahsoon R, and Yao X, A survey and taxonomy of self-aware and self-adaptive cloud autoscaling systems, arXiv preprint arXiv:1609.03590, 2016.
37. Toutouh J, Muñoz A, and Nesmachnow S, Evolution oriented monitoring oriented to security properties for cloud applications, in Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018;1-7.
38. Buyya R, Srirama SN, Casale G, et al. A manifesto for future generation cloud computing: Research directions for the next decade. *Computing surveys*. 2018;51(5):1-38. doi:10.1145/3241737
39. Cardoso WF and Martins E, Using a search and model based framework to improve robustness tests in cloud platforms, in Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing, 2018;67-76.
40. Sousa TB, Ferreira HS, Correia FF, and Aguiar A, Engineering software for the cloud: automated recovery and scheduler, in Proceedings of the 23rd European Conference on Pattern Languages of Programs, 2018;1-8.
41. Liu H, Lu S, Musuvathi M, and Nath S, What bugs cause production cloud incidents? in Proceedings of the Workshop on Hot Topics in Operating Systems, 2019;155-162.
42. Han J, Kim S, Kim T, and Han D, Toward scaling hardware security module for emerging cloud services, in Proceedings of the 4th Workshop on System Software for Trusted Execution, 2019;1-6.
43. Cico O and Cico B, Reliable cloud software development architectures and business models case study: Rideaas and GAE launcher, in Proceedings of the 9th Balkan Conference on Informatics, 2019;1-8.
44. Alonso J, Stefanidis K, Orue-Echevarria L, Blasi L, Walker M, Escalante M, et al., DECIDE: an extended devops framework for multi-cloud applications, in Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing, 2019;43-48.
45. Önder ST and Balci O, Architecture and design of a cloud-based visual simulation environment in 2019 Winter Simulation Conference (WSC), 2019; 2737-2748.
46. Alshazly AA, ElNainay MY, El-Zoghabi AA, and Abougabal MS, A conceptual view for an enhanced cloud software life-cycle process (CSLCP) model, in Proceedings of the 2020 9th International Conference on Software and Information Engineering (ICSIE), 2020;1-5.
47. Iliashenko O, Lukianchenko E, and Lohyeeta N, A selection approach to the criteria for evaluating cloud platforms for conducting IT projects, in Proceedings of the International Scientific Conference-Digital Transformation on Manufacturing, Infrastructure and Service, 2020;1-7.
48. Schmitt N, Vobl R, Brunnert A, and Kounev S, Towards a benchmark for software resource efficiency, in Companion of the ACM/SPEC International Conference on Performance Engineering, 2021;179-182.
49. Altevogt P, Kiss T, and Denzel W, Modular performance simulations of clouds, in Proceedings of the 2011 Winter Simulation Conference (WSC), 2011;3295-3306.
50. Hanna EM, Mohamed N, and Al-Jaroodi J, The cloud: requirements for a better service, in 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), 2012;787-792.
51. Almeida A, Dantas F, Cavalcante E, and Batista T, A branch-and-bound algorithm for autonomic adaptation of multi-cloud applications, in 2014 14th IEEE/ACM International Symposium on Cluster, Cloud Grid Computing, 2014;315-323.
52. Giannakopoulos I, Konstantinou I, Tsoumakos D, and Koziris N, Aura: recovering from transient failures in cloud deployments, in 2017 17th IEEE/ACM International Symposium on Cluster, Cloud Grid Computing (CCGRID), 2017;762-765.
53. Benfenatki H, da Silva CF, Benharkat AN, and Ghodous P, Cloud Application development methodology, in 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014;13-20.
54. Akbar MA, Mahmood S, Meshram C, Alsanad A, Gumaiei A, AlQahtani SA. Barriers of managing cloud outsource software development projects: a multivocal study. *Multimed Tools Appl*. 2021. doi:10.1007/s11042-021-11245-9
55. Chawla P, Chana I, Rana A. Framework for cloud-based software test data generation service. *Softw: Pract Exp*. 2019;49:1307-1328.
56. Chauhan MA, Babar MA, Benatallah B. Architecting cloud-enabled systems: a systematic survey of challenges and solutions. *Softw: Pract Exp*. 2017; 47:599-644.
57. Chen X, Lin J, Lin B, Xiang T, Zhang Y, Huang G. Self-learning and self-adaptive resource allocation for cloud-based software services. *Concurr Comput: Pract Exp*. 2019;31:e4463.

58. Iyer GN. Cloud testing: an overview, *Encyclopedia of Cloud. Comput Secur.* 2016;327-337.
59. Pahl C, Jamshidi P, Weyns D. Cloud architecture continuity: change models and change rules for sustainable cloud software architectures. *J Softw: Evol Process.* 2017;29:e1849.
60. Chen X, Wang H, Ma Y, Zheng X, Guo L. Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model. *Future Gener Comput Syst.* 2020;105:287-296. doi:10.1016/j.future.2019.12.005
61. Akbar MA, Shameem M, Mahmood S, Alsanad A, Gumaei A. Prioritization based taxonomy of cloud-based outsource software development challenges: fuzzy AHP analysis. *Appl Soft Comput.* 2020;95:106557. doi:10.1016/j.asoc.2020.106557
62. Bahsoon R, Ali N, Heisel M, Maxim B, Mistrik I. Introduction. Software architecture for cloud and big data: an open quest for the architecturally significant requirements. In: *Software Architecture for Big Data and the Cloud*, ed. Elsevier; 2017:1-10.
63. Chawla P, Chana I, Rana A. Cloud-based automatic test data generation framework. *J Comput Syst Sci.* 2016;82(5):712-738. doi:10.1016/j.jcss.2015.12.001
64. Chung L, Hill T, Legunsen O, Sun Z, Dsouza A, Supakkul S. A goal-oriented simulation approach for obtaining good private cloud-based system architectures. *J Syst Softw.* 2013;86(9):2242-2262. doi:10.1016/j.jss.2012.10.028
65. Fylaktopoulos G, Goumas G, Skolarikis M, Sotiropoulos A, Athanasiadis D, Maglogiannis I. CIRANO: an integrated programming environment for multi-tier cloud based applications. *Procedia Comput Sci.* 2015;68:42-52. doi:10.1016/j.procs.2015.09.222
66. Heinrich R, Jung R, Zirkelbach C, Hasselbring W, Reussner R. An architectural model-based approach to quality-aware devops in cloud applications. In: *Software Architecture for Big Data and the Cloud*, ed. Elsevier; 2017:69-89.
67. Jeferry K, Kousiouris G, Kyriazis D, et al. Challenges emerging from future cloud application scenarios. *Procedia Comput Sci.* 2015;68:227-237. doi:10.1016/j.procs.2015.09.238
68. Peiyu L, Dong L. The new risk assessment model for information system in cloud computing environment. *Procedia Eng.* 2011;15:3200-3204. doi:10.1016/j.proeng.2011.08.601
69. Shahzad F. State-of-the-art survey on cloud computing security challenges, approaches and solutions. *Procedia Comput Sci.* 2014;37:357-362. doi:10.1016/j.procs.2014.08.053
70. Pietrantuono R, Russo S. A survey on software aging and rejuvenation in the cloud. *Softw Qual J.* 2020;28:7-38.
71. Akerele O. Software process simulation modelling for agile cloud software development projects: techniques and applications. In: *Strategic Engineering for Cloud Computing and Big Data Analytics*, ed. Springer; 2017:119-139.
72. Ali MM, Huda S, Abawajy J, Alyahya S, Al-Dossari H, Yearwood J. A parallel framework for software defect detection and metric selection on cloud computing. *Clust Comput.* 2017;20(3):2267-2281. doi:10.1007/s10586-017-0892-6
73. Chana I, Chawla P. Testing perspectives for cloud-based applications. In: *Software Engineering Frameworks for the Cloud Computing Paradigm*, ed. Springer; 2013:145-164.
74. Cretella G, di Martino B. An overview of approaches for the migration of applications to the cloud. *Smart Org Smart Artifacts.* 2014;67-75. doi:10.1007/978-3-319-07040-7_8
75. de Paula ACM and de Carneiro GdF, A systematic literature review on cloud computing adoption and migration, in International Conference on Evaluation of Novel Approaches to Software Engineering, 2016;222-243.
76. Ezenwoke A, Daramola O, Adigun M. QoS-based ranking and selection of SaaS applications using heterogeneous similarity metrics. *J Cloud Comput.* 2018;7:1-12.
77. Fylaktopoulos G, Goumas G, Skolarikis M, Sotiropoulos A, Maglogiannis I. An overview of platforms for cloud based development. *SpringerPlus.* 2016; 5:1-13.
78. Markande K, Murthy SJ. Leveraging potential of cloud for software performance testing. In: *Cloud Computing*, Ed. Springer; 2013:293-322.
79. Mesbahi MR, Rahmani AM, Hosseinzadeh M. Reliability and high availability in cloud computing environments: a reference roadmap. *HCIS.* 2018; 8(1):1-31. doi:10.1186/s13673-018-0143-8
80. Muppalla AK, Pramod N, Srinivasa K. Efficient practices and frameworks for cloud-based application development. In: *Software Engineering Frameworks for the Cloud Computing Paradigm*, ed. Springer; 2013:305-329.
81. Nawaz F, Mohsin A, Fatima S, and Janjua NK, Rule-based multi-criteria framework for SaaS application architecture selection, in IFIP International Conference on Artificial Intelligence in Theory and Practice, 2015;129-138.
82. Rai R, Sahoo G, Mehruz S. Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration. *SpringerPlus.* 2015; 4:1-12.
83. Shin D, Akkan H, Claycomb W, Kim K. Toward role-based provisioning and access control for infrastructure as a service (IaaS). *J Internet Serv Appl.* 2011;2(3):243-255. doi:10.1007/s13174-011-0035-x
84. Spaanenburg L, Spaanenburg H. Software-from-the-cloud. In: *Cloud Connectivity and Embedded Sensory Systems*, ed. Springer; 2011:43-81.
85. Xu X, Wu W, Wang Y, Wu Y. Software crowdsourcing for developing Software-as-a-Service. *Front Comp Sci.* 2015;9(4):554-565. doi:10.1007/s11704-015-4900-9
86. Ashraf A, Byholm B, and Porres I, A multi-objective ACS algorithm to optimize cost, performance, and reliability in the cloud, in 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015;341-347.
87. Chandu P and Kata D, Integrating and enhancing the quality of services in cloud computing with software testing, in 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016;2008-2010.
88. Chen Z, Yang L, Huang Y, Chen X, Zheng X, Rong C. PSO-GA-based resource allocation strategy for cloud-based software services with workload-time windows. *IEEE Access.* 2020;8:151500-151510. doi:10.1109/ACCESS.2020.3017643
89. Bai X, Li M, Chen B, Tsai W-T, and Gao J, Cloud testing tools, in Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE), 2011;1-12.
90. Cai J and Hu Q, Analysis for cloud testing of web application, in The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), 2014;293-297.
91. Chen X, Zhu F, Chen Z, Min G, Zheng X, Rong C. Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Trans Cloud Comput.* 2020;1117-1129. doi:10.1109/TCC.2020.2992537

92. Cocco L, Mannaro K, and Concas G, A model for global software development with cloud platforms, in 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012;446-452.
93. Ewenike S, Benkhelifa E, and Chibelushi C, Classifying collaborative approaches for cloud based collaborative software development, in 2017 International Conference on the Frontiers and Advances in Data Science (FADS), 2017;47-52.
94. He J, Han J, Yang Y, Grundy J, and Jin H, QoS-driven service selection for multi-tenant SaaS, in 2012 IEEE Fifth International Conference on Cloud Computing, 2012;566-573.
95. Incki K, Ari I, and Sözer H, A survey of software testing in the cloud, in 2012 IEEE Sixth International Conference on Software Security and Reliability Companion, 2012;18-23.
96. Leonard J, Dynamics of cloud-based software as a service in small communities of complex organizations, in 2014 47th Hawaii International Conference on System Sciences, 2014;3778-3787.
97. Mandal AK, Changder S, and Sarkar A, Selection of services for data-centric cloud applications: a QoS based approach, in 2013 2nd International Conference on Advanced Computing, Networking and Security, 2013;102-107.
98. Marimuthu C and Sekaran KC, Software development for cloud: an experiential study, in 2013 International Conference on Cloud Computing and Big Data, 2013;183-190.
99. Okumoto K, Asthana A, and Mijumbi R, Brace: cloud-based software reliability assurance, in 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2017;57-60.
100. Ru J and Keung J, An empirical investigation on the simulation of priority and shortest-job-first scheduling for cloud-based software systems, in 2013 22nd Australian Software Engineering Conference, 2013;78-87.
101. Saa P, Moscoso-Zea O, Costales AC, and Luján-Mora S, Data security issues in cloud-based Software-as-a-Service ERP, in 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), 2017;1-7.
102. Gias AU, van Hoorn A, Zhu L, Casale G, Düllmann TF, and Wurster M, Performance engineering for microservices and serverless applications: the RADON approach, in Companion of the ACM/SPEC International Conference on Performance Engineering, 2020;46-49.
103. Mayuranathan M, Murugan M, Dhanakoti V. Enhanced security in cloud applications using emerging blockchain security algorithm, *Journal of Ambient Intelligence and Humanized. Comput Secur.* 2020;1-13.

How to cite this article: Khan HU, Ali F, Nazir S. Systematic analysis of software development in cloud computing perceptions. *J Softw Evol Proc.* 2022;e2485. doi:[10.1002/smr.2485](https://doi.org/10.1002/smr.2485)