Towards Software Engineering Paradigm for Software as a Service

Niyati Baliyan
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee, Haridwar, India
niyati.baliyan@gmail.com

Sandeep Kumar
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee, Haridwar, India
sandeepkumargarg@gmail.com

***Abstract*—The Software as a Service model of Cloud Computing offers economies of scale through the pay per use model; however, it renders the modern software very different from traditional software. Hence, there is a need to adapt Software Engineering approach in a manner that will make the development process and delivery of Software as a Service more efficient and of high quality. After performing literature review, a classification of ongoing research in this direction of adaptation is presented. Various research gaps in the areas of software development process, software reengineering, measurement, metrics, and quality models targeted at Software as a Service are identified, which can be a first step towards the definition of standards and guidelines for Software as a Service development.**

Keywords—cloud computing; software as a service; quality; software engineering; reengineering;web engineering

I. INTRODUCTION

Sometimes known as subscription software, Software as a Service (SaaS) essentially separates software ownership by the SaaS provider from its use by the SaaS customer. SaaS is a new model which delivers software as utility service and charges on a per-use basis. SaaS benefits include separation of concerns and economies of scale, among others [1]. SaaS is the future of the Web, providing the ability to compose services dynamically, thereby overcoming the inherent limitations of traditional software, associated with its use, deployment, and evolution.

It is known that Web applications differ from traditional software in terms of several functional and non-functional requirements, continuous evolution, and aesthetics among others [2]. Large scale Web system development requires a well-defined process in order to ensure quality and maintainability. Therefore, existing Software Engineering principles, and methods should be adapted to Web development for SaaS [3].

The notion of viewing SaaS from a Software Engineering perspective is relatively new. In this work, it is aimed to explore how the business model called Cloud Computing needs a more standardized process for delivery of SaaS, which is fit for applications with reusable components. By developing a benchmark Software Engineering process for SaaS, the benefits of SaaS could be exploited to a greater extent.

There is a need to analyze the inter play of Software Engineering and reengineering process for SaaS. Certain Software Engineering models for measurement of quality may be developed for SaaS in order to serve as a guideline for improvement of quality of service of Cloud based software applications.

The remainder of this paper is structured as follows. In Section 2, we outline the review process used here. Section 3 describes the hierarchical organization of relevant studies into groups and subgroups based on a common theme of objectives. The state of the art in Software Engineering paradigm for SaaS is reviewed in Section 4, while Section 5 presents some observations made from the review. Section 6 discusses current research challenges before presenting concluding remarks in Section 7.

II. REVIEW METHOD

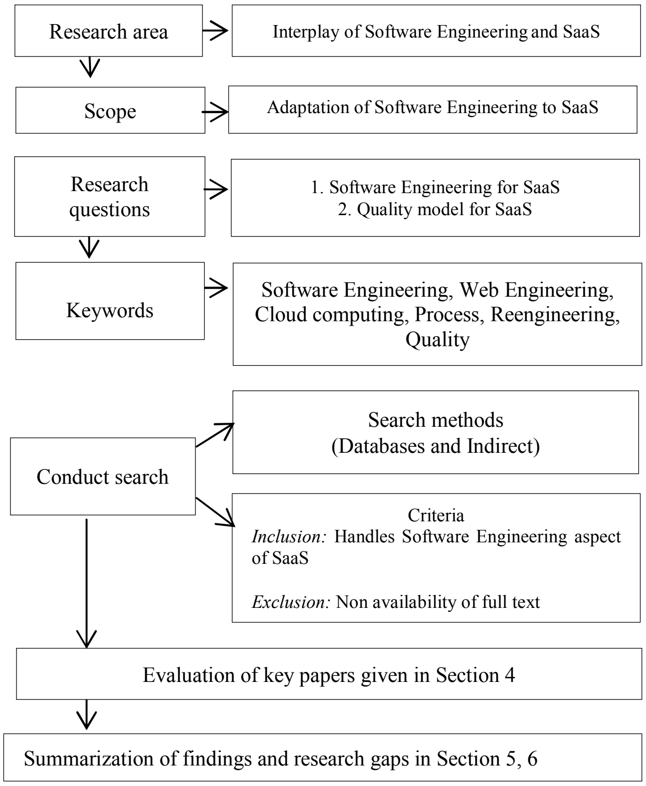
Research area
Interplay of Software Engineering and SaaS
Scope
Adaptation of Software Engineering to SaaS
Research questions
1. Software Engineering for SaaS
2. Quality model for SaaS
Keywords
Software Engineering, Web Engineering, Cloud computing, Process, Reengineering, Quality
Search methods
(Databases and Indirect)
Conduct search
Criteria
Inclusion: Handles Software Engineering aspect of SaaS
Exclusion: Non availability of full text
Evaluation of key papers given in Section 4
Summarization of findings and research gaps in Section 5, 6

Fig.1. Review strategy and process

978-1-4799-5173-4/14/$31.00 ©2014 IEEE

Fig.1. shows the process followed in order to conduct the review. The scope of our study is limited to include adaptations of Software Engineering in light of SaaS. Section 5 includes certain findings made from the literature review in Section 4. Section 6 enlists some open issues in research on Software Engineering for SaaS development.

III. TAXONOMY OF LITERATURE

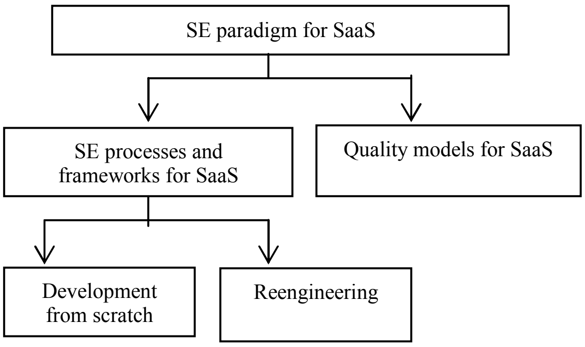

Fig.2. Classification of Literature

Certain studies are found to have common research objectives; they have been grouped together as shown in Fig. 2. At the highest level in the hierarchy, lies the area of interest, i.e. Software Engineering paradigm for SaaS.

This is bifurcated into processes (Section IV (A)) and quality models (Section IV (B)). Software Engineering for SaaS deals with software development and reengineering methods suited for Cloud based applications. In other words, an answer to the following question is sought: how will the Software Engineering process change with the advent of delivery of SaaS on the Cloud. Furthermore, Software Engineering processes can be classified into: development from scratch, and reengineering approaches. In literature, only a few adaptations of quality models for SaaS are found, these have been discussed in Section IV (B).

IV. REVIEW OF LITERATURE

The relevant studies are discussed in this Section as per the groups made earlier in Section 3.

A. Software Engineering Process and Framework for SaaS

1) Development from scratch for SaaS

La and Kim [4] define certain intrinsic and extrinsic characteristics of SaaS, and describe a systematic process to develop SaaS applications. The defined characteristics of SaaS are shown to be handled in one or more phases of the proposed SaaS development life cycle. The life cycle is shown to result in highly reusable SaaS. More comprehensive evaluation of the development technique can be carried out in the future.

Joshi et al. [5] underline the absence of any holistic view of life cycle of virtualized services delivered on a Cloud environment. A life cycle comprising of five phases – requirement, discovery, negotiation, composition, and consumption is proposed. For the purpose of monitoring service quality, authors also propose certain metrics which span over one or more phases in the life cycle. The proposition of service life cycle and its monitoring is novel and potentially useful to define deliverables at each stage. This life cycle for virtualized services may be accepted as the reference benchmark with time and usage.

Guha and Al-Dabass [6] give an overview of challenges facing SE in the era of Cloud Computing. They propose an adaptation of the agile life cycle model of software called Extreme Programming. Though the notion of developing Cloud based applications using an agile methodology seems reasonable, a practical scenario where adherence to such life cycle has proved to be efficient is missing. Careful examination and validation of many large scale SaaS needs to be performed in order to standardize any life cycle.

Sharma and Sood [7] focus on capitalizing Model Driven Architecture (MDA) for development of SaaS applications. The platform independent nature of SaaS calls for a SaaS specific development methodology. The transformation of a model to SaaS could be automated in the future. Moreover, the SaaS development framework could be extended to cater to interoperability of multiple SaaS, which are developed using this method.

Sharma et al. [8] advocate the application of Service Oriented Architecture (SOA) at the backbone of SaaS development in order to utilize full capability of SaaS. Authors analyze the convergent effect of SOA, MDA and SaaS in a way that results in a good quality SaaS.

2) Reengineering for SaaS

Zhou et al. [9] describe the steps involved in migrating legacy enterprise software to Cloud, using ontologies. Here, loosely coupled modules are identified as prospective service candidates for reuse. This is achieved via building of separate ontologies for the source code, the application framework, and the data. Authors also develop a prototype toolkit using Protégé tool, ATL toolkit and Hibernate framework, which supports such migration. Their approach is validated through an open source ERP + CRM system. In the future, the extent of human intervention in this semi-automated approach can be estimated.

Song et al. in [10], [11] identify SaaS characteristics namely: configurability, multi tenancy and scalability as different from traditional Web applications. Various command sets have been defined corresponding to these characteristics. For wide acceptance, the proposed methodology needs to be available as an easy to use tool.

Saleh [12] draws attention to a lot of effort required in reengineering non SaaS applications to fit SaaS paradigm. Therefore, author suggests the migration of traditional Web applications to SaaS. The novelty of the author's conceptual framework for migration lies in its focus on configuration and customization layer, which account for changing business and user requirements. The process and effect of migration has been viewed at separate layers in the framework. Various layers in the proposed framework are – back end layer and, configuration and customization layer; responsible for log files, and business logic, UI, workflow and database configuration respectively. A popular Human Resource Management (HRM) system called OrangeHRM was employed as case study for this approach. However, practical demonstration and validation of the migration framework, which is still ongoing, could prove significant for the adoption of non-SaaS to SaaS migration approach.

B. Quality models for SaaS

The work most relevant to SaaS quality is by Lee et al. [13], who present a comprehensive model for assessing the quality of SaaS. First, quality attributes of SaaS that differ from software are identified. Next, additional SaaS quality attributes relevant to SaaS are measured. The approach used in this work is simple and direct; it needs to be checked against real SaaS applications for accuracy of results.

Cancian et al. [14] identify the lack of any software quality model which has been customized to fit the SaaS scenario. Based on interviews and survey, they categorize quality criteria for the service delivery model of Cloud into two classes, i.e., product and process quality criteria and point out that the establishment of such criteria is useful to service customers for selecting services and service providers, and to service providers for improving the quality of their services. In the future, SaaS quality criteria can guide the evaluation and improvement of a SaaS development process.

Garg et al. [15], [16] present a framework SMICloud that measures standard QoS attributes predefined by CSMIC. Their additional contribution lies in highlighting the challenges to develop metrics for each of these quantifiable attributes, and in the ranking of Cloud services based on quality.

In their paper, Wang et al. [17] assert that a Cloud service is provided at various levels and involves many features, therefore Cloud service varies greatly and its evaluation is thus difficult. Authors evaluate Cloud service providers on computing power, seamlessness, storage and security; this serves as an indicator of quality of service to the Cloud user. They use monitored data on quality and evaluate it through fuzzy logic control. Their approach is by far most effective and accurate for service-oriented Cloud Computing. However, not all SaaS may be based on SOA where the logical unit of functionality is a service. The concept of SaaS needs to be viewed on a larger scale, where the delivery of entire software in form of service is stressed upon. Wang's simulation results have been validated; and their approach to handle unpredictability can well be applied exclusively to SaaS.

Wen and Dong [18] propose a quality model to evaluate SaaS from different perspectives, i.e., provider, customer and platform. However, the hypothesis about quality evaluation framework and the resulting maturity levels of SaaS is not backed by any real world applications or concrete evidences of correctness. A possible future work is to find evidences while handling quality from every aspect and to find ways of developing tools for the same. Also, quality evaluation from one or more perspective can be done at run time.

Some quality factors have been identified from literature and a fuzzy model has been proposed to assess SaaS quality. This model may assist a SaaS customer to choose a higher quality service and may also serve as a guideline to SaaS provider to improve the quality of service provided by him in [19].

V. Observations

Following observations were made from review of relevant literature.

While most of the development or reengineering approaches are based on reverse engineering to obtain models and meta models, only a few are based on computational modeling or relational language. Some works employed variations of existing SE processes or notations such as: Component Based Software Engineering, Business Process Modeling Notation, Software Product Lines and others. Theoretical frameworks were proposed by a few in an attempt to define Software Engineering for SaaS. Researchers have adapted agile process model for Cloud based software development and handled the role of Cloud provider in all life cycle stages (Fig. 3).

The most important characteristic of the Cloud based software is their ability to adapt with changing requirements and with changing context, therefore the element of agility is justified [20].

Measurement of quality of SaaS appeared to be a potential area of research since quality is an important criterion for choice of SaaS and existing software quality models are not fit for services, due to fundamental differences in their characteristics.

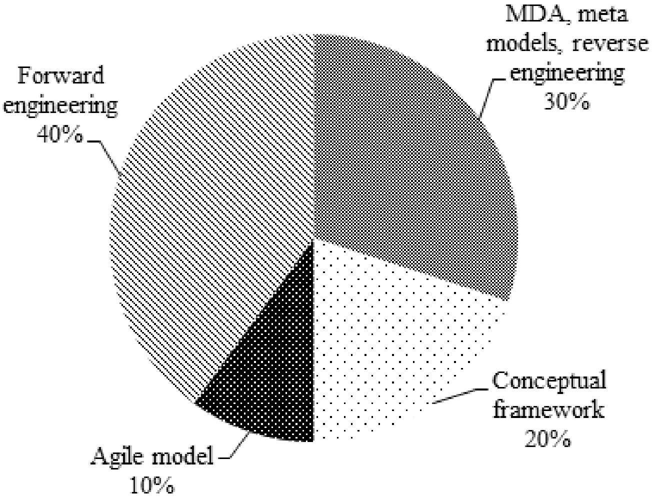

Fig.3. Methodologies used for adaptation of Software Engineering process to SaaS

Fig.4. shows the contribution of different quality modeling techniques used for SaaS in the literature.

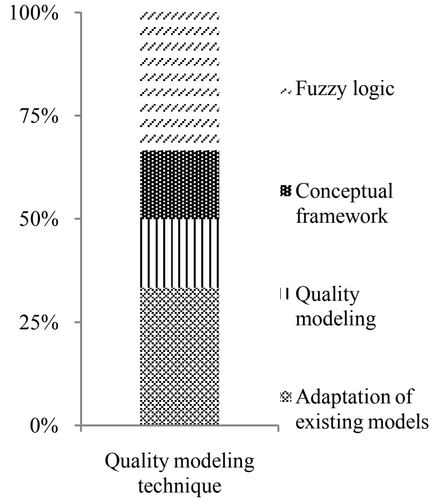

Fig.4. Techniques used in quality modeling for SaaS

VI. Research Gaps

A few open issues have been identified through literature survey and analysis. In addition, few anticipated solutions have been provided wherever applicable.

In the face of growing changes in the way software is developed and delivered, the adaptation of Software Engineering to SaaS model seems rewarding. Different aspects of Software Engineering that need adaption for Web Engineering in the face of SaaS may be identified and few of them may be further examined. Since the characteristics of SaaS differ greatly from that of legacy software; the underlying software principles, processes and metrics should be adapted. In this manner, Software Engineering will maintain relevance and ensure quality software product and an efficient development process. One or more of these directions can be explored for further research. It is only when the old school approach comprising standards and principles is merged with agility in today's development for the existing SaaS, that benefits may be exploited to the fullest.

There exist no universally accepted frameworks for development of Cloud software which handles both development from scratch and reengineering. One may examine what lacks in the existing process models for SaaS and thus explore the possibility of modifying an existing Software Engineering process model for their development or devise a novel one, which handle the case of both development from scratch and reengineering. Development from scratch is particularly useful for small and medium scale enterprises, startups which are numerous in India; while established industry giants will sooner or later migrate to Cloud, given their advantages of cost cutting and automation respectively.

Since we cannot improve what we cannot measure, the feasibility of tracking relevant metrics during and after SaaS development can be considered. Among others, it might be tried to explore the need and benefits of adapting certain metrics for SaaS. Novel metrics and/or models for quantifying cost, quality and other attributes of SaaS may be developed. Another research issue is to propose a quality model that captures various attributes which determine SaaS quality.

While reengineering existing Web applications to migrate to Cloud, a lot of pre-processing is required in terms of reverse engineering. One way to avoid it is to develop standard models for developing SaaS that incorporate sound engineering principles from beginning itself, thus reducing the cost and effort of pre-processing. Another possibility is the development of complex automated tools for migration of WS and legacy software to Cloud. However, there is a trade-off between initial cost and automation. Nevertheless, in the long run, most applications will have to be Cloud compatible; hence it is an investment worth consideration.

There can be two possibilities for SaaS, either build from scratch or reengineer existing web services and legacy software for SaaS. Hence, the proposed Software Engineering model should encompass both these cases. Majority of literature is focused on reengineering as Cloud as a trend is currently catching up. Therefore, existing applications are transformed to exploit their benefits. However, the future Web shall see dominance of SaaS and hence they must be incorporated in the beginning phases of Software Development Life Cycle (SDLC). This strengthens our case in favor of essential and urgent custom Software Engineering life cycle models for Cloud.

Cloud technologies and related processes have not yet reached their full potential. Many capabilities and associated processes have not yet been developed and understood to a degree that allows their exploitation. In consequence, there is a demand for developing, adapting, extending, and maintaining software that supports Cloud Computing. It is widely unknown, which practices, techniques, and methods are effective for Cloud-related software development and maintenance and how to select appropriate techniques, methods, models and tools that are suitable for specific development goals and environments [21].

Role of Cloud provider needs to be incorporated in SDLC for Cloud SaaS. Cloud favorable technologies such as XHTML, JQuery, JavaScript, Python, AJAX, Ruby on Rails and agile development should be incorporated in SDLC for Cloud.

Functional and non-functional requirements differ for SaaS and legacy software. Multi tenancy and service candidacy of migrated software, need to be handled.

VII. Conclusion

The future shall see mainstreaming of Cloud business model; hence it is imperative to develop a framework for the process behind development of software on this platform. Our prime concern is to study Software Engineering discipline in terms of its coincidence with SaaS. The aim is to exploit maximum benefits from these fields and obtain quality software. It is not recommended that software development methodologies be simply integrated with service deployment, nor is it suggested to draw a straightforward linkage between software development and service deployment models. The two domains must be integrated in a standardized yet flexible manner. In addition, it becomes important to measure the quality of SaaS, in order to be able to improve it. Some of the interesting research problems among others are - adaptation of Software Engineering process models, quality models, measurement and metrics, for SaaS.

References

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, and I. Stoica. "Above the clouds: A Berkeley view of cloud computing." Dept. Elect. Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28, 2009.

[2] R. S. Pressman and D. B. Lowe. *Web engineering: a practitioner's approach*. McGraw-Hill Higher Education, 2009.

[3] S. Kumar and S. Sangwan. (2011). "Adapting the Software Engineering Process to Web Engineering Process." *Int. J. of Computing and Bus. Research.* 2(1).

[4] H. J. La and S. D. Kim. "A systematic process for developing high quality SaaS cloud services," in *Cloud Computing*, Springer Berlin Heidelberg, 2009, pp. 278-289.

[5] K. P. Joshi, A. Joshi, and Y. Yesha. "Managing the Quality of Virtualized Services," in *Proc. Annu. SRII Global Conf.*, 2011, pp. 300 - 307.

[6] R. Guha and D. Al-Dabass. 2010. "Impact of web 2.0 and cloud computing platform on software engineering," in *IEEE Int. Symp. Electron. System Design (ISED)*, 2010, pp. 213-218.

[7] R. Sharma and M. Sood. "Cloud SaaS: Models and Transformation," in *Advances in Digital Image Processing and Information Technology*, Springer Berlin Heidelberg, 2011, pp. 305-314.

[8] R. Sharma, M. Sood, and D. Sharma. "Modeling Cloud SaaS with SOA and MDA," in *Advances in Computing and Communications*, Springer Berlin Heidelberg, 2011, pp. 511-518.

[9] H. Zhou, H. Yang, and A. Hugill. "An ontology-based approach to reengineering enterprise software for cloud computing," in *IEEE 34th Annu. Comput. Software and Applicat. Conf. (COMPSAC)*, 2010, pp. 383-388.

[10] J. Song, F. Han, Z. Yan, G. Liu, and Z. Zhu. "A SaaSify tool for converting traditional web-based applications to SaaS application," in *IEEE Int. Conf. Cloud Computing (CLOUD)*, 2011, pp. 396-403.

[11] J. Song, Z. Yan, F. Han, Y. Bao, and Z. Zhu, Z. "Introducing SaaS capabilities to existing web-based applications automatically," in Proc. *14th Asia-Pacific Web Conf.*, 2012, pp. 560-569.

[12] E. Saleh. "Migrating Traditional Web Applications into Multi-Tenant SaaS," in *Proc. 6th Ph. D. Retreat of the HPI Research School Service-oriented Syst. Eng.*, 2013, pp. 145-155.

[13] J. Y. Lee, J. W. Lee, D. W. Cheun, and S. D. Kim. "A Quality Model for Evaluating Software-as-a-Service in Cloud Computing," in *Seventh ACIS Int. Conf. on Software Eng. Research, Manage. and Applicat.*, 2009, pp. 261–266.

[14] M. H. Cancian, J. Carlo, R. Hauck, C. G. Von Wangenheim, and R. J. Rabelo. "Discovering Software Process and Product Quality Criteria in Software as a Service." in *Product-Focused Software Process Improvement*. Springer Berlin Heidelberg, 2010, pp. 234–247.

[15] S. K. Garg, S. Versteeg, and R. Buyya. (2013, Jun.). "A framework for ranking of cloud computing services." *Future Generation Computer Systems*, 29(4), pp. 1012–1023.

[16] S. K. Garg, S. Versteeg, and R. Buyya, "SMICloud: A Framework for Comparing and Ranking Cloud Services," in *Fourth IEEE Int. Conf. Utility and Cloud Computing*, 2011, pp. 210–218.

[17] S. Wang, Z. Liu, Q. Sun, H. Zou, and F. Yang.(2012, May). "Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing." *J. of Int. Manufacturing*.

[18] P. X. Wen, and Li Dong. "Quality Model for Evaluating SaaS Service," in *Fourth Int. Conf. on Emerging Intelligent Data and Web Technologies (EIDWT)*, 2013, pp. 83-87.

[19] N. Baliyan, and S. Kumar. "Quality Assessment of Software as a Service on Cloud Using Fuzzy Logic," in *IEEE Second Int. Conf. Cloud Computing in Emerging Markets*, 2013, pp. 1-6.

[20] S. Venkatraman and B. Wadhwa. (2012, Sep.). "Cloud Computing: A Research Roadmap in Coalescence with Software Engineering." *Software Engineering: An Int. J. (SEIJ)*. 2(2), pp. 7-18 .

[21] S. Stuckenberg and A. Heinzl. "The Impact of the Software-as-a-Service Concept on the Underlying Software and Service Development Processes," in *Proc. PACIS*, 2010.