Information and Software Technology xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

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

Agile development in the cloud computing environment: A systematic review

Muhammad Younas[*,a,b], Dayang N.A. Jawawi[a], Imran Ghani[c], Terrence Fries[c], Rafaqut Kazmi[a]

[a] Universiti Teknologi Malaysia (UTM), Malaysia
[b] Government College University Faisalabad (GCUF), Pakistan
[c] Indiana University of Pennsylvania, Indiana, Pennsylvania, United States

ARTICLE INFO

Keywords:
Agile
Agile software development
Agile methodology
Cloud computing
Systematic review

ABSTRACT

Background: Agile software development is based on a set of values and principles. The twelve principles are inferred from agile values. Agile principles are composition of evolutionary requirement, simple design, continuous delivery, self-organizing team and face-to-face communication. Due to changing market demand, agile methodology faces problems such as scalability, more effort and cost required in setting up hardware and software infrastructure, availability of skilled resource and ability to build application from multiple locations. Twelve (12) principles may be practiced more appropriately with the support of cloud computing. This merger of agile and cloud computing may provide infrastructure optimization and automation benefits to agile practitioners.

Objective: This Systematic Literature Review (SLR) identifies the techniques employed in cloud computing environment that are useful for agile development. In addition, SLR discusses the significance of cloud and its challenges.

Method: By applying the SLR procedure, the authors select thirty-seven (37) studies out of six-hundred-forty-seven (647) from 2010 to 2017.

Result: The result of SLR shows that the techniques using existing tools were reported in 35%, simulations in 20% and application developed in 15% of the studies. Evaluation of techniques was reported in 32% of the studies. The impact of cloud computing was measured by the classification of four major categories such as transparency 32%, collaboration 50%, development infrastructure 29% and cloud quality attributes in 39%. Furthermore, a large number of tools were reported in primary studies. The challenges posed by cloud adoption in agile was reported as interoperability 13%, security & privacy 18% and rest of the primary studies did not report any other research gaps.

Conclusions: The study concludes that agile development in cloud computing environment is an important area in software engineering. There are many open challenges and gaps. In particular, more quality tools, evaluation research and empirical studies are required in this area.

1. Introduction

The aim of agile methodology is to help software teams, think differently, work efficiently, deliver on time, keep learning and re-learning from previous iterations. In order to achieve this aim, agile manifesto was proposed with four core agile values leading to twelve principles. The composition of agile values are as follows [37].

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

It can be seen and is already widely acknowledged that more importance should be given to the value on the left side (in bold), whereas, previously (before the introduction of this manifesto), more importance was given to the right side value. It does not mean that the right hand side values are not important. They are, however, no more important than the left side values.

- First value or feature refers to individuals or team members, customers and business people. The interaction among these individuals should occur through the face-to-face communication, self-organizing team and by tuning teams' behaviour.

* Corresponding author.
E-mail address: younas.76@gmail.com (M. Younas).

https://doi.org/10.1016/j.infsof.2018.06.014
Received 17 December 2017; Received in revised form 25 June 2018; Accepted 25 June 2018

- Second value refers to focus on working software. This implicitly suggests technical excellence, simplicity and good design.
- Third value refers to customer collaboration in the form of working closely with business people and developers together. Furthermore, customer satisfaction is achieved through early, frequent and continuous delivery of software. This continuous delivery shows the pace of development progress.
- Fourth value refers to customer satisfaction that becomes firm with accommodating the change in requirements at any stage of development and reducing the time to commit change.

According to this manifesto, twelve (12) agile principles were also proposed. Agile principles further explain and enhance the importance of agility in software development. However, these agile guidelines are not fully implemented due to rapid change in market demand and while terms are working in distributed environment. Several hindrances exist such as scalability, transparency [58], face-to-face communication [51], availability of experts [43], smooth control of development, ability to build applications from distributed locations [61] and resource management [14,58]. The changing demands require an environment to test new ideas. The provision of resources for testing news ideas increases the development cost.

In order to deal with these hindrances, cloud computing provides an environment to quickly test new ideas in the marketplace [48,53]. Cloud Computing has the potential to reduce the cost of agile development through data sharing, distributed application, prioritizing tasks and by providing infrastructure (hardware and software) [61]. Cloud computing enhances the development process by eliminating the need for installations procedures, software patches, and re-installation [50]. Cloud services provide storage and computing resource based on pay per use [17,39]. Cloud computing extends the existing agile process through fast delivery, lowering cost and increasing software quality [6,58].

As we discussed earlier, agile software development is based on frequent delivery, strong working relationship between user-developer, technical excellence and accommodating change in any stage of development. The question is how can these features of agile be implemented in a cloud computing environment. In order to answer this question, the review study will analyse agile management and development methods in conjunction with cloud computing. This analysis will explore the re-usability of various tools and interoperability of cloud services, as there are some existing studies in this area. Therefore, the SLR study first shows the growth of research in this area. The study identifies different techniques, their evaluation, comparison and use of tools in agile and cloud computing environment. The SLR study explores the effect of cloud over agile software development, issues, challenges reported in primary studies and future trends in this area.

This paper is organized into six sections. The following section introduces background and motivation of study. Section 2 describes research method to conduct SLR. Section 3 presents the concise results of the SLR and discussion. Section 4 describes the threats regarding the validity of primary studies. Finally, the Sections 5 and 6 describe future work and conclusion, respectively.

1.1. Background of agile development in cloud computing

The conjunction of agile and cloud is beneficial to distributed application development, data sharing, prioritising task, transparency and infrastructure building [58,61,66] in the sense that cloud computing affects the ecosystem of agile software development with increasing prominence [23,43,62]. Ecosystem means a system or group of interconnected elements in agile perspectives such as development environment, teams interconnectivity and the whole system. Other benefits are that cloud computing reduces cost, enables scalability and performance in agile software development [12,17]. In addition, agile's practice of communication is implemented with the help of cloud-based social technologies [23]. However, in terms of academic research, there is not much research that discusses such findings, share research gaps and related issues in both these domains, together. Separately, agile and cloud computing have enough peer-reviewed material but not on both. Therefore, this review paper focuses on the literature containing the studies on both agile methodology and cloud computing.

Table 1
List of related review studies.

Study reference	Review focus	Studies reviewed	Time covered
[24]	Overview of AGSD and cloud	8	2008–12
[2]	SLR on SE and cloud	17	2008–10

To explore more, the authors have searched online research sources to find relevant studies using agile development in cloud computing environment. In the literature, there are only two review studies on agile software development and cloud computing. The list of previously reviewed studies is given in Table 1.

The first review study [24] reported the challenges and benefits in Agile Global Software Development (AGSD) and cloud computing. This study examined eight (8) studies from 2008 to 2012. Furthermore, they included white papers, which are considered as grey material in research community, in their review [2]. They identified the challenging factors such as synchronous communication, collaboration difficulties, communication bandwidth, tool support, large team, office space, multiple sites and coordination among distributed team members. The study concluded that socio-cultural (difference of custom, lifestyle and culture on distant team members) issues exist, and that more effective cloud services are needed. They also suggested that there is a need for an artificial intelligence based service that can detect communication weaknesses and abnormalities among team members.

The second review study [2] mainly focused on the big umbrella of software engineering and cloud computing environment, not specifically agile software development. The review study discussed the problem faced by the developer and also explained the benefits of cloud computing for developing software using agile methodologies. The review study examined seventeen (17) studies from 2008 to 2010. They identified issues such as multi-tenancy, need for standardization and interoperability, issues due to concurrence execution of test cases.

First motivation point for our study is that the previous review studies have considered eight and seventeen studies from 2008 to 2010 and 2008 to 2012. We enhance the review with an increased number of articles (37) and classify the type of solutions for practicing agile development in cloud computing environment, their evaluation and comparison. Re-usability of tools in agile management and agile process implementation in conjunction with cloud is discussed. We identify the studies that addressed agile based and cloud-based features, the classification of impact due to cloud computing on agile software development, the challenges in conjunction with cloud computing and identify the research gaps in these areas.

1.2. Agile software development methodology

For the beginners, this section presents a concise introduction to agile software development and the important terms used in this area. Agile methods are based on small development cycles, continuous integration of software versions, adaptive planning, team collaboration, customer involvement, and feedback. Agile software development has various methods [60].

1.2.1. Scrum

Scrum is defined as a flexible, holistic product development strategy where developers work as a unit to reach a common goal" [44]. In Scrum, iteration is called **Sprint**, with a usual duration from one week to one month. At the beginning of project, **Sprint Planning** is started to specify and prioritize the features. The list of prioritized features is

called **Product Backlog or Sprint Backlog**. The product backlog is also called ultimate to-do list.

The daily discussion on the progress and problems of the project is done in daily stand-up meetings [66]. At the end of the iteration, team members share the performance status of iteration and identify the activities to improve the development, known as **Retrospective Meeting**. The **Scrum of Scrum** is a large group of scrums, used to scale team of large size project.

1.2.2. Extreme programming

The agile method eXtreme Programming (XP) has features such as continuous concrete feedback from short cycles of development, face-to-face communication, simplicity in design, adaptive approach to accommodate the change in business requirement, automated testing process and must identify and correct the wrong going processes. The XP enforced the use of twelve practices [33]. A small piece of task used as iteration or deliverable is called **User Story**. The requirement document is called **Story Card**. Burn-down chart predicts how quickly you and your team are burning through customer's user stories.

1.3. Dynamic systems development method (DSDM)

It is a software development framework initially used as software development method [30]. It was inspired by independent rapid application development (RAD) and promotes it. They gave the concept of vendor independent approach called Atern. There are eight principles of DSDM Atern. The famous core techniques of DSDM are timeboxing, modeling, configuration management and prototyping.

1.4. Feature driven development (FDD)

The founder of FDD was Jeff De Luca. This methodology was developed while planning a system for the Singapore Bank [54]. FDD consists of five (5) elements to explain the development methodology such as Develop Overall Model, Build Feature List, Plan by Feature, Design by Feature, and Build by Feature. Milestones are used to track the progress of each element and define the reporting state. The feature lists are reviewed and prioritized by a team member. The weekly 30-minute meeting is arranged to discuss features and then compared to track the requirements.

There are other agile methods, such as Adaptive Software Development (ASD), Kanban, Lean Development, Crystal and more. However, only the popular and more reported methods in primary studies are discussed here.

1.5. Cloud computing as a technology provider to agile software development

The brief definition of Cloud Computing by NIST [1] is as:

> "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

The cloud computing definition points out five essential components, three service and four deployment models [36]. The description of cloud services and related tools to support ADCC is given in Fig. 1. The columns show the agile artefacts and rows show the cloud services.

The cloud's deployment models are private, public, hybrid and community. Usually, enterprises and organization use a private cloud, third parties use the public cloud, the merger of the public and private cloud is called the hybrid cloud, and community cloud can be shared among different organizations [56]. Cloud computing has three service models known as Platform, Software and Infrastructure as a Service (PaaS,SaaS and IaaS) [27]. Examples of SaaS include web-based email,

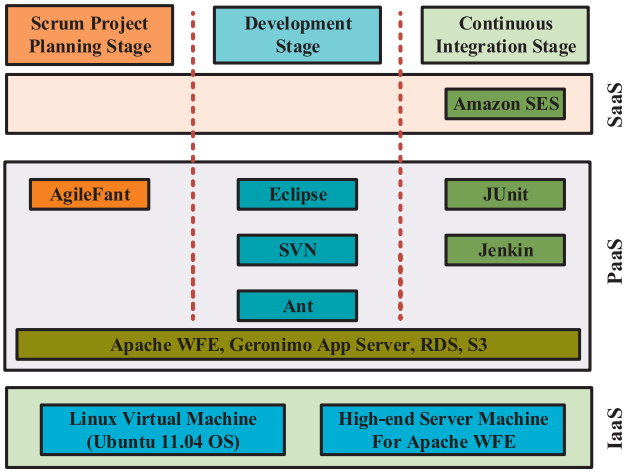

Fig. 1. Service model for scrum planning, development and integration stages by [65] - study [S17].

Microsoft 365, Google Apps and Salesforce [11]. PaaS facilitate an environment for developing, running and managing applications. PaaS allows the project manager to provide a customized environment to the developer by using cloud service. PaaS helps in deploying the software solutions. Well known PaaS services include Microsoft Azure and Google App Engine, Cloud Foundry and Heroku [11].

IaaS facilitates secure, low-cost, virtualized and scalable computing resources through the Internet [55]. This service also allows full control in hosting software, hardware, operating system, server, and data storage to the user [22]. However, it allows limited control of networking resources [42].Furthermore, IaaS provides dedicated physical server and virtual server instance [42]. Amazon web service and IBM cloud provide IaaS services.

1.6. Study approach and scope

Authors identify (37) studies by searching different publication sources. The primary studies are classified into research type, solution contribution types and explore the various kinds of approaches for solutions. The review study will answer:

1. What is the current status of cloud computing technology in practicing agile software development?
2. What types and topics of research are being focused by researchers in this area?
3. Does cloud computing matter in agile software development?
4. what is the impact of cloud computing on agile software development?
5. Why and how do cloud computing matter in agile software development, and to what extent does this matter?
6. How the current solutions use cloud computing in agile software development?
7. What are the challenges in adopting cloud computing in agile software development?
8. What is left for us to enhance agile software development using cloud computing?

Here the review study does not focus on the agile development for cloud computing, the architecture of cloud and technological discussion of cloud computing. We only focus on Agile Development in Cloud Computing environment (ADCC) or the facilitation of cloud computing that helps in agile developments. The term agile and cloud amalgamation or composition, agile development in / using cloud computing are similar. Here Agile means agile software development.

This section provides the brief introduction to agile and cloud

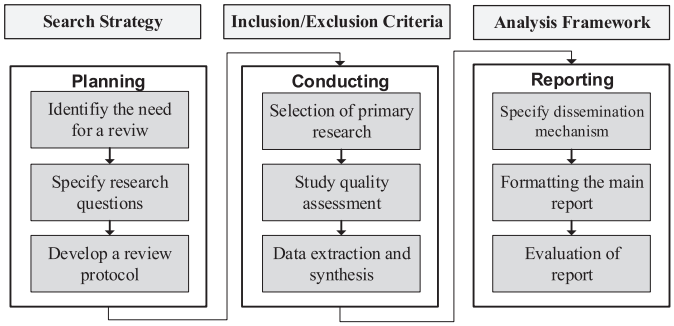

Fig. 2. Research methodology for SLR adopted by [34].

concepts. In the next section, we present the SLR protocol, identification of research questions, quality of studies and their critical analysis.

2. Systematic Literature Review

SLR is a systematic way to find the research questions (RQs), evaluate it and interpret according to the area or study of interest [8,41]. Process authenticates that the existing work is fair, seems fair and minimizes the biases in a review. According to guideline discussed by Kitchenham [34], SLR has three (3) steps: planning, conducting and documenting as shown in Fig. 2. Need of research is already described in Section 1.1. The specification of research questions is given in Table 2. The selection of primary study and search procedure is presented in Section 2.2 and inclusion-exclusion criteria are given in Table 5. Data extraction method is discussed in Section 2.4. Finally, the result of SLR is given in Section 3.

2.1. Research questions

The RQs are framed after analysing the previous review studies and discussion with domain experts. Research questions have further sub-research questions for additional clarification. The SLR study uses (Population, Intervention, Comparison, Outcome and Context) (PICOC) method to define the research questions [49]. Here **Population** is the domain of software engineering and information systems, **Intervention** is agile software development methodology and cloud computing technology, **Comparison** of primary studies on characterization, **Outcomes** are benefits, challenges, solutions and hypothesis for future research and **Context** is a structured investigation to consolidate research undertaken. The SLR study has the following research questions listed in Table 2.

2.2. Data sources and search strategy

The research database sources recommended by the studies [34,47] are used as a primary source as listed in Table 3.

The selected data sources are commonly used in the domain of software engineering and information system. The search string is executed on the databases given in Table 3. The search string is designed by starting a preliminary search to find out the volume of research material and the search string used by existing SLRs and different combination of search strings (derived from research questions). The SLR study lists the synonyms of keyword related to the subject, abbreviations and alternative spellings. The results after executing search strings on various data sources are given in Table 4.

Table 3
Database on-line sources.

Library source	URL to access
ACM Library	http://www.acm.org
IEEE Xplore	http://www.ieeexplore.ieee.org
Science Direct	http://www.sciencedirect.com
Springer	http://www.springerlink.com
Web of Science	http://www.webofknowledge.com
Wiley	http://www.onlinelibrary.wiley.com

2.3. Criteria for inclusion and exclusion of primary studies

In this phase, the studies are narrowed down to the collection of quality studies which are conceptually and thematically relevant to answer the RQs. For the reliability of primary studies included in SLR, the inclusion-exclusion criteria are shown in Table 5. The criterion is designed according to the suggestion described by the various studies [34,59].

2.4. Primary studies after applying inclusion-Exclusion criteria

The complete procedure for selection of studies is performed in four phases as depicted in Fig. 3.

The selection process is performed by examining all retrieved studies. The execution detail of search phases is given as:

Phase 1. The search string is executed on all databases listed in Table 3. The sum of all articles searched is 647, and detail of results from all data sources is given in Table 4.

Phase 2. The articles are excluded based on title. Articles that are difficult to identify based on the title are forwarded to next phase.

Phase 3. After reading the abstract, the articles are identified. Duplicate papers are also eliminated in this phase.

Phase 4. After reading the full text it is ensured that the article is relevant to our study. The inclusion-exclusion criteria (given in Table 5) is also used in this phase.

2.5. Primary study

After examining the studies, authors identified 37 primary studies selected from conference proceedings, journals, and book chapters

Table 2
Research questions.

Research question	Question statement
RQ 1	What is the state-of-art research in agile development using cloud computing environment?
RQ 1-1	What diverse type of research are identified in primary studies and their extent of contribution?
RQ 1–2	What artefacts of ADCC are reported in primary studies?
RQ 1–3	What are the solution approaches used in primary studies for agile software development in cloud computing environment?
RQ 2	What are the impacts of cloud computing on agile software development?
RQ 2-1	What are the benefits of cloud computing in agile software development?
RQ 2-2	What are the challenges faced in practicing ADCC?
RQ 3	What are the research trends and gaps in ADCC?

Table 4
Search strings, data sources and studies identified.

No.	Name	Result	Query
1	ACM	29	Agile software development AND Cloud Computing
2	IEEE	234	(Agile method OR Agile software development OR Agile process OR Agile development OR Agile model OR Scrum) AND (Cloud OR cloud computing)
3	Science Direct	115	(Agile method OR Agile software development OR Agile process OR Agile development OR Agile model OR Scrum) AND (Cloud Collaboration OR Cloud Computing)
4	Wiley	46	("Agile method" OR "Agile software development" OR "Agile process" OR "Agile development" OR "Agile model" OR scrum) AND ("Cloud Collaboration" OR"Cloud Integration" OR"Cloud Computing" OR"Service composition" OR "Service Integration" OR"Service API")
5	Springer Link	191	agile software development AND cloud computing
6	Web Of Science	32	(Agile method OR Agile software development OR Agile process OR Agile development OR Agile model OR scrum) AND (Cloud Collaboration OR Cloud Integration OR Cloud Computing)
	Total	647	

spanning 2010 to 2017. In phase 1, 647 studies are searched by executing the search string. After title-based selection, the studies reduced to 200. In phase 3, after reading the abstract and removal of duplicate studies, it short listed to 50. After full text reviewing the total studies are finalized to 37. All four phases of the selection process are described in Fig. 3. The list of identified primary studies is listed in Table 6.

This section explains the review protocol and selection process of primary studies. In the next section, we will present the results of SLR collected from the primary studies.

3. Results

This section outlines the result of SLR with respect to research questions.

3.1. RQ1: What is the state-of-art research in agile development using cloud computing environment?

The SLR categorizes primary studies into three perspectives:

1. Demographics detail.
2. Research type and contribution
3. Focus area

The demographics and evolution of studies is shown in Fig. 4.

With respect to publication, out of 37 short-listed primary studies, 51% conference papers, 35% journal articles and 14% are book chapters. Eleven (11) studies published in 2013. A gradual decline was observed in conference and book chapters as shown in Fig. 4. After demographics details of primary studies, the diverse types of research are discussed in next section.

3.1.1. RQ1-1: What diverse type of research identified in primary studies and their extent of contribution?

Primary studies are categorized according to Wieringa s classification [63], detail of contribution is given in Fig. 5.

Out of 37 primary studies, methodologies (5%), framework (24%),

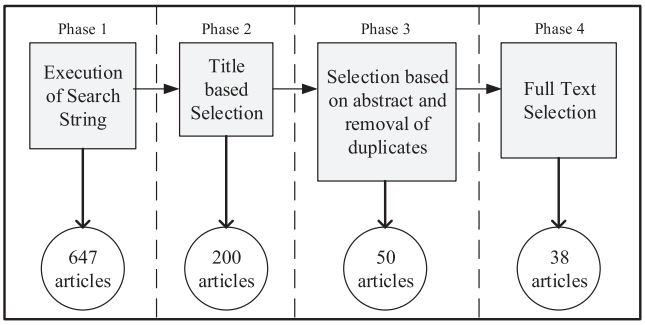

Fig. 3. Phases of selection process.

Application developed using ADCC (8%), Tools to help ADCC (8%) and the Model proposed are (27%). With respect to research type contribution case study (14%), survey paper (3%) and others are (11%). The detail of identified primary studies according to type is given in Table 6).

The statistical distribution of solution proposal studies based on contribution type is given in Fig. 5. Furthermore, the detailed discussion of solutions is coming in Section 3.1.3. Contribution with respect to agile and cloud artefacts is described in next section.

3.1.2. RQ1-2: What artefacts of ADCC are reported in primary studies?

The domain of the study is based on two aspects; the software process i.e. agile software development and the provision of technology i.e. cloud computing. Here we consider the contribution of primary studies reported regarding the agile software development features and cloud computing features. For the convenience of the reader, the description of the features is given below.

Service support. Resources provided by cloud through the Internet are called cloud services.

Scalability. It is the property of cloud which allows the user to avail peak demand successfully and avoid underutilization of computing

Table 5
Inclusion and exclusion criteria.

	Criteria	Rationale
Inclusion	Papers published after 2010.	To limit the study in scope
	The article must address Agile and Cloud in the domain of software engineering and Information system	The study focussed on Agile and cloud computing.
	Studies that purpose solution, experiment or evaluation of agile and cloud amalgamation.	Interested in specific solutions, metrics, and challenges in the agile-cloud domain.
Exclusion	Other than English language	Paper in English language
	The article must in the form of keynote, abstract, short paper or thesis	The studies do not provide a reasonable amount of information for an objective decision
	The article discusses only agile or only cloud computing perspective	The study integrates Agile methodologies and Cloud computing.

Table 6
List of primary studies.

Study ID	Reference	Artefacts/Description	Type
S1	[58]	Comparison of scrum and XP integration and benefits	OTH
S2	[23]	Case study to answer cloud help to solve challenges of agile	CS
S3	[62]	Solution for crowd-sourcing on distributed locations.	FRM
S4	[17]	User and developer's collaboration, need of end users.	SUR
S5	[51]	Web-based solution for global scrum management.	MTH
S6	[6]	Issues in a mobile-based agile development.	FRM
S7	[65]	Project management tool for agile and cloud.	TOL
S8	[5]	System dynamic model for continuous delivery of software	MOD
S9	[43]	Benefits and challenges.	FRM
S10	[31]	Prototype for managing river water-level cloud	APL
S11	[20]	Model for testing in agile and cloud.	MOD
S12	[66]	Agile and cloud connection, benefits.	CS
S13	[22]	Impact of cloud on agile.	MOD
S14	[46]	Extended agile model.	MOD
S15	[30]	Case study of warehouse project, comparison, and benefits	CS
S16	[12]	Factors of efficiency in agile-cloud development.	TOL
S17	[38]	Migration into the cloud benefits and challenges.	MOD
S18	[45]	A case study in the Erickson mobile company.	CS
S19	[42]	For migration to the cloud.	FRM
S20	[57]	Space weather application for physicists developed in agile and cloud.	APL
S21	[52]	Enhance scrum model for cloud testing. Benefits of Agile and cloud	FRM
S22	[10]	Propose a system dynamics model. Analysis by simulation of two models.	MOD
S23	[29]	How to accelerate the software development in agile and cloud	CS
S24	[32]	The impact of cloud on agile development.	FRM
S25	[9]	Web based application based on agile and cloud. Cloud service integration.	TOL
S26	[56]	Adaptive cloud development model using agile base cloud development.	MOD
S27	[4]	Agile cloud base application for drop log Air Force crew.	APL
S28	[3]	Cloud impact on software engineering	OTH
S29	[55]	The mockAPI model using agile development approach	MOD
S30	[15]	Case study to find end to end traceability in hybrid cloud computing environment	MOD
S31	[53]	Framework for automated testing and deployment	FRM
S32	[64]	A framework for agile and cloud computing, based on existing tools.	FRM
S33	[19]	System for managing test case on Jenkins server	MOD
S34	[[7]	cloud service composition	MTH
S35	[18]	Case study on mobile cloud application	OTH
S36	[16]	Overview of agile and cloud	FRM
S37	[40]	Overview of tools for ADCC	OTH

Legend:- CS: case study, MTH: methodology, FRM: framework,APL: application, MOD: model, TOL: tool, OTH: other.

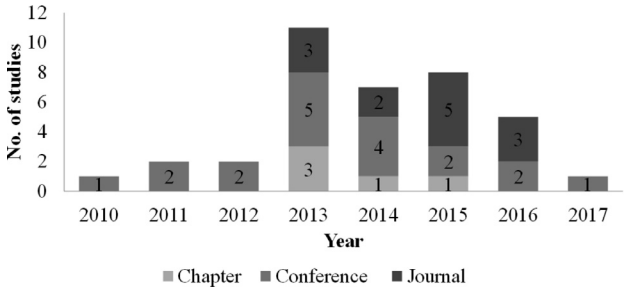

Fig. 4. Primary studies and year-wise trend.

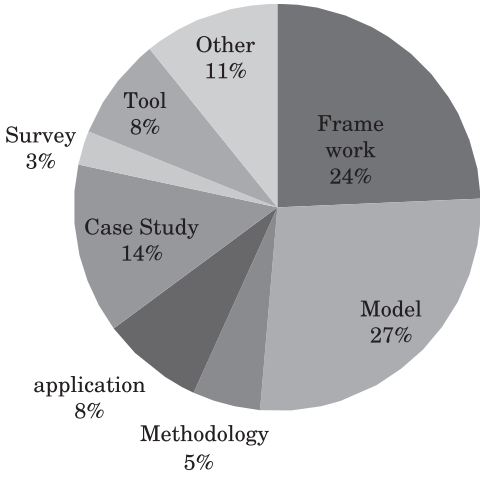

Fig. 5. Distribution of primary studies solution contributions.

resources. With respect to agile development, scalability means large team support.

Maintainability. It is a non-functional requirement in software development process. Maintainability is inversely proportional to the complexity and defects in software and directly proportional to experience and skill of software developer.

Portability. It is the property of a system that shifts it from one environment to another environment i.e. hardware, operating system, or software.

Interoperability. It is the degree of measure to which different systems or component work together.

Cloud service API. It is an application program interface which allows the user of the cloud to interact cloud provider's services

Transparency. Transparency means visible, to keep all project activities visible to everyone.

The primary studies based on cloud computing features and contribution in percentage is listed in Fig. 6. The Fig. 6 depicts that Scalability / large team support (39%), cloud service support (PaaS, SaaS, etc.) is (34%) of the primary studies. The less attention is given to portability (5%), interoperability (13%) and maintainability (3%). The contribution with respect to agile software development feature and percentage of primary studies that addressed the feature is given in Fig. 7.

The contribution with respect to agile emphasis features is Team collaboration (50%), automated testing (26%), and provision of development environment on the cloud (29%) studies. Only 3% of the total studies share the holistic view of traceability in different artefacts of software development phases. Details of other features are presented in Fig. 7. The impact of features is described in Section 3.2.1.

3.1.3. RQ1-3: What are the solution approaches used in primary studies for practicing agile software development in cloud computing environment?

Primary studies have following techniques for practicing ADCC.

Solutions using existing tools and techniques. Due to growth in the software industry and changing behaviour of business, the development process becomes speedy. Agile uses different tools for managing software development activities, such as analytics, budgeting, velocity metrics, time tracking and reporting, resource management, Kanban board, sprint management, and customer feedback. The tools for project management are **AgileFant, CloudForge**, and **Jira**. Examples

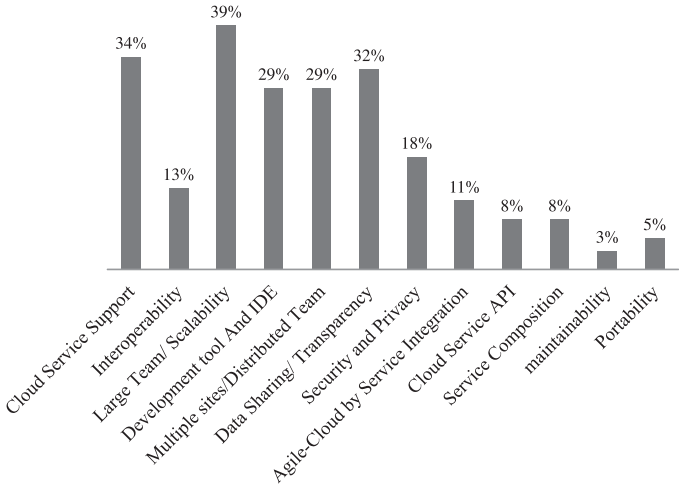

Fig. 6. Distribution with respect to cloud computing features.

of primary studies using tools to provide a solution in ADCC are listed in Table 7.

Source code repositories are used for managing code and their versions. The primary purpose of code repositories is to store, version and save the code, configuration of files and an essential part of team collaborations. Examples of source code repositories are **BitBucket** [S6], **GitHub** [S2, S3, S4, and S5], **CodeSpace, GoogleCode, SourceForge**, and **Unfuddle**.

In distributed as well as the local environment, different communication practices are used for collaboration among team members. For example, Skype-based scrum meeting [S12, S15], discussion forums, wikis, real-time reports are also used for team collaboration [S26]. Cloud computing provides support for sharing data storage and processing power to the users. The AWS-EC2 instance provides databases for sharing data [S6 and S17].

Cloud computing also provides an environment for testing (i.e., **Jenkins, XUnit**, and **Junit**) and deployment (i.e., **Puppet, Chef, Jclouds**, and **Whirr**) of software [40].

As a whole, the agile software development process is practiced with the help of various tools. A block diagram of the working mechanism is described in Fig. 8. The major artefacts are cloud platforms, Agile management tools, code repositories, collaboration and communication means, all of which are combined to form a framework to practice ADCC.

Although this type of solution increases re-usability of tools, there are some drawbacks, such as developer facing problem in adjusting to a new environment and sometimes configuration management between different artefacts become complicated. The details of complications are described in Section 3.2.2 and 5.

Conceptual/simulator based solutions. The conceptual-based solution has two sub-categories using system dynamics and using process modeling. The System Dynamics (SD) modeling usually deals with time delays, schedule, and budgeting and workload pressure. SD models analyse dynamic variables that changed over time. Three studies [S8, S16, and S22] adopt SD modeling in their solution.

A study [S8] proposed a model for effortless and risk-free Continuous Delivery (CD) of software. They validate the simulation first with structural validation i.e. with internal variables that have an environmental, human and technological effect on CD and with behaviour validation i.e. the response of developers on the duration of the test and builds. For other validation they used behaviour validation, taking data from real-time projects for comparison with simulated results.

Other studies [S16 and S22] proposed an efficient tool for making a strategic decision in distributed agile development. The study represents a solution in the form of level, flow , and auxiliary variables. Delay variables consider planning time, building time, time to deploy skilled resource and time to set infrastructure hardware and software as shown in Fig. 9. The study [S22] evaluates the model on Vensim simulation modeling tools.

The study [S13] proposes extreme programming model for cloud that has two roles, software developer and cloud provider. The study

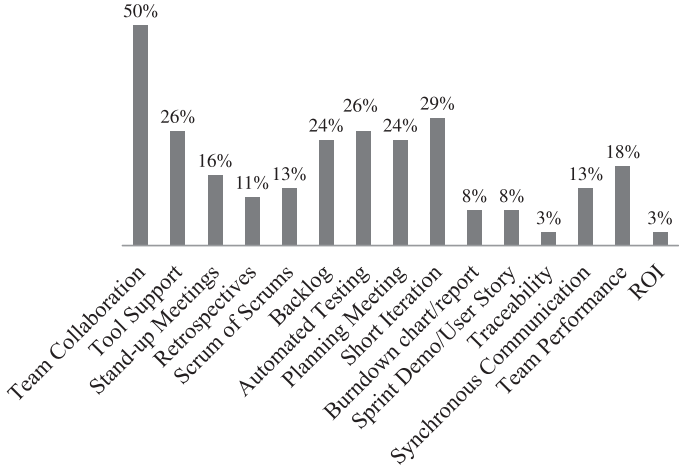

Fig. 7. Distribution with respect to agile software development features.

Table 7
Primary studies using existing tools for solution proposal.

Study	Team Collaboration	Agile / cloud support	Project management	Code repository / Development IDE	Agile model	Evaluation
S3	Confluences	Chef / puppet	Cloud Spoke on Top coder	GitHub, Cloud IDE	-	Pilot Project
S5	-	mongoDB	-	Ruby on Rails	Agile	-
S12	Skype	Cloud Forge	-	Eclipse	-	case study
S15	Skype	Google App Engine	-	Eclipse	DSDM	case study warehouse
S17	Cloud Telephony	EC2-AWS	agileFant	-	Agile	migration from agile to cloud
S26	Discussion forum, wikis, instant reports	Team Forge, Cloud Forge	Team Forge	Eclipse, Visual Studio	Agile	-
S35	-	Google App Engine	-	Eclipse IDE	-	Survey

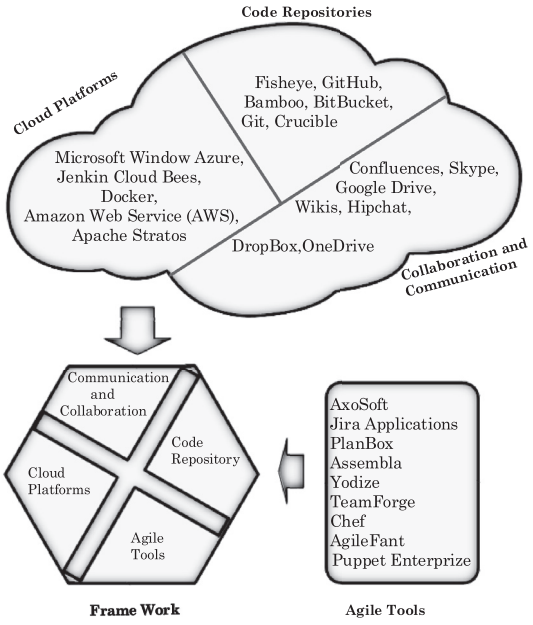

Fig. 8. Generalized work mechanism for ADCC by our study [S32].

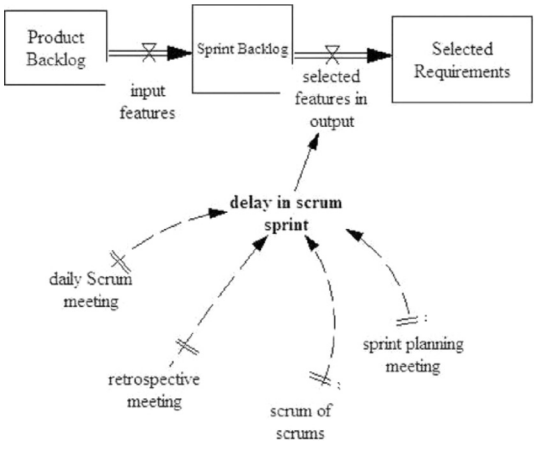

Fig. 9. Delay due to meetings in agile development [S22].

estimates the cost of effort applied and time consumed in cloud computing environment. Another study [S4] presents the extended agile software development model. The study highlights the role of software engineer and cloud providers in agile software development lifecycle. Indeed, the studies [S13 and S14], both share the same concept of process. It is a little different than study [S14] that discusses only challenges and [S13] also presents enhanced Constructive Cost Model (COCOMO) model to calculate effort.

Conceptual solutions are chosen by the researchers because analysis of development environments is very complex [S22]. The conceptual modeling is an easy way to design a solution [S16]. However, the real-life projects show the real challenges in the system. The list of solutions and their significant artefacts are given in Table 8.

Applications developed using ADCC. Some studies [S10, S20, S25, and S27] share their experiences while developing an application in agile and cloud computing environment. Study [S10] develops a prototype for managing rivers water level, where users are government agencies and city leaders. The major artefacts used are Google App Engine, Google Map API, SQL Server and Selenium IDE. Study [S20] develops a web-based application Cloud-Enabled Space Weather Platform (CESWP). They conclude that the CESWP reduces the problems such as selection and configuration of operating systems, specification and maintenance of hardware, and the configuration of tools used while building a model in space weather research.

Another study [S27] develops MAFFS Drop Log Application using Software as a Service (SaaS). This system is used to track the fire-fighting activities. The system uses Microsoft Azure and agile development methodology. Another study [S25] proposes a solution to develop community applications for computing research. They use Ruby on Rail web application and MongoDB for data sharing and software as a service (SaaS). List of solutions and major artefacts used are given in Table 9.

It is observed that in earlier studies, other means (eucalyptus) are used instead of known cloud services. However, in newer studies, primary focus is on the facilitation of cloud services. Only two studies [S20 and S25] validate the applications. However, overall satisfaction level from the user is acceptable. Furthermore, all four applications are web-based.

Build tool/Environment for ADCC. Another type of solution is by developing some tool / environment to support and facilitate ADCC as shown in Table 10. The study [S6] presents a solution ShaMoCloud designed in 3-tier architecture and implemented in a hybrid cloud environment. The ShaMoCloud uses Amazon EC2 cloud and SQL database for storage. They evaluate their work by comparing with Team Software Process (TSP) that is developed using waterfall model. They also compare with DSDM model in the study [S15].

Another study [S7] designs a model and tool for agile software project management using Scrum methodology. Project management layer explains how the project distributed tasks into sprints and show their transition using state diagram. The model has two parts, TOAST client and TOAST server. They validate the MOAT model by a web-based project Cooperative Sale System of Car Insurance (CSCI). Their study is applied and briefly explains the working mechanism of a management tool and their model with examples. However, this study does not focus on code versioning. There is less information on how to use the development environment in cloud computing or on the local system.

Table 8
Primary studies using simulation approach.

Study	Basic Type	Major factors under discussion	Modeling approach	Agile process	Evaluation nature	Results
S8	Model	Continuous delivery of S/W, effortless, risk-free	System dynamics	Agile	Structural and behaviour validation	-
S13	Model	developer and cloud provider role, Time estimation	Process modeling	Extended XP	COCOMO Model for comparing cost	S/w dev. With cloud computing consume less time
S14	Model	Impact of cloud on software development,Cost,complexity	Software process modeling	Extended agile	-	Software complexity increase due to communication
S16	Tool	Effort, time, performance	System dynamics	Scrum	-	Proposed a tool
S22	Model	Error and delay using vensim simulator	System dynamics	Scrum	Compare traditional versus cloud solutions	Achieve high development rate with cloud computing

Solution by adoption / Migration into cloud. Some studies [S17, S18, S19 and S24] adopt cloud computing in their organization. Study [S19] identifies the factors that affect the process of migration into the cloud and the relation between the factors that help in the success of migration such as sharing application or tools, communication and coordination. The study highlights the requirements for the framework such as selection of agile methodology, cloud service and role of stakeholders both inside and outside of the organization.

Study [S17] discusses the issues in the adoption of cloud in agile scrum development. They compare cost, time and quality with and without cloud. They conclude that agile-cloud is better with respect to cost and time. On-premise agile development is better in terms of quality and security.

Study [S24] depicts the issues regarding distributed teams such as lack of trust due to culture difference among team members and time-based issues. Study [S18] conducts a case study in Erickson R&D organization. They organize 32 interviews and four observation sessions at three different sites during 2013–2014. They identify that the classification of application with respect to security is important before cloud migration. Furthermore, the team should know the effort required to coordinate and communicate with cloud service provider. This type of solutions is closer to real-time implementation.

The synthesis of existing solution's rationale shows that most solutions are based on using existing tools. The contribution of all solution techniques is shown in Fig. 10. In addition, it is observed that due to less practicing of agile and cloud computing in industry and difficulty in conducting industry survey, instead of practical experimentation, some studies prefer conceptual solutions for ADCC.

3.2. RQ2: Impact of cloud computing on agile development

The SLR study explores the impact of cloud computing on agile development by proposing a taxonomy as shown in Fig. 11. The taxonomy discusses four major factors that affect software development such as:

1. Impact of cloud quality attributes
2. Impact on collaboration
3. Impact on transparency
4. Impact on development infrastructure

The impact can be positive in the form of benefits of cloud computing for agile development and can also be negative in the form of challenges due to cloud computing.

3.2.1. RQ2-1: What is the positive impact or benefits of cloud computing in agile software development?

Impact of cloud quality attributes. The cloud computing quality attributes and the services have an important role to help, influence and accelerate the agile software development. Quality attributes are further divided into sub-factors as shown in Fig. 11.

Scalability has two perspectives. In agile development perspective, scalability is characteristic of managing and controlling a large team. For example, management of sprint, backlog, and feedback activities. A large number of experienced team leaders and staff are required in order to manage all these activities. Furthermore, an extensive infrastructure in the form of hardware and software resources is needed to manage a large team (cloud perspective).

Cloud computing provides all resources with any infrastructure size without any delay [S11, S30 and S32]. In contrast, some studies reported that scalability or large team support is a challenge and an unanticipated issue [S1 and S22].

Another perspective of scalability is with respect to cloud computing automation that provides a scalable environment [S6, S9, S11 and S23]. Out of nine (9) studies as shown in Table 11, six (6) studies reported a positive impact of cloud computing on scalability (large team support) in agile software development.

Second sub-factor is interoperability, which is defined as provision for a user to work on-premise or on some cloud [S23]. Study [S3] reported interoperability in crowd-sourcing context. Cloud ecosystem is hampered and users are not allowed to change vendor of cloud so as to optimize the resources at the different level of organization [S9 and S23]. Cloud computing has drawbacks in terms of lock-in and interoperability concern [S19]. The study [S23] highlights the need for standardization and interoperability between different service providers of the cloud.

Generally, cloud services provide API and platform. However, the proprietary API makes it difficult for a customer in switching to another cloud provider [S23]. It is observed in primary studies that there is no positive impact of cloud computing regarding interoperability. The effect of various studies is concluded in Table 11.

Table 9
Application developed using ADCC.

Study	Basic Type	Major artefacts used	User	Agile process	Evaluation nature	Results
S10	Web-based, mobile based	Google App Engine, Map API, SQL server, Selenium IDE	government agencies and city leaders	Model-driven	-	Prototype can be build with cloud
S20	Web-based	Groovy and grail, Eucalyptus	Physicist in space weather research	Scrum	rating of artefacts on certain criteria	Eucalyptus is not suitable as cloud
S25	Web-based Model	Ruby on Rail, Mongo DB	applications for computing research	Agile	based on usage rate	usage rate growing shows positive
S27	Web-based	Microsoft Azure	To track the firefighting activities	Agile	-	Study claimed to achieve all objectives

Table 10
Tools for ADCC.

Study	Application type	Major factors / Artefacts under discussion	Agile process	Evaluation	Results / Conclusion
S6	mobile based	Amazon EC2, SQL database	XP	Compare with a study [S15] as a base study	Cloud use is positive
S7	client-server based	project management tool for agile	-	Controlled Experiment	use of tool simple and easy

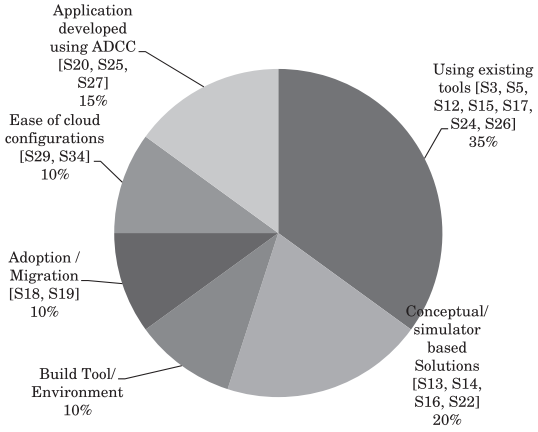

Fig. 10. Contribution of existing studies with respect to solution techniques.

Third sub-factor is maintainability, which has been paid little attention in the primary studies. The cloud computing has no significant effects on portability although it is still a challenge in cloud computing.

Table 11
Impact of cloud on different quality artefacts.

No.	Artefact	Total studies	Significant positive impact	Significant negative impact	No significant impact	Mixed impact
1	Scalability	9	S6, S9, S11, S22, S23, S32	-	S1, S30	S17
2	Maintainability	1	-	-	S13	-
3	Portability	3	S27	-	S23, S28	-
4	Interoperability	4	-	S9, S19, S23	S3	-

There is little discussion of this issue in primary studies.

Cloud computing provides technology services by eliminating capital investment and also reduces running cost i.e. no need for hiring especial hardware engineer for managing servers. The taxonomy subdivides cloud services into PaaS, SaaS, IaaS, Everything as a Service (XaaS) and Test as a service (TaaS).

PaaS provides hardware and software tools/ platform to the developers. In addition, PaaS provides the same environment to all users

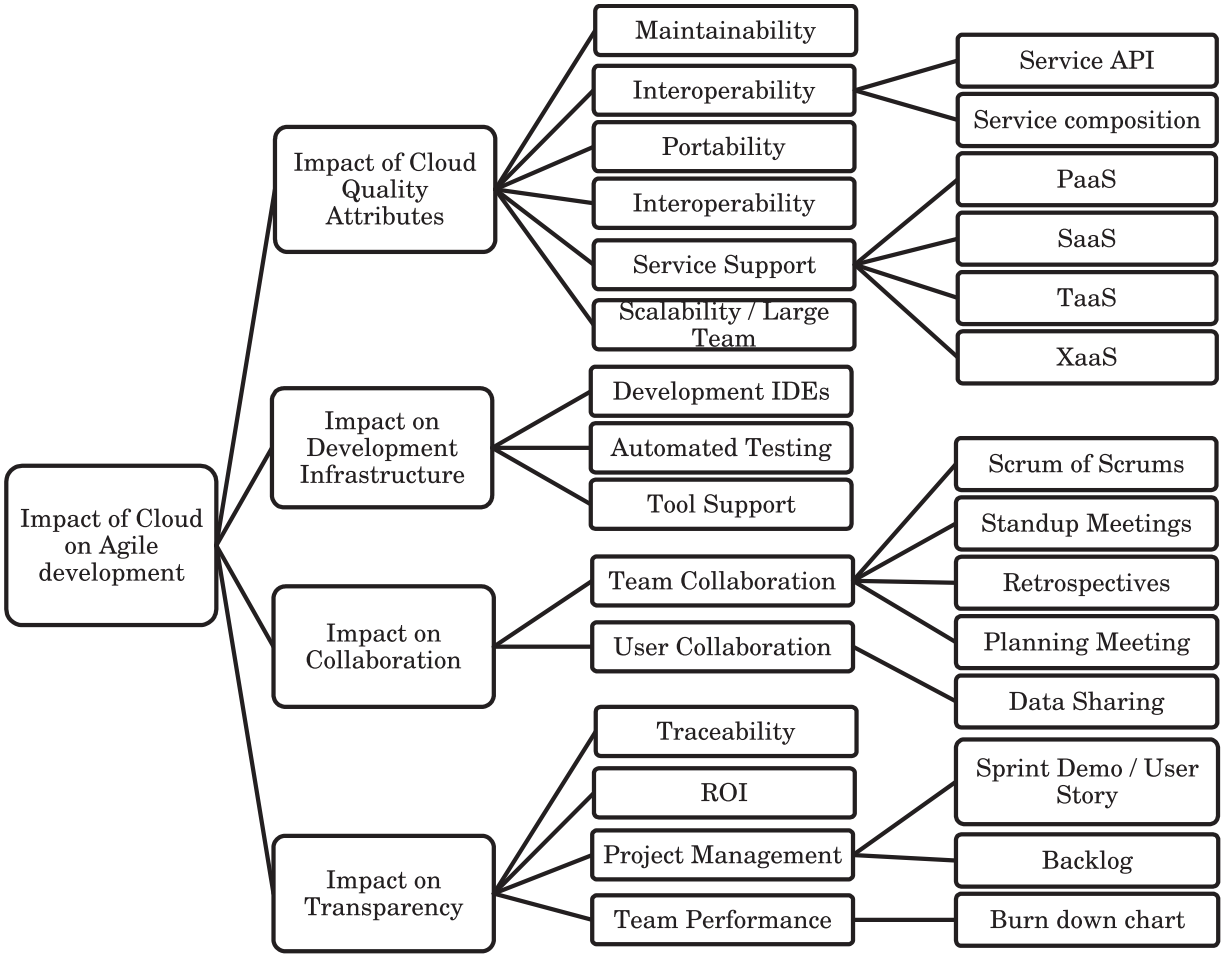

Fig. 11. Impact of cloud on different artefacts in ADCC.

Table 12
Impact of cloud services in agile software development.

No.	Cloud service	Total studies	Significant positive impact	Significant negative impact	No significant impact	Mixed impact
1	PaaS	7	S4, S6, S9, S12, S32	-	S3, S35	-
2	SaaS	4	S5, S27	-	S2, S24	-
3	TaaS	1	S21	-	-	-
4	IaaS	6	S13, S20, S29	S19	S15	S9
5	XaaS	1	-	-	S18	-

locally or globally distributed [S3, S4 and S9] and centrally shared.

The user can view the updating in software without interrupting the development process and can actively participate in user-feedback. Various service providers such as Google App Engine-PaaS service [S35], CloudForge-PaaS platform [S12], Microsoft Azure [S32], Amazon EC2 cloud, IBM CloudOne [S6] are reported in primary studies. The impact of PaaS service on agile development [S4, S6, S9 and S12] is shown in Table 12.

SaaS facilitates easy administration, automatic updates, and patch management [S29]. It is on-demand service and measured on the pay per use principle. SaaS's code management applications allow users to share their code globally [S5 and S27]. For example, **CodeSpaces, GitHub, GoogleCode, SourceForge**. SaaS provides collaboration tools that eliminate the need for a special tool [S2 and S24]. Testing as a Service (TaaS) is a cloud-based platform to deliver automated testing service. It provides test labs including application under test and test tools [S21].

IaaS provides a simple, low-cost and efficient environment to implement and scale application [S29]. This service also allows full control in hosting software, hardware, operating system [S20], server and data storage to the user [S13]. In a study [S15], agile development methods and IaaS composition allow rapid development and testing for the agile team. It is observed that most organizations prefer to use IaaS as compared to SaaS [S9]. Furthermore, IaaS provides dedicated physical server and virtual server instance. However, a study reported that virtual server's instances do not like to share hardware [S19]. Another study [S18] introduces XaaS platform. They refer to XaaS as, a set of services known as product.

The current section synthesizes that the cloud computing service (PaaS, SaaS, and IaaS) has a significant positive impact on scalability. Cloud computing has a less positive impact on maintainability, portability, interoperability, service API and service composition.

Impact on development infrastructure. Organizations have invested much in the provision of development infrastructure especially in the distributed environment, compute intensive and scientific application development. Usually, project starts are delayed due to unavailability of infrastructure i.e. hardware support , software configurations or installation of software and patches. This problem is increased when the development team is scattered logistically at distant positions (distributed environment). On the other hand, due to changing market demand, there is a need for testing environments to test new software. In order to solve this issue, cloud computing provides a highly customizable and optimized-utilization of IT resources.

The resources are in the form of test servers [S11, S12, S21, S31 and S33] and deployment servers [S4]. Due to interconnected servers, the users and development team members can view and analyse the change in development artefacts at various levels. The team can share prototype version with the product owner, user and client of the project instantly. Furthermore, it may help in accepting requirements and maintaining the log for traceability. These resources are easy to access and usable in local as well as the distributed environment. The list of resources and their providers is given in Table 13.

Agile software development has benefits of cloud computing [S22] such as zero initial investment for software development infrastructure, no extra cost for maintenance of hardware and optimum utilization of resources locally and on the cloud, faster time to market, easy set-up, and deployment of applications.

Cloud computing provides development environment such as **chef puppet** [S3, S37], **CloudIDE, Jira, GitHub** [S5], **Eclipse** on the cloud [S12 and S15], Microsoft Azure [S16, S27], Ruby and Rails [S5 and S27]. Cloud computing provides readily available, highly scalable, low cost and multiple machines without building own infrastructure. For automated testing, cloud computing provides environment such as **Amazon EC2** and **Jenkins** Instance [S4, S11, S31 and S33], **CloudForges** [S12] testing servers for local as well as for cloud environment. All these facilities provided by the cloud computing have a positive impact on agile development. Organizations can also avail these resources without computing services such as dedicated servers for testing, but with a high cost. The availability of resources for testing is itself an expensive task.

Impact on collaboration. The agile principles emphasize frequent collaboration and continuous feedback [S2]. In distributed locations, the collaboration among team members produces the effect of unity and reduces the time to resolve the issues. The collaboration can be in the form of sharing performance status of work and identifying the activities to improve the development process. A dedicated build server is required for the retrospective meeting, the cost must be paid [S17], but it is an important principle of agile development [S16 and S32].

Other ways of collaboration in agile software development are sharing of data (such as automated build and test reports, prototypes, ideas, requirement documents), stand up meeting, sprint planning and a scrum of scrums. For sharing of automated build reports, dedicated build server and administrator is required for maintaining networking resources. When the team is sitting in geographically distant locations, it becomes difficult in sharing release build [S17] and related documents at runtime.

Another way of collaboration is through means of communication for the sharing of information such as discussion forums, Wikis [S26], cloud telephony by Amazon web service (AWS) [S17], special tools **confluence** and **bootstrap today** [S3 and S24] and **Skype**[S5, S12 and S15]. Cloud services facilitate cheap and easy integration with other enterprises [S22]. Some other collaboration means and technology used are calendars, email, instant messaging, screen sharing, shared document spaces, social media, telephones, and video conferencing [S2]. Some studies reported that collaboration activities such as on-board meeting, telephony utility, discussion blog and group chat are used for the daily meeting or stand-up meeting [S16, S17, S22, S28, S29 and

Table 13
Development environment and service providers.

No.	Description	Tool and Studies
1	Cloud Environment Providers	Heroku [S2, S4], Engine Yard [S2], Microsoft Azure [S2, S4, S6,S27], Google App Engine [S2, S15], Force.com [S2],Amazon Web Service [S4, S6, S17], IBM [S6]
2	Software development Environment	Chef/Puppet [S3,S23 , S24], CloudIDE [S3] , Eclipse IDE [S15,S24], Maven [S23, S31], Ant [S23, S31], Ivy [S23], Gradle [S23], Rake [S23], Jenkins [S4, S11,S23, S31, S33], Hudson [S23],Travis [S23], Teamcity [S24], Selinium [S24], LoadRunner [S24]

Table 14
Collaboration supporting tools.

No.	Description	Tool and studies
1	Requirement document sharing	GoogleDocs [S2], DropBox [S2], Team Foundation Server (TFS) [S2,S24], Pivotal Tracker [S24]
2	Team Communication	Cockpit [S21], Mingle [S21], Skype [S5,S12,S15], Basecamp [S24], Microsoft share point [S24], Bootstrap today [S3, S24], Confluences [S3,S24]

S32]. The scrum meeting is done by using **Cockpit and Mingle** [S21]. However, a study [S17] considers it a challenge in Distributed Agile software development. The study S22 suggest that retrospective meeting, sprint planning meeting and Scrum of Scrums may cause a delay in software development. The list of communication and collaboration tools is given in Table 14.

The different agile software development activities, their importance, and way of implementation in cloud computing environment shows that the cloud computing supports and helps agile software development in local as well as in the distributed environment. Team collaboration is reported by the (50%) of primary studies [S2-S5, S9, S12, S15-S17, S19-S22, S24, S26, S28-S30,S32].

Impact on transparency. In agile software development, lack of communication and collaboration reduces the visibility and transparency in the project. Agile Scrum methodology aims to keep everything visible to the members. Transparency may be inside and outside the project team. Transparency in agile software development is achieved by performing the activities such as user story [S21, and S29], burn-down chart [S22] and product backlog to predict and plan future task for upcoming sprints [S4]. If a defect is detected in the project, the team fixes the bug and delivers it depending on the priority in backlog list [S32]. All these activities are accomplished with the help of tooling provided by the cloud computing listed in Table 15.

On the other hand, traceability is the property of project which enhances the transparency and visibility about a particular part of the information and their relationship. Cloud computing helps in achieving real traceability of code [S12], i.e., if there occurs a minor change in code, it should be reflected and observed globally throughout the organization. Furthermore, the client is updated with the change in software due to immediate deployment. A study [S30] presents a holistic model for traceability in hybrid cloud computing environment.

Return On Investment (ROI), an essential economic factor for transparency in software development, is neglected in primary studies. The use of various artefacts to achieve transparency advocates that cloud computing has a positive impact on transparency. Cloud computing helps in order to perform all these Agile software development activities.

In contrast, some studies reported that transparency and data sharing is a threat to agile software development [S1, S3-S6, S9, S12, S17, S26 and S30]. The detail of threats is given in Section 3.2.2.

The synthesis of primary studies revealed that cloud computing solved the issue of scalability and reduced cost by providing technological services (in terms of hardware and software infrastructure). Furthermore, communication problems among team members in AGSD are solved due to cloud computing.

The above discussion is regarding the positive impact of cloud computing on agile software development. We try to explain using different perspectives. The negative influence and challenges due to cloud computing are discussed in next section.

3.2.2. RQ2-2: What are the challenges faced in practicing agile development using cloud computing environment?

In this section, the SLR identifies the challenges reported in primary studies.

Overhead of cloud provider. In agile software development, requirement gathering involves customers, users and developers; however due to cloud computing environment, cloud providers are also included in this activity [S13 and S14]. Cloud provider will know the size, architectural details, virtualization strategy and infrastructure usage. Virtualization technique is used to deal with many customers in parallel to fulfill the high demand for customer quality of service agreement required. Cost estimation also depends on the cloud server providers, who estimate the cost of infrastructure to be used.

Communication and coordination among cloud provider and software engineer. It is hard to establish the interaction between a software engineer and cloud provider [S13, S14 and S28]. The amount of interaction depends on the type of cloud. In private cloud, the user has more power of governance and requires less interaction than public cloud. The software engineer feels comfortable with private cloud due to security and liberty, but it is costly.

Security threat. Study [S19] reported that security is a primary concern in cloud computing, and the threat can be due to data and code ownership. Enterprises feel uncomfortable in residing their data on a public cloud [S17]. Furthermore, public cloud causes a security threat for organizations regarding confidential data [S19 and S28], controllability, visibility, performance, and availability [S28]. In addition, project environment security is exposed to the public domain. Cloud takes ownership of code and control. Furthermore, identity and access management is an issue because it is hosted in the public domain. One solution to this problem is to use the private cloud, but it increases the cost of software development.

Lock-In and interoperability concerns. In cloud computing, the big issue is interoperability; there are no universal standards or interface defined for collaboration among different cloud platforms. Every cloud provider has its own services. The selection of cloud service is another challenge. After selecting a cloud vendor, change of provider leads to vendors lock-in [S9, S17, S19 and S24].

Overhead of changed environment. The software developer's community feels uncomfortable in a new platform such as cloud computing environment. The un-comfortability is due to lack of guidance for cloud computing standard architecture [S19]. Study [S6] reported that extra effort is spent by the team to search the open libraries for developing mobile cloud application. Additional effort is needed for modification and integration in existing code.

Table 15
List of transparency supporting tools.

No.	Description	Tool and Studies
1	Agile project management	JIRA [S2, S5], Mingle [S2], Rally [S2], ScrumWorks [S2], Trac [S2], VersionOne [S2], Xplanner [S2], AgileFant [S17], Stats [S23], jmxtrans [S23], Metrics [S23], Esper [S23], Ganglia[S23], Graphite[S23], Cube[S23], CloudSpoke on Topcoder[S3], Trustie[S3], REDMINE[S4]
2	Code management	Code Spaces [S2], GitHub [S2-S6], GoogleCode [S2], SourceForge [S2], Unfuddle [S2], GitLab [S4], BitBucket [S4, S6], Subversion [S23, S24], Git [S23, S24], Mercurial [S23]

Lack of practical experience. In most of the empirical studies, facilitation of cloud computing in agile software development practices is theoretical. There is a lack of exploration of practical aspect and challenges [S17]. The industry experts pointed out non-technical problems, for example, lack of training and leadership. The study [S19] reported that it is difficult to find the enterprise having agile development methodology and cloud environment that can answer questions related to process practices (such as efficiency, productivity, etc.).

Requirement for online connectivity. One challenge of cloud computing is to maintain an environment for different stakeholders of the organization [S17 and S19]. This provision increases the cost of development. In addition, the provision of a platform for development and testing environment increase cost.

Development environment threat. The use of software tools in public domain may present a threat in the form of publicizing code, configuration files or build data. This may lead towards the high risk in the confidentiality of project [S6 and S17].

Identity and access management. Web hosting on AWS cloud is in the public domain, so it is a concern among users of hosted web portals. This will allow the user of one web portal to access some other web portal and hence be prone to the false identification and unauthorized access [S9, S17 and S28].

Compliance to legal standards. The European Union (EU), practices legal standards regarding security and ownership of organizations' data [S17 and S24]. Public cloud set up is not acceptable there. This is a challenge for global companies.

Quality concerns. The study [S17] reported that in cloud computing environment there is enhancement regarding cost and time. However, with respect to quality, on-premise software development produces quality software as compared to cloud environment due to security concerns. Communication and security threats are described by the majority of the reviewed studies as threats in this area.

3.3. RQ3: Research trends and current gaps

The SLR study uses a bubble plot with four (4) facets that provide a quick overview of studies as shown in Fig. 12. The research facet relates to the research classification, the evaluation facet refers to an evaluation method, and the contribution facet relates to i) cloud features and ii) agile features. The first quadrant represents the research classification by cloud features. Taxonomy is proposed for classification of studies, and all cloud features are discussed in detail in Section 3.2.

The second quadrant represents evaluation methods by cloud features. The third quadrant represents evaluation method by agile features. The fourth quadrant represents research classification by agile features. The result of SLR is depicted in the bubble plot as shown in Fig. 12.

The size of a bubble represents the frequency of studies for the artefacts corresponding to coordinates axes. The major categories / artefact described in proposed solutions are collaboration (8), scalability (7), and transparency (6) studies.

The most reported research contribution type is proposed solution. The least reported artefacts are portability and maintainability, and research type is validation research.

With respect to research type contribution, the significant proportion of the research is solution proposal (73%). The study distributes the solution proposal into different solution techniques. Out of twenty-one (27) studies, solution proposal using existing tools [S3, S5, S12, S15, S17, S26 and S35] in seven (7) primary studies (35%), solution using system dynamics modelling [S13, S14, S16 and S22] in four (4) studies (20%), solution described by agile and cloud adoption/migration [S18 and S19] in two (2) studies (10%). Rest of the solutions are focused on other techniques and primary studies as shown in Fig. 12.

The proposed solutions with respect to contribution types [63] are methodologies [S5 and S34] used in two (2) studies (5%), framework [S3, S6, S9, S19, S21, S24, S31, S32 and S36] used in nine (9) studies (24%), Application developed using ADCC [S10, S20 and S27] are in three (3) studies (8%), Tools to help ADCC [S6,S7, S16 and S25] are in four (4) studies (8%) and model proposed in primary studies [S8, S11. S13, S14, S17, S22, S29, S30 and S33] are ten (10) (27%) out of the total (37) primary studies.

The second research type contribution is case study conducted in [S2, S12, S15, S18 and S23] in five (5) studies (14%), survey paper (3%) and other (11%).

The contribution with respect to agile emphasis features are Team collaboration (owner and user included in the team) is focused in 19 primary studies (50%), Transparency is discussed by (12%) studies, and discussion about the provision of development environment on the cloud is found in 11 (29%) studies. Only 2 (5%) studies share the holistic view of traceability in different artefacts of software development phases.

The contribution with respect to cloud emphasis features is relatively small in number because the majority of the studies discuss the benefits and impact of cloud on agile development. The contribution of studies [S1, S3, S5, S6, S12, S15, S16, S18, S21-S24, S26, S30 and S32] focused on scalability / large team support is 15 (39%), cloud service support (PaaS, SaaS etc.) [S1, S3, S4, S6, S12, S15, S17, S18, S26, S27, S30 and S32] is discussed by 12 (31%) primary studies. Less attention is given to portability (5%), interoperability (13%) and maintainability (3%).

3.4. Discussion

Based on the review of the literature, cloud computing does not alter the agile software development process. However, the cloud computing helps agile software development process by reducing cost and increasing simplicity. The findings of the research questions are as follows:

(RQ1): What is the-state-of-art research in agile development using cloud computing environment? In total 37 studies are selected out of 647 studies searched by different database between 2010 and 2017. All studies included in SLR cover the area of ADCC. The RQ classifies the primary studies into research type and contribution type. The SLR also classifies and explains the different solution techniques reported in primary studies. In addition, the frequency of reported artefacts is listed in this RQ.

The findings of the review with respect to the demographic point of view are, solution proposal (55%), case studies 13%, surveys (2%) and none of the studies conducted validation research. The solution proposal of existing primary studies is characterized into six (6) categories. Out of 21 primary studies, Solution using existing tools (35%), Conceptual/ simulator based Solutions (20%), build tool/environment for ADCC (10%), solution using Agile-Cloud adoption / migration (10%), solution for ease of cloud configurations (10%) and solution presented by developing application using ADCC (15%).

With respect to artefacts, the majority of the studies reported team collaboration feature is (50%) and Development IDE (29%). Furthermore, primary studies that developed a mobile application for experimentation in the agile-cloud environment are (11%).

On the other hand, it is observed that few studies reported on testing. There is a need for more studies performing unit testing, integration testing and regression testing in ADCC. More discussion can be done on test as a service, especially for ADCC environment. Another topic that requires researcher's attention is traceability. There is only one study that discusses the holistic traceability. There needs to be an investigation on requirement-requirement traceability, requirement to

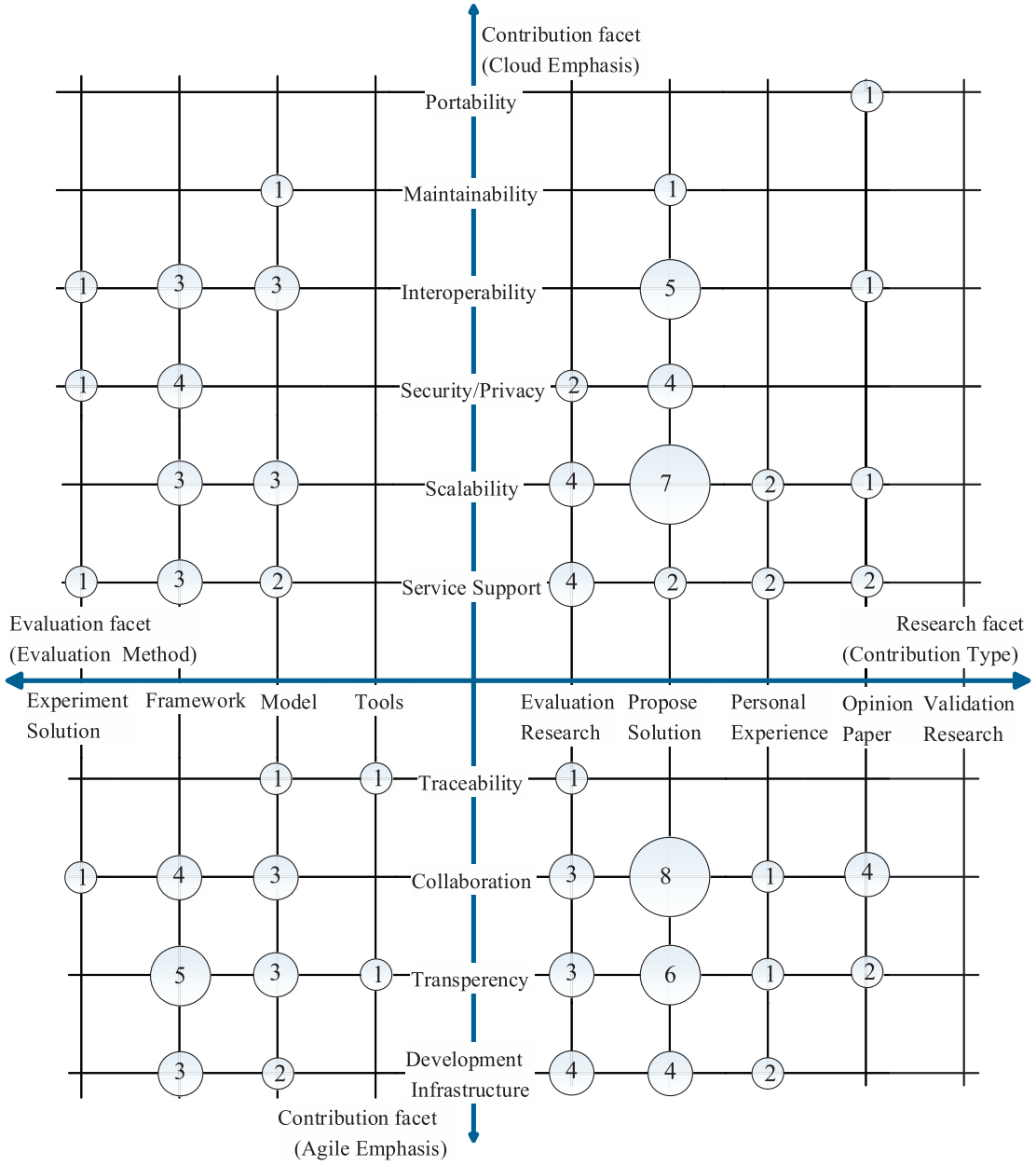

Fig. 12. Research contribution in practicing ADCC.

test, requirement to design and the more specialized area of traceability. Furthermore, the majority of studies discussed the Scrum, i.e., agile method.

(RQ2): What are the impacts of cloud computing on agile software development? The findings of the SLR study presents a simple but useful taxonomy which elaborates how cloud computing helps agile software development. The SLR study can be classified into four major types: impact of cloud quality attributes on agile software development, the impact of cloud on development infrastructure, the impact of cloud on collaboration and impact of cloud on transparency. The SLR study observed that scalability and service support (PaaS, SaaS etc.) are reported in the majority of primary studies. Cloud computing resolves the issue of scalability in a simple way with minimum effort and cost. Provision of infrastructure for development (hardware and software) and communication and control of development team, a big problem in global agile software development, is resolved by cloud computing. In addition, cloud computing helps to increase transparency in software development activities through smooth execution of user story and backlog management, burnt down chart and traceability. It is observed that fewer studies report on traceability and automated testing.

The challenges identified in primary studies while developing software in cloud computing environments are security threats, lock-in, and interoperability concern, the overhead of changed environment, lack of practical experience, the requirement for on-line connectivity, development environment safety, identity and access management, compliance to legal standards and quality concerns.

(RQ3): What are the research trends and gaps in agile development using cloud computing environment? The maturity of studies and type of research contribution is as solutions proposal (50%), evaluation research (17%). There is the low frequency of validation and philosophical research. There needs to be open access to results for replications of experiments which will lead to generalizing conclusions for different areas of ADCC.

The result of SLR study shows that cloud computing is a natural partner of agile software development. The desires of agile software development and services of cloud computing bind them. Desires mean

the quality attributes of agile software development. All 12 principles of agile are helped by cloud computing. There are several implications of the SLR study on practitioners and researchers. The merger of Agile and cloud computing has benefits and drawbacks, however, due to the inclusion of cloud computing, all phases of SDLC are affected. This merger opens several channels for researchers. For example, due to cloud inclusion, the cost and effort estimation of software changed. This will lead to new studies with new dimensions. Similarly, requirement elicitation in ADCC, traceability, security due to the cloud, communication, testing, integration and many more directions are waiting for research. The SLR has following implications:

1. The researcher will obtain awareness of cloud computing in the agile software development.
2. The solutions (environment, project management, testing, integrations and other development activities) so far developed for ADCC. These solutions will help practitioners in adopting ADCC.
3. Use of tools increases the transparency in software development activities.
4. Positive and negative impact of cloud computing on agile software development will help practitioners and industries in the adoption of ADCC.
5. In ADCC, tool support has an important role.
6. Small enterprises have less interest in ADCC due to lack of awareness.
7. Cloud service providers have an important role in ADCC.
8. There is lack of practitioners involved in ADCC.
9. The SLR helps the practitioners in the selection of development environment, testing environment and tooling required to establish ADCC.

4. Threats to validity

The SLR presents the taxonomy of solution for ADCC and the different factor affecting the ADCC. Although the result of SLR study is reliable, there is a chance of bias and validity threats during design, conduct, analysis or conclusion phase of SLR study [8]. Search strategy identifies maximum possible studies available in the literature by avoiding bias. It is possible to miss some studies due to grey literature such as technical reports and Ph.D. theses.

The study relates to different communities such as software engineering, in particular agile development, information system, and networks. The process of identifying primary studies is done by considering bias, internal validity and external validity suggested by both CRD Guidelines [35] and the Cochrane Reviewers Handbook [59] and also recommended by the study [8].

In addition, the validation of the study is based on the adoption of a procedure to conduct the review. The clarity of analysis will enable the researchers to judge the trustworthiness of results objectively.

5. Challenges and future work

The communication and collaboration issues in agile global software development (AGSD) are resolved to some extent by using cloud computing services and social networks. Another problem in AGSD is the provision of resources such as development, testing, and deployment environment and the difficulties in controlling development activities. Cloud computing also resolves these issues. Scalability issue is resolved with the help of cloud computing. Interoperability and portability are the issues of cloud computing also introduced here in ADCC. Security threats of cloud computing are introduced in ADCC. Based on SLR study, we suggest that research be conducted in the following areas:

5.1. Need of experiment-based studies

It is observed that there are some research trends which have not been addressed thus far. The majority of the article's research contributes proposed solutions (55%), but there is a lack of evaluation research (15%). Furthermore, in solution proposal, there is a lack of evaluation and benchmark experimentation results. More industrial case studies are required for exploring the real-time systems in ADCC.

5.2. Traceability of development artifacts

Traceability is focused on two studies only; more attention needed in terms of requirement-requirement traceability, the requirement to test, the requirement to design and the more specialized area of traceability. Furthermore, there can be more studies to highlight the role ADCC in requirement elicitation, especially in a distributed environment. Traceability of miss communication and lack of communication between team members can be traced by some alarm mechanism.

5.3. Explore of testing artifacts

Most studies discuss automated testing as a benefit of cloud computing; some studies [S21 and S33] suggest different cloud testing services for testing. However, there is a lack of detailed studies for agile testing in ADCC. A study [S21] discusses Test as a Service (TaaS). However, more studies are required to do more experimentation with a different type of testing in ADCC. Research should be done on performing unit, integration, performance and regression testing in ADCC and about the control of various testing activities.

5.4. Lack of practitioners interest

The practitioner interest lies in the real-time experimental solutions and empirical studies of the area. The validation is required for existing solutions proposed so far to generalize conclusions for different areas of ADCC. The research needs to be focused on industry projects or professional projects.

5.5. Inclusion of more agile methods

Only a few agile software development methodologies (Lean-Agile, Pair Programming, DSDM and Extreme Programming) have been reported in primary studies. There are many more agile software development methodologies (FDD, Kanban, TDD and Crystal, etc.) that have not been the focus of research so far [S12 and S18].

5.6. Lack of validation research

The result of SRL review shows that validation research is needed in all types of contribution as shown in Fig. 12. Evaluation requires through real-world case studies and experienced reports in this area, which will increase the confidence of researcher and practitioner [26,67].

5.7. Overhead of new environment

The majority of studies use different tools to propose a solution for ADCC. However, these tools have the extra overhead of configurations on different platforms. There is a requirement for a single platform for development teams to work [S22]. Furthermore, there should be a solution that incorporates industry level agile software project management in cloud computing environment [S16].

5.8. Agile requirement

The bottleneck in agile software development is the maturity of the requirement that is implicitly improved with the involvement of client [25]. More empirical studies are required to focus on this problem in ADCC. Non-functional requirement (NFR) in agile software development is neglected due to the nature of agile software process [28] and also neglected by the client [21] and industry practitioners [13]. There is a need for more empirical research in this area.

5.9. Need of studies based on cost estimation

A framework for cost estimation of software by keeping cloud computing cost in view is also required.

5.10. Cloud computing and parallel activities

Cloud computing provides resources to work in the parallel environment. in the, authors' opinion, there should be some research to enable agile software development with parallel teams and job.

In a future aspect, there should be more case studies and industrial survey of firms and enterprises implementing ADCC. There are a number of solution proposals in this area. These solutions should be tested through validation research papers. More research is required for exploring the traceability mechanism in ADCC. Furthermore, an alarm system is needed which detects communication weakness and abnormalities among team members.

6. Conclusion

The systematic literature review has a number of implications for research. It indicates that there is a need for more empirical studies focusing on practical perspective in the area of agile software development and cloud computing. SLR classifies the primary studies into research contribution, solution types, agile-cloud features and artefacts. The study ranks the solution proposal into using existing tools, conceptual solutions, cloud adoption/migration and ease in the configuration of the cloud. The insight of the studies shows that cloud computing has a positive impact on agile software development. Infrastructure support, virtualization, automated testing and deployment, communication platforms facilitate agile software development. Cloud computing reduces delivery time to market of software. Cloud computing increases transparency through smooth execution of user story, backlog management and traceability. The challenges introduced due to cloud computing are security threats, lock-in and interoperability concerns, lack of practical experience and development environment safety. Furthermore, compliance with the standard and quality concerns is reported in primary studies. Most of the studies present their solutions by using existing tools and cloud services, inferring that the growth is relatively at the end of cloud services and solution providers and fewer researchers are involved in this area.

Acknowledgement

We are thankful to Ministry of Science, Technology and Innovation (MOSTI) to support this research under eScience grant vote: 4S113. We are also thankful to Universiti Teknologi Malaysia (UTM) for providing us with the research facilities and UTM Research University grant vote: 13H63.

References

[1] A. Abello, J. Ferrarons, O. Romero, Building Cubes with Mapreduce, Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP, ACM, 2011, pp. 17–24.
[2] S. Abhishek, M. Frank, A roadmap for software engineering for the cloud: Results of a systematic review, IGI Global, Hershey, PA, USA, 2014. 1–16.
[3] G. Raj, K. Yadav, J. Arunima, Emphasis on Testing Assimilation Using Cloud Computing for Improvised Agile SCRUM Framework, Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE),2015 International Conference on, IEEE, 2015, pp. 219–225.
[4] B.K. Allen, G.W. Romney, P.P. Dey, M.D. Romney, Collaborative academic-government agile development of a cloud prototype fire retardant drop log application for wildfire management, J. Res. Innov. Teach., Public. Nation. Uni. 8 (1) (2015) 99–115.
[5] O. Akerele, M. Ramachandran, M. Dixon, System Dynamics Modeling of Agile Continuous Delivery Process, Agile Conference (AGILE), 2013, IEEE, 2013, pp. 60–63.
[6] F. Almudarra, B. Qureshi, Issues in adopting agile development principles for mobile cloud computing applications, Procedia Comput Sci 52 (2015) 1133–1140. 2015.
[7] H. Benfenatki, C.F.D. Silva, A.-N. Benharkat, P. Ghodous, Cloud-based Business Applications Development Methodology, IEEE 23rd International WETICE Conference (WETICE), IEEE, 2014, pp. 275–280.
[8] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J.Syst.Softw. 80 (4) (2007) 571–583. 2007.
[9] N. Caithness, M. Thurston, A simple drop and compute model for a saas cloud infrastructure for advanced research computing, Oxford e-Research Centre/2013, 2013.
[10] L. Cocco, K. Mannaro, G. Concas, A Model for Global Software Development with Cloud Platforms, 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2012, pp. 446–452.
[11] B. Cohen, 2013. Paas: new opportunities for cloud application development, Computer/46 9 (2013) 97–100.
[12] G. Concas, K. Mannaro, L. Cocco, Managing a Global Software Project under an Agile and Cloud Perspective, Proceedings of the International Conference on Software Engineering Research and Practice (SERP), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2013. 1.
[13] D. Domah, F.J. Mitropoulos, The NERV Methodology: A Lightweight Process for Addressing Non-functional Requirements in Agile Software Development, SoutheastCon, IEEE, 2015, pp. 1–7.
[14] A. Dumbre, S.P. Senthil, S.S. Ghag, Practicing Agile Software Development on the Windows Azure Platform, Infosys Whitepaper/(2011), (2011). https://pdfs.semanticscholar.org/e45e/1c9969ae4c591e4311eee204f40e184b40bf.pdf .
[15] U. Durrani, J. Richardson, J. Lenarcic, Z. Pita, Lean traceability solution through SLAM model: a case study of a hybrid delivery team in a hybrid cloud computing environment. 22nd australasian software engineering conference: ASWEC, Eng. Aus. 19 (2013).
[16] D. Farroha, B. Farroha, An Agile Enabler Framework: Architecting Services in the Clouds, MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM, IEEE, 2012, pp. 1–6. 2012.
[17] S. Franken, S. Kolvenbach, W. Prinz, I. Alvertis, S. Koussouris, Cloudteams: bridging the gap between developers and customers during software development processes, Procedia Comput Sci 68 (2015) 188–195. 2015.
[18] M.M.A. Ghosh, W. AlSarraj, A case study on academic services application using agile methodology for mobile cloud computing, Intern. J. Res. Eng. Sci. (IJRES) 2016 (2016).
[19] B.P. Gopularam, C.B. Yogeesha, Mechanism for on Demand Tag-based Software Testing in Virtualized Environments, Advanced Computing (ICoAC), 2012 Fourth International Conference on, IEEE, 2012, pp. 1–5.
[20] B.P. Gopularam, C.B. Yogeesha, P. Periasamy, Highly Scalable Model for Tests Execution in Cloud Environments, Advanced Computing and Communications (ADCOM), 2012 18th Annual International Conference on, IEEE, 2012, pp. 54–58.
[21] D.D. Gregorio, How the Business Analyst Supports and Encourages Collaboration on Agile Projects, IEEE International Systems Conference (SysCon), IEEE, 2012, pp. 1–4.
[22] R. Guha, D. Al-Dabass, Impact of Web 2.0 and Cloud Computing Platform on Software Engineering, International Symposium on Electronic System Design (ISED), (2010), pp. 213–218.
[23] T. Haig-Smith, M. Tanner, Cloud computing as an enabler of agile global software development, Issues Inf. Sci. Inf. Technol/ 13 (2016) 121–144. 2016.
[24] S.A.M. Hasaba, A. Faraahib, An Overview of Applying Cloud Computing in Addressing Agile Global Software Development Challenges, Reef Resources Assessment and Management Technical Paper (RRAMT)/, (2014). 2014.
[25] J. Highsmith, 2001. history: The agile manifesto, 2001, Sitio: http://www.agilemanifesto.org/history.html.
[26] V. Ivanov, A. Rogers, G. Succi, J. Yi, V. Zorin, What Do Software Engineers Care about? Gaps between Research and Practice, Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ACM, 2017, pp. 890–895.
[27] I.L. Bagiwa, I. Ghani, M. Younas, M. Bello, Systematic literature review on cloud adoptionn, Intern. J. Internet Broadcast Comm. 8 (2) (2016) 1–22.
[28] S. Jeon, M. Han, E. Lee, K. Lee, Quality Attribute Driven Agile Development, Software Engineering Research, Management and Applications (SERA), 2011 9th International Conference on, IEEE, 2011, pp. 203–210.
[29] V.E. Jyothi, K.N. Rao, Effective implementation of agile practices–in collaboration with cloud computing, Intern. J. Current Eng.Technol./4 3 (2014). 2014.
[30] S. Kalem, D. Donko, D. Boskovic, Agile Methods for Cloud Computing, 36th International Convention on nformation & Communication Technology Electronics & Microelectronics (MIPRO), (2013), pp. 1079–1083.
[31] S. Karouw, H. Wowor, Using Cloud Computing for Building DAS Tondano

Mitigation Disaster Information System Prototype, Informatics and Computing (ICIC), International Conference on, IEEE, 2016, pp. 257–261.
[32] S. Karunakaran, Impact of cloud adoption on agile software development, Springer, 2013. 213–234.
[33] K. Keefe, M. Dick, Using Extreme Programming in a Capstone Project, Proceedings of the Sixth Australasian Conference on Computing Education, Australian Computer Society, Inc., 2004, pp. 151–160. Vol. 30.
[34] S. Keele, Guidelines for performing systematic literature reviews in software engineering, EBSE, 2007 Technical report ver. 2.3. ebse technical report.
[35] K.S. Khan, G.T. Riet, J. Glanville, A.J. Sowden, J. Kleijnen, Undertaking Systematic Reviews of Research on Effectiveness: CRD'S Guidance for Carrying Out or Commissioning Reviews, Number 4 (2n. NHS Centre for Reviews and Dissemination, (2001). and others.
[36] S.H.H. Madni, M.S.A. Latiff, Y. Coulibaly, Resource scheduling for infrastructure as a service (iaas) in cloud computing: challenges and opportunities, J. Netw. Comp. Appl./ 68 (2016) 173–200. and others. 2016.
[37] Manifesto, 2001. manifesto for agile software development, 2001, http://agilemanifesto.org.
[38] M. Manuja, Manisha, Moving Agile Based Projects on Cloud, IEEE International Advance Computing Conference (IACC), (2014), pp. 1392–1397.
[39] P. Mell, T. Grance, The NIST definition of cloud computing, 2011, and others. 2011.
[40] M. Miglierina, Application Deployment and Management in the Cloud, Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on, IEEE, 2014, pp. 422–428.
[41] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew, P. Shekelle, L.A. Stewart, Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-p) 2015 statement, Systematic reviews/4 1 (2015) 1. 2015.
[42] G. Mwansa, E. Mnkandla, Migrating Agile Development into the Cloud Computing Environment, IEEE 7th International Conference on Cloud Computing (CLOUD), (2014), pp. 818–825.
[43] A. Nazir, A. Raana, M.F. Khan, Cloud computing ensembles agile development methodologies for successful project development, Intern. J. Mod. Edu.Comp. Sci. (IJMECS)/5 11 (2013) 28. 2013.
[44] N. Nikitina, M. Kajko-Mattsson, Developer-driven Big-bang Process Transition from Scrum to Kanban, Proceedings of the International Conference on Software and Systems Process, ACM, 2011, pp. 159–168.
[45] M. Paasivaara, B. Behm, C. Lassenius, M. Hallikainen, Towards Rapid Releases in Large-scale Xaas Development at Ericsson: A Case Study, IEEE 9th International Conference on Global Software Engineering (ICGSE), IEEE, 2014, pp. 16–25.
[46] S. Patidar, D. Rane, P. Jain, Challenges of Software Development on Cloud Platform, in: World Congress on Information and Communication Technologies (WICT), IEEE, pp. 1009–1013. 2011.
[47] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, EASE 868–77. 2008.
[48] r. Shriver, 2012. AGILE CLOUD DEVELOPMENT: THE FUTURE OF SOFTWARE, 2012, http://www.virtualizationpractice.com/agile-cloud-development-the-future-of-software-16226/.
[49] M. Petticrew, H. Roberts, Systematic reviews in the social sciences: A practical guide, John Wiley & Sons, 2008.
[50] B. Portelli, 2010. the beauty of agile in the cloud, 2010, https://www.agileconnection.com/article/beauty-agile-cloud.
[51] M.R.J. Qureshi, I. Sayid, Scheme of global scrum management software, Intern. J. Inform. Eng. Elect. Bus.(IJIEEB)/ (2015). 2015.
[52] P. Raj, V. Venkatesh, R. Amirtharajan, Envisioning the cloud-Induced transformations in the software engineering discipline, Springer, 2013. 25–53.
[53] N. Rathod, A. Surve, Test Orchestration a Framework for Continuous Integration and Continuous Deployment, Pervasive Computing (ICPC), 2015 International Conference on, IEEE, 2015, pp. 1–5.
[54] S.R. Palmer, M. Felsing, A practical guide to feature-driven development, Pearson Education, 2001.
[55] J.M. Rivero, S. Heil, J. Grigera, M. Gaedke, G. Rossi, MockAPI: An agile approach supporting API-first web application development, Springer, 2013. 7–21.
[56] S. Singh, I. Chana, Introducing agility in cloud based software development through ASD, Intern. J. u-and e-Service, Sci. Technol./6 5 (2013) 191–202. 2013.
[57] E. Toews, B. Satchwill, R. Rankin, J. Shillington, T. King, An Internationally Distributed Cloud for Science: The Cloud-enabled Space Weather Platform, Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing, ACM, 2011, pp. 1–7.
[58] A. Tuli, N. Hasteer, M. Sharma, A. Bansal, Empirical investigation of agile software development: cloud perspective, SIGSOFT Softw. Eng. Notes/39 4 (2014) 1–6. 2014.
[59] M.V. Tulder, A. Furlan, C. Bombardier, L. Bouter, Updated method guidelines for systematic reviews in the cochrane collaboration back review group, Spine/28 12 (2003) 1290–1299. Editorial Board of the Cochrane Collaboration Back Review Group, and others. 2003.
[60] VersionOne, 2017. 11th state of agile report, 2017, https://explore.versionone.com/state-of-agile/insights-from-the-11th-state-of-agile-report-2.
[61] W. Wang,.R.A.S.D.i. t. Cloud, 2011, https://www.open.collab.net.
[62] T. Wei-Tek, W. Wenjun, M.N. Huhns, Cloud-based software crowdsourcing, Internet Computing, IEEE/18 3 (2014) 78–83. 2014.
[63] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements Engineering/11 1 (2006) 102–107. 2006.
[64] M. Younas, I. Ghani, D.N.A. Jawawi, M.M. Khan, A framework for agile development in cloud computing environment, J. Internet Comp. Serv.(JICS)/5 (2016) 67–74. 2016.
[65] C. Yung, Y.-T. Lin, Implementing TOAST, a tool for agile software project management in cloud computing environments, J. Softw. (JSW)/ (2015) 1310–1318. 2015.
[66] I.I.S.S.S. Zarinah, M. Kasirun, AGILE-BASED SOFTWARE PRODUCT DEVELOPMENT USING CLOUD COMPUTING SERVICES: FINDINGS FROM A CASE STUDY, Sci. Intern.J. (Lahore)/ (2013) 1045–1052. 2013.
[67] M.V. Zelkowitz, D. Wallace, Experimental validation in software engineering, Inf. Softw. Technol. 39 (11) (1997) 735–743.