A Systematic Literature Review to Uncover SaaS Adoption Issues by SMEs – Reasons and Solutions to the Adoption Problem

Olugbenga A Adenuga[1], Ray M. Kekwaletswe[2], Olukorede T. Adenuga[3]
UNISA[1], University of Witwaterand[2], Tshwane University of Technology[3]
South Africa

Abstract

Software as a Service (SaaS) is a cloud technology model rendered by service providers to deliver software usage to organisations via the Internet. The model created a vivacity that fascinates small medium enterprises (SMEs to acquire software at a reasonably reduced price. In this study, we explore how SMEs, adopt SaaS by conducting a systematic literature review (SLR). The SLR revealed the problems SMEs encountered during adoption of SaaS. The search on the literature for the SLR was carried out on six academic databases. The search retrieved 682 articles; 42 articles were selected for the review on adoption of SaaS. The selected articles contextualised the cases; extracting the problems, solutions, and causes of the problems during adoption of SaaS. A total of 58 issues, 32 causal associations as well as 26 treatments were reported. SaaS security problem; support problem was reported most frequently, while probably the most crucial reported issues were related to SaaS security and operation problem. Causally, SaaS operation problem and SaaS support are mostly connected to other themes, while SaaS fault tolerance problem had the least reported issues. The discovered issues, solutions, causes of problems assists to resolve the problems SMEs encountered during adoption of SaaS.

1. Introduction

A systematic literature review (SLR) in management information systems on major academic databases was conducted in this study. The SLR explore previous research that are relevant to the present study and identified the problems that SMEs faces while adopting SaaS. In addition, the reasons why SMEs are discouraged from adopting SaaS were revealed. A critical look at the solutions to the problems were discussed, prompting the rationale for the study [1, 2, 3]. The emphasis on the *adoption of SaaS* is on the business continuity of SMEs, which is critical to most developing country economy [4, 5, 6]. The ubiquitous nature of Information Technology (IT) and the frequent changes in technology continue to revolutionalise the terrain for SMEs business continuity. SaaS as an IT delivery model exist in the cloud tends to be what SMEs need in order for their continued existence [1]. SaaS are attractive IT models for SMEs tend to reduce costs and at the same time increase the flexibility of SMEs IT environment [7].

On the surface, SMEs are highly attractive segment for SaaS, prompting adoption of IT to support their business continuity [2]. The resources for SaaS become need-based and on-demand [7]. Adopting SaaS enables SMEs to harness the benefits such as information security, storage, zero % downtime, resource optimisation, adoption of latest software applications, anywhere and anytime [8,9,10]. SaaS model is a new paradigm in information systems (IS) which allows enterprise application software and operating system (OS) platform delivery over the internet. This enables SMEs to use the software at any time and at any place. Authors [8] stresses that this innovative service differs from traditional IS in that it does not require enterprises to invest in their own IS resources and infrastructures. Authors [11] points out the benefit these varying service provisions present to SMEs by stressing on the services available in SaaS.

The on-demand usage of enterprise software services from SaaS rapidly evolves towards a viable outsourcing option for SMEs [12]. The successful adoption of SaaS is dependent on the ability of the SMEs to assess the numerous services offered that best suits SMEs need. Adopting SaaS help SMEs to concentrate efforts and resources on their line of business. Therefore, this study investigates the problems that hinders SMEs from adopting SaaS and identify the reason and the solution that has been proposed in prior studies.

This paper is structured as follows; firstly, the background to the study, secondly, the method used for literature search, and the result of the literature search. The authors concluded with the discussion on the limitation in the study.

2. Method use for literature search

The objective in this study is to investigate what has been published in major academic databases in management information systems concerning SMEs adoption of SaaS. The need to understand SMEs adoption of SaaS problems and the causes of the problems. This study, therefore, raised the following research questions to assists in achieving the objective.

RQ1: What SaaS adoption issue has been discussed in prior studies?

RQ2: What are the reasons for the SaaS adoption issues?

RQ3: What is the proposed solution for the SaaS adoption issues?

The investigation is limited to SMEs adoption of SaaS, carried out by using the search terms for SaaS adoption and implementation.

The study employed SLR to ensure that adequate related studies are retrieved as specified in [11]. The study further describes the technique to gather all empirical evidence, critically assess it, and to arrive at the conclusion that summarizes the research. The SLR procedure began with collection of similar and all available studies on SaaS, identified situation in the study analysis, present discussion and result section of this paper. The exploration procedure of the literature retrieved comprises of four steps: (i) search strategy, (ii) filtering approach, (iii) information removal as well as (iv) synthesis. The subsection that follows discussion on the strategy used in searching through the academic databases.

2.1. Strategy used for searching

Extensive search of literature was carried out on academic databases over the two periods presented in Table 1. The first part of the literature survey was carried out between 3 March 2018 and 18 March 2018. The second part of the literature survey was conducted between 1 November 2018 and 22 November 2018. The academic databases searched include open access to thesis and dissertation (oatd), Scopus, Association of Computing Machine (ACM), Association of Information Systems electronic Library (AiSeL), Sage, and IEEE xplore. The search term for the retrieval of the literatures are “Software as a Service OR SaaS”. The retrieved literatures were filtered using publication date between 2013 to 2018 (01/01/2013 – 12/31/2018) and all publication inclusive.

Despite all these benefits, the outcome of the SLR conducted indicated that there still exists limited study in the area of SMEs adoption of SaaS as shown in Figure 1 with the conditions for the inclusion and exclusion of articles.

Scanning through the title of the retrieved articles that are published in English Language. Seventy-six articles passed the exclusion conditions, out of the seventy-six articles, forty-two were included in the final SLR review for the study.

2.2. Coding of data

Coding was done using word cloud in Atlas.ti application version 8. This helps to extract the information through grounded theory [12]. The following three steps were implemented in the coding: (i) conceptual coding, (ii) selective coding and (iii) axial coding.

Table 1. Articles retrieved between March and November 2018

Academic Database	**March 2018**	**November 2018**
OATD	15	10
Scopus	201	31
ACM	82	9
AiSeL	143	26
Sage	25	12
IEEExplore	98	30
Total	**564**	**118**

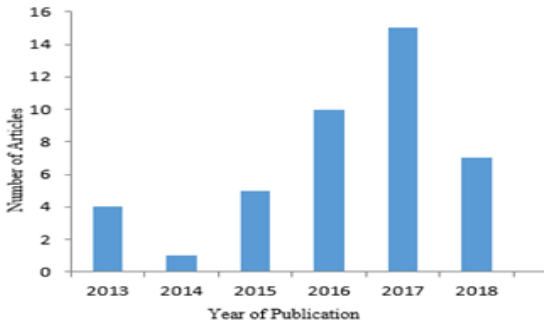

Figure 1. Number of articles published per year

Conceptual coding articles help with the analysis of issues that have emerged when SMEs adoption of SaaS was done. Identifying instances of problems is extremely interpretive work and just including problems, which are named explicitly. For instance, difficulties were not considered inclusive enough. The following quote was coded with the codes "placement problem", even in case it was not explicitly described to become a problem:

“*As the workloads of uses of SaaS applications; materials capacities continue to change over time, as well as the dynamic, therefore the answer found for the original SaaS placement might have been to reconfigure in order to retain the SaaS performance and then to maximise the resource used.*”

--- Case D5

All the problems were examined if any causes or solutions for that issue had been discussed in that case, if so, the study coded the concepts as causes and solutions. The following quote was coded with the code’s “reason”, for “placement problem”. This can be translated into the sentence “reason for placement problem”.

“*SaaS service providers face optimisation problem as they need to satisfy customers (SMEs), to satisfy the previously established as well as proved demands for dynamic optimisation problems that SMEs garbled with in SaaS adoption.*”

---Case D13

Likewise, the subsequent quote was coded with the codes "solution to SaaS adoption problem". This may be checked out as "delivery model that resolves the organisation problem".

"*SaaS delivery model for SMEs, assists in business operation, will have an additional advantage of using the perfect process supplied by SaaS. This kind of benefits may boost the intention of SaaS procurement.*"
---Case D18

During axial coding, this particular analysis made connections between the codes created during conceptual coding. This particular analysis connected each answer code to every issue code, which it had been mentioned to resolve. Similarly, this particular analysis connected each issue code to every issue code, which it had been mentioned causing. This study did not separate problem and leads to codes, since causes might usually be viewed as problems also. On the contrary, the study divided the codes purely being sometimes problems, solutions or reasons, even when several remedies had been thought tricky in the posts. For instance, the answer "multi-tenancy" can be hard in case the "performance, security as well as availability" issue is existing. But to code this particular, we utilized the issue code "multi-tenancy" as related to "performance, security and availability".

Selective coding was carried out to form codes that were applied in the articles. This includes situations, which described the issue code but did not think it over as being a faced issue, they were definitely coded to ground the coded much better and look for variance in the issue idea. Additionally, an issue concept was combined to increase the abstraction amount of coding. For instance, the subsequent quote was coded with "security" throughout selective coding:

"*An insecure SaaS placement might permit a malicious user to handcraft parameters which bypass security checks as well as access very sensitive information of tenants. Placing SaaS elements to a weak server might result in possible security risks to these pieces and also, so, might violate Service Level Agreements (SLA) constraints.*"
---Case D14

Additionally, in this study, the code "prevented problem" was introduced as an issue that ideally was talked about to have been fixed before being an issue. For instance, the subsequent quote was coded with the codes "access "vulnerability": and problem"

"*Access authorization operation from various sources sharing techniques are described to occur as well as deploy the entry control program in SaaS business model. Every tenant should be independent and can authorise subtenants of its access to get into their own resources, of which includes private SaaS services and data.*"
---Case D34

2.3. Second-order headings

The thematic synthesis was carried out based on [13] work on SaaS reliability measurement metric. Thus, all the problems, reasons and solutions identified were coded and synthesised into different themes. We looked at the different activities of SaaS reliability measurement metric and they are operational, security, fault tolerance, support and monitoring. Financial implication in the framework was not included in the thematic synthesis because SMEs adoption of SaaS is only an enabler of SMEs businesses according to [14]. The choice to use these themes serves as indicator for the themes were not choose beforehand but were rather identified as issues.

2.4. Evaluation of criticality

We selected probably the most crucial problems for every case to be able to see which problems had the biggest impact on adoption. The number of the most crucial problems was not constrained. The two conditions for selecting the most crucial problems are the most serious issues that prevented SME from adoption of SaaS, and the most critical enablers that enabled SME adoption of SaaS.

Enabling factors were collected in some instances, no crucial issues were mentioned, but a few crucial enablers were emphasised. Probably the most crucial problems were extracted by three different methods. These methods are elaborated as explicit, implicit and causal.

Explicit: in case the content entirely emphasised an issue, or even in case it had been mentioned explicitly in the content that an issue was probably the most crucial, subsequently that problem was selected as an explicit issue. For instance, in case SC4, in which several issues have been provided, one was emphasised as probably the most critical:

"*Exclusively, SaaS applications involve executable business processes or maybe workflows, variation factors are spread with the workflow definitions, and therefore the profitable customisation of workflow driven SaaS applications relies upon current workflow customisation approaches as well as customisation techniques.*"
---Case D4

Implicit: This is when the issues were interpreted, in any case, could be viewed as probably the most crucial. These interpretations had been compared in

between the writers to mitigate bias and comprehensive explanation of the process.

Causal: the sources provided in the articles had been taken into consideration, by thinking about the additional main causes of SaaS adoption problems, the reasons and the solutions. The issue might be viewed as critical, though it was caused by the flexibility, in order to achieve guarantee service availability and so forth.

2.5. Reliability and validity of reviewed literature

The first author, causing single researcher bias, which had to be mitigated, first performed the search filtering, data extraction and synthesis. The search bias was mitigated by constructing the review protocol according to the guidelines by [11]. To avert single researcher's bias, the other author reviewed the review protocol.

The mitigation for selection bias has two experts produce impartial inclusion/exclusion on random sample selection of 255 articles out of the total 682. The random sampling was carried out to reduce the effort necessary for assessing the validity.

This accounted for moderate representation in the number of selected articles for the discussion in the study [15].

Bias in the criticality assessment was mitigated on the 42 articles selected by using the exclusion condition; there have been 38 complete agreements, 4 partial agreements in between the supervisor and the student, which means that several of the selected codes had been the same for most situation in the final selection of articles. The reliability and the validity process had an effect not just on the selected criticality assessments but on the codes, as well which enhanced the dependability of the study for the SLR.

2.6. Selection of articles for review

When extracting information from the forty-two selected articles for the literature review, it was distinguished that several of the articles did not include some information regarding issues on SaaS adoption studies. The articles that did not contain any extra problems are D5, D21, D17, D46 and D26. Article D48 contained issues but had been duplicated to Article D2, which analysed exactly the same situation.

All the cases reported throughout the years 2013 and 2018 are not especially shocking, because SaaS gained attention in the 20th century but studies in this specific region begin gaining recognition within the last ten years. Nevertheless, over fifty percent of the cases had been reported prior to 2010, which reveals an increasing interest in the topic.

Not all the articles contained quotations regarding issues on SaaS adoption had been reported. For instance, papers D28 and D23 checked for comprehensive explanation, but did not list any issues. In comparison, two papers that had probably the most quotations were D14 with 38 quotations and D34 with 13 quotations. That is mainly because these two articles exclusively described issues as well as difficulties that differs from the context in this study without considering any found issues.

2.7. Assessment of literature for quality

They contained explanations on methodology. Twenty-one (21) articles are descriptive reports. Two articles directly analysed the issues or maybe challenges (D6, D30), and consequently, it was decided in the study to never sort the outcomes according to the cause.

3. Result of the literature search

Causally in total, seven (7) relationships were identified in the themes from the SLR. These are categorised as SaaS operation problem with SaaS support, which had been the most connected themes. SaaS problem and SaaS support are regarded as personalization, reliability, and multi-tenancy problem. SaaS security and fault tolerance are performance and overcloud problem. The total number of identified SaaS problem are fifty-eight (48).

3.1. SaaS problem

Organisations purchase enterprise software with the intention of achieving smooth running for their enterprise-wide business processes. Organisations do this by paying for the enterprise-wide business application software up front. In the early 1990s, SaaS emerged as a new software application-licensing model [16].

Software-as-a-Service (SaaS) business model is revolutionalise around the way software is shipped to organisation. Adopting SaaS comes with several advantages viz a viz lowering expenses, increase efficiency, leverage newest technology with no extra purchase, improve focus on core company procedures and improve innovation [13]. For SMEs, SaaS can be handy not just in reinforcing the IT capability but enhance organisation operations simultaneously [15].

As the workloads of applications, resources capacities and dynamics continue to change over time, there is need for placement, which is reconfigured to keep the SaaS performance, and to optimize the used resource [16].

3.1.1. Reason for SaaS problem. SaaS service providers face optimisation problem, as they need to satisfy customers (SMEs), satisfy the previously established entity as well as proved demands for dynamic optimisation problems. Authors [17] presented two essentials including both IaaS and PaaS. SaaS adoption in research is able to encompass IaaS and PaaS adoption research. Additionally, SaaS differs from program service provider where users could select the service [18].

3.1.2. Solution to SaaS problem. SaaS offers not just the improvement of the whole info systems (IS) infrastructure, like the platform [15], but SaaS also supply a greater quality of information compared to the traditional techniques where the IS design as well as operated by a third party for SMEs. SaaS has a shorter and flexible period of growth in addition to improved scalability, which could boost SaaS to a greater quality than could in traditional approaches. SaaS adoption enhance the company tasks because of the very best business train method applicable to SaaS delivery model. SMEs that procure SaaS for the business operation have an additional advantage of using perfect practice process supplied by SaaS. This kind of benefits may boost the intention of SaaS procurement [15].

Adoption of SaaS offer SMEs the latest cutting-edge IT, which involves less operating and capital investment. SMEs has enough IT capability; it might lessen the danger of adopting SaaS since SMEs is going to have the capacity to overcome problems such as the shortage of experience and knowledge in software adoption. SMEs is able to reduce the danger of adopting SaaS in case the company doesn't have adequate financial resources, but the top management of IT understands and positively encourages the adoption of the technology [15].

3.2. SaaS operational problem

The operational theme consists of discussion in the literature on workflow, composite, multi-tenancy and applications. The discussion result is presented as follows:

Cloud computing allows the advancement of multi-tenant applications by sharing computing as well as storage resources. In a multi-tenant Software-as-a-Service (SaaS) structure, several tenants talk about an application program and supporting hardware [10]. Multi-tenant Software-as-a-Service (SaaS) programs reveal one runtime among many buyer organisations (tenants) [17]. Multi-tenancy enables numerous clients (i.e. tenants) being consolidated into identical functional phone system. Multi-tenancy describes the capability that an individual structure has to deliver several tenants with only one software instance [18]. Multi-tenancy is an architectural design for SaaS apps, which allows different consumers (tenants) to talk about the very same example of a software [19].

Multi-tenancy introduces complexities in program development, management and deployment. When sharing an actual example of an application, it is essential to make sure performance, availability and security [10].

A workflow is an executable software program artefact able to automate a big number of methods in applications like company processes (e.g. creating a session appointment in a hospital) as well as batch processing (e.g. document development as well as distribution out of substantial quantity of raw data). In order to allow reusability and flexibility, SaaS may be presented in a composite form, wherein a pair of mingling application program and information pieces cooperate to develop a higher-level practical SaaS [20]. A composite SaaS is made up of loosely coupled communicating subsystems each one with specific company run to develop a higher-level practical phone system. Nevertheless, adopting the composite design to supply SaaS might add various problems to the cloud provider in terms of source management [20].

3.2.1. Reason for SaaS Operational Problem. Run-time customization entails a broad range of program artefacts like user interfaces, databases, web services, business process or workflow definitions. There is no single approach when choosing multi tenancy architecture, therefore, SaaS offer architects with help for producing proper trade off choices when following a workflow customization program [17].

The consequences of SaaS architects are due workflow customization tactic at the beginning of the workflow driven development life cycle. SaaS program can be difficult to return in later stages but the acknowledgements of the crosscutting effect on several other application artefacts.

Different customization techniques have different effect on varied and occasionally conflicting factors of SaaS, which range from the support for scalability to the compatibility with growth operations activities. It is virtually impossible for customers signing up to the application with the SaaS licensing model to personalize the application based on their preferences [21]. Buyers that desire to customize enterprise applications need to fit the unique on premises licensing style and incur a customization cost.

3.2.2. Solution to SaaS Operational Problem. To have several variants of the application program in one runtime instance, SaaS suppliers define variation areas in the application program on the foundation of several variants that could be tailored for run time [17]. SaaS providers offer tenants with management dashboard to customize the application based on specific needs of their organisation. An outcome is

not possible to determine a one size fits all. SaaS architects need to make an architectural trade off when following a specific workflow customization technique. An optimum trade off, nonetheless, is possible when all quality implications of workflow personalization techniques are popular. Customization is preferably done in a self-service fashion so that the principal stakeholders, specifically the SaaS provider as well as tenant organisations, do not need to communicate in the customization phase.

SaaS placement in a cloud datacenter is another problem to be resolved by designing algorithm to conquer an efficient optimisation challenges. The business model resulting from the algorithm design aimed at "economies of scale" and a software delivery model with support for the team.

3.3. SaaS support problem

The placement of SaaS software program and data components is fully necessary as it might influence the protection functionality level of the SaaS and user experience.

An insecure SaaS placement might permit a malicious user to handcraft parameters, which bypass security checks as well as access very sensitive information of tenants. Placing SaaS elements in a weak server might result in possible security risks and might violate Service Level Agreements (SLA) constraints. Datacenter is essential to create security aware SaaS placement techniques with risks that are high, separated from SaaS with lower risks.

In handling these issues, the formal definition associated with new tenant-based access management design based on access management role (ARBAC) for STA and MTA in service-oriented SaaS called (TMS-ARBAC) was presented [22]. The need for the provision of IT decision rights between company and IT units in the application level. This add the governance of uses sent on premise and all those shipped with a SaaS model [23].

3.3.1. Reason for SaaS security problem. The inferred security rating for every hosting computer is depends on the threat level of the hosted SaaS pieces. It is utilised to penalize probably the most susceptible servers. Security requirements is still a challenge since Cloud computer users are reluctant when selecting a SaaS Service Provider [23]. Security challenges are data leakage, multi-tenancy, data sharing, resource location, and availability. These problems include access, data locality, and data breaches, information confidentiality and segregation, authentication, virtualization vulnerability, and authorization. Security risk is frequently created out of the advantage of SaaS service providers, vulnerability, and security threats. Authorizing businesses on different resource sharing techniques are develop and deploy as entry control program in SaaS business model. Every tenant should be autonomous and will authorize the subtenants to get their own resources, which includes private service pieces and information. Each subtenant might not just inherit its tenant's information, but also additionally customize their own applications and permit others to use their resources [22].

3.3.2. Solution to SaaS security problem. The SaaS Security Problem solution is associated with the presented costs and SLA. SaaS providers and owners should discuss the security requirements with SaaS Service Providers in an effort to select the earth that best suits their needs. Many studies attempted to tackle security worries by permitting cloud customers as well as suppliers to establish a deal protection needs in different extensions of security SLA models [15].

Presently, almost all current access management designs for MTA are derived from role-based access management (RBAC) and management role based access control (ARBAC) [22]. Aside the security strategies supplied by MTA, STA access management must handle the following issues such as privacy sharing tenants and subtenants private service pieces and data. A tenant can grant own resources like information, customized elements to their subtenants, and might not allow their tenants or method administrators the ability to access them. A tenant manages their own resources, produce subtenants and grant source entry privileges to them [22].

3.4. SaaS support problem

Quality of Service (QoS) are measurable non-functional characteristics, which distinguish services and forms the grounds for program selection [22]. Majority of the current cloud service choice techniques have ignored serious QoS dimensions demands including privacy and security, accountability, usability, and guarantee formulating a foundation for cloud service ranking and selection [24].

With SaaS OverCloud version, the needs make a razor thin overlay level to facilitate the SaaS compatibility over heterogeneous cloud infrastructures and platforms [24]. SaaS OverCloud design facilitate the SaaS compatibility over heterogeneous cloud infrastructures and platforms.

In contemporary SaaS based cloud services personalization is able play a crucial part like the internet or mobile applications, this make provision for a good motivator to better understand the way to enhance SaaS efficiency. Nevertheless, less interest

is given on the specific provider's job relationship in affecting SaaS efficiency [25].

3.4.1. Reason for SaaS support problem. The discrimination of solutions based on QoS info is a treatment towards decreasing service option overload as the cloud service. QoS design includes Performance Indicators Key (PIK) for decision-making. QoS design comprises the key equivalent attributes of every service, and ideal for matching user QoS specifications to services' QoS feature. As much operating program, a multi-tenant SaaS software should ultimately go through upgrade, and evolution which are needed to run for run time, and maximally admire tenant SLAs. A number of procedures are usually applied to reduce impact of SaaS operation during a powerful enactment of gradual tenant and an upgrade by tenant activation of an improvement in term of workloads [26].

The issue involves determining what applications to set up on every computer bunch of the provider and the way to assign buyers to the clusters in an effort to supply backup and primary service to clients providing of a cluster disaster, while reducing full price [14]. With each software request by each customer support, and the readily available capacities of the computer system clusters. The requirement to designate buyers to unique clusters is thus paramount to effective working of SaaS in the customer's premises. Furthermore, the importance to determine what programs to set up on every group like that in case a person is given to a certain bunch, subsequently most apps requested by the buyer should be placed on that group [14].

3.4.2. Solution to SaaS support problem. [24] proposed the use of similarity metrics to get ranking cloud services in cloud e-marketplace context based on QoS characteristics. The standing is attained by locating the similarity in between the person provided QoS specifications and those of various other people in the past. Six QoS attributes with a regular information format are utilized as a qualifier for the values for flexibility, security, and usability [15].

With SaaS OverCloud version, by fulfilling the next needs, we try to make a thin overlay level, which could facilitate the SaaS compatibility over heterogeneous cloud infrastructures and platforms [15].

3.5. SaaS fault tolerance problem

Application placement is another major challenge that relates to the product offering and location. Therefore, there is a need to decide the locations of the products and the infrastructure that will support the allocation of the product to customers [1]. Authors [1] looked at resource as well as SLA constraints for optimum application placement. Each applicant placement program is represented by a composite particle (a pair of elementary particles), wherein every particle belongs to a feasible hosting server. The quality of placement, the computation time, along with dependencies between SaaS elements will be the primary problems that have being resolved much more in depth [27].

SaaS Fault Tolerance provided a set of computation as well as storage servers in the information facility with the cloud communication network along with a composite. SaaS aim is to determine the placement of SaaS program and information parts upon the computation and storage servers like ideal SaaS efficiency, the SaaS execution period is minimized and maintained at a reasonable level as the functionality of hosting servers.

As the variety of SaaS program gets bigger, so too is the needs of owners in customization terminology of software and program quality. Customization of software suggests a selection of program to meet up with the demands of many unique SaaS client [28].
The issues mentioned previously, is emphasised by [25] that regardless of the increasing trend of SaaS, the failure rate of SaaS is many and high service providers claimed to have very poor overall performance. Some are suspicious about the service providers' potential to notice the expected performance.

3.5.1. Reason for SaaS tolerance problem. SaaS placement has reusability and flexibility Fault Tolerance Problem but provides far more complexity to the associated control undertaking with a composite SaaS, particularly in the distributed and very powerful cloud environment.

For every placement, it is important to meet up with placement demands from the most effective servers from the associated server farm the dependencies between SaaS elements.

To fix these issues, [25] proposed a way dependent on particle swarm optimisation (PSO) to control the placement of SaaS program and information pieces combined with their constraints, requirements, and dependencies. The technique utilizes an algorithm where sub swarms of servers are built to cooperate and develop via a selection of iterations with a cooperative learning program. The greatest servers among computing servers cooperate to create the perfect positioning program of the entire SaaS storage servers. Additionally, [28] proposed the SaaS cloud personalization framework that allowed the facilitation of the group of tastes and the distribution of corresponding SaaS services. The goal of [28] strategy is in providing an integrated SaaS structured environment solution whereby provisioned with enhanced personalization quality and efficiency.

4. Limitation of the study

The strategy used in this study assists with the accurate retrieval of literatures relating to SaaS adoption. The limitation arises from the fact that the sturdiness of the evidence whether in the everyday relationships are real, or most vital issues were most crucial in reality; and whether the solutions are in fact the most appropriate for the problems.

Additionally, not all related articles could be downloaded, because the university library does not have subscription for some of the academic journals returned in the search. This are in the region of 0.5 article out of every 30 articles returned. In addition, book chapters and e-Book are not included as part of the downloaded articles.

The data collected from the number of articles retrieved may have been interpreted with researcher's bias. The filtering technique contained interpretative elements and therefore results from them may vary. During information extraction, most of the identified problem may be simply interpretations of the authors. This is applicable to the interpretation of the solutions and causes towards the view that the authors find as a lesser number of issues than could be reality.

5. Conclusion

A systematic literature review was conducted on major academic databases in management information systems to unravel the problems experienced by SMEs during SaaS adoption. The SLR conducted in this study, use search strategy that produced 682 separate papers that were filtered and synthesised to identify SME adoption problems, the reason for the SaaS adoption problems and the solutions of these problems from previous studies.

The outcome of the SLR activity yields 42 papers used in the final literatures for the study. The discussion on the outcome of the SLR was based on the following: SaaS problem, SaaS operational problem, SaaS security problem, SaaS support problem, and SaaS fault tolerance problem.

Although we identified various kinds of SaaS adoption problems as well as the criticism, we cannot articulate how these problems are spreading and the reasons for some problems tend to be more critical compared to others. These shortcomings could be answered in future studies, which will investigate how theoretical framework might help the study understand the context better.

In this paper the research questions were raised and answered. The questions raised as solution appeared in the previous studies remain to be known if the same problems will affect SMEs adoption of SaaS. Future study will conduct fieldwork to unveil the truth regarding the problems.

6. References

[1] Amiri, A. (2017) Application placement in computer clustering in software as a service (SaaS) networks. *Inf Technol Manag*, 18: 161-173.

[2] Zhaojun Y., Jun, S., Yali, Z., and Ying, W. (2015) Understanding SaaS adoption from the perspective of organizational users: A tripod readiness model. *Computers in Human Behavior*, 45: 254-264.

[3] Skilton, M. (2010) Building return on investment from Cloud Computing http://www.opengroup.org/cloud/whitepapers/ccroi/index.htm (.Access Date: 3 August, 2018)

[4] Winkler, T. J. and Brown, C. V. (2015) Horizontal Allocation of Decision Rights for OnPremise Applications and Software-as-a-Service. *Journal of Management Information Systems*, 30 (3): 13-48.

[5] Sultan, N. A. (2011) How SMEs can manage. *International Journal of Information Management*, 31: 272-278.

[6] Wu, W. W., Lan, L. W., and Lee, Y. T. (2011) Exploring decisive factors affecting an organization's SaaS adoption: a case study. *International Journal of Information Management*, 31(6): 556-563.

[7] Mezni, H., Sellami, M., and Kouki, J. (2018) Security-aware SaaS placement using swarm intelligence. *J Softw Evol Proc.* https://doi.org/10.1002/smr.1932 (Access Date: 23 June, 2018).

[8] Han, J. and Kim, J. (2017) Design of SaaS overCloud for 3-tier SaaS Compatibility over cloudbased multiple boxes. *Proceedings of CFI'17,* Fukuoka, Japan.

[9] Palos-Sanchez, P. R., Arena-Marquez, F. J., and Aguayo-Camacho, M. (2017) Cloud computing (SaaS) adoption as a strategic technology: results of an empirical study. *Mobile Information Sysytems*, Article ID 2536040. https://doi.org/10.1155/2017/2536040 (Access Date: 4 August, 2018)

[10] A. Jumagaliyev, and J. Whittle (2016) "Model-Driven Engineering for Multi-Tenant SaaS application development", *CrossCloud'16*, London, United Kingdom.

[11] Fernndez-Alemán, J. L., Seor, I.C., Lozoya, P. A., and Toval, A. (2013) Security and privacy in electronic health records: a systematic literature review. *J. Biomed. Infor*, 46(3): 541-562.

[12] Corin, J. and Strauss, A. (1988) Grounded theory research: Procedures, Canons, and Evaluative Criteria. 13(1): 1-19.

[13] Vidhyalakshmi, R., and Kumar, V. (2018) CORE framework for evaluating the reliability of SaaS products. *Future Generation Computer Systems*, 72: 23-36.

[14] Amiri, A. (2016) Application placement and backup service in computer clustering in Software as a Service

(SaaS) networks. *Computers & Operations Research*, 69: 48–55.

[15] Kim, S. H., Jang, S. Y., and Yang, K. H. (2007) Analysis of the Determinants of Software-as-a-Service Adoption in Small Businesses: Risks, Benefits, and Organizational and Environmental Factors. *Journal of Small Business Management*, 55(2): 303-325.

[16] Chainbi, W. and Sassi, E. (2008) A multiswarm for composite SaaS placement optimization based on PSO. *Software Practice Experiment*, 48: 1847-1864.

[17] Makki, M., Van Landuyt, D., Lagaisse, B., and Joosen, W. (2018) A comparative study of workflow customization strategies: Quality implications for multi-tenant SaaS. *The Journal of Systems & Software*, 144: 423-438.

[18] Rico, A., Noguera, M., Garrido, J., Benghazi, K., and Barjis, J. (2016) Extending multi-tenant architectures: a database model for a multi-target support in SaaS applications. *Enterprise Information Systems*, 10(4): 400-421.

[19] Bezemer, C. and Zaidman, A. (2014) Performance optimization of deployed software-as-a-service. *The Journal of Systems and Software*, 87: 87-103.

[20] Hajji, M. A., and Mezmi, H. (2018) A composite particle swarm optimization approach for the composite SaaS placement in cloud environment. *Soft Compu*, 22, 4025-4045.

[21] Li, S., Cheng, H. K. Duan, Y., and Yang, Y. (2017) A Study of Enterprise Software Licensing Models, *Journal of Management Information Systems*, 34(1), 177-205.

[22] Zuo, C., Xie, M., Qi, G. and Zhu, H. (2017) Tenant-based access control model for multi-tenancy and sub-tenancy architecture in Software-as-a-Service, *Front. Comput. Sci*, 11(3), 465-484.

[23] Winkler, T. J. and Brown, C. V. (2015) Horizontal Allocation of Decision Rights for OnPremise Applications and Software-as-a-Service, *Journal of Management Information Systems*, 30(3), 13-48.

[24] Ezenwoke, A., Daramola, O., and Adigun, M. (2018) QoS-based ranking and selection of SaaS applications using heterogeneous similarity metrics, *Journal of Cloud Computing Advances, Systems and Applications*, 7(15), 1-12.

[25] Chou, S., Chang, Y., and Hsieh, P. (2018) Understanding the performance of software-as-a-service based on service-dominant logic, The *Service Industries Journal*, 34(2), 19-32

[26] Gey, F., Van Landuyt, D., and Joosen, W. (2016) Evolving Multi-tenant SaaS Applications through Self-Adaptive Upgrade Enactment and Tenant Mediation, Proceedings *of the 2016 IEEE/ACM 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems.*

[27] Mezni, H. and Kouki, J. (2017) A multi-swarm based approach with cooperative learning strategy for composite SaaS placement, *Proceedings of the SAC'17,* Marrakesh, Morocco, 399-404.

[28] Fan, H., Hussain, F. K., Younas, M., and Hussain, O. K... (2015) An integrated personalization framework for SaaS-based cloud services, *Future Generation Computer Systems*, 53, 157-173.

