

# Requirements Engineering Process for Software-as-a-Service (SaaS) Cloud Environment

Anum Tariq<sup>1</sup>, Dr. Shoab Ahmed Khan<sup>2</sup>, and Sundas Iftikhar<sup>3</sup>

Department of Computer Engineering, College of Electrical and Mechanical Engineering (CEME)

National University of Science and Technology (NUST), Islamabad, Pakistan

[anum\\_tariq11@hotmail.com](mailto:anum_tariq11@hotmail.com), [kshoab@yahoo.com](mailto:kshoab@yahoo.com), [sundas\\_iftikhar@ymail.com](mailto:sundas_iftikhar@ymail.com)

**Abstract**— This paper deals with the important topic of Requirements Engineering in cloud computing, specifically on Software as a Services (SaaS) cloud model and presents a new methodology for Requirements Engineering of SaaS by identifying two key questions to handle Requirements in SaaS. The paper evolves by considering a CMMI modification by adding a new element in such a process, specifically devoted to SaaS and proposed a modification in the traditional Requirements Engineering process. The paper also identifies a set of requirement categories that must be considered in SaaS, which may not always be considered in the development process of traditional software.

**Keywords**— Software Requirements Engineering; Software as a Service (SaaS); Cloud Environment, CMMI, Cloud Assessment.

## I. INTRODUCTION

Software industry is progressively considering cloud computing to increase working competency and grasp most of the market as cloud computing is the demand of industry due to its major features like On-demand self-service, rapid elasticity, Ubiquitous network access, measured services and Resource pooling [1]. There are many forms of Cloud computing like infrastructure as a service (IaaS), platform as a service (PaaS), and Software as a service (SaaS). SaaS is indicated as the novel trend in the delivery of application software. This model of distribution visualizes an interest headed software market where organizations gather and deliver services as and when needed to address a specific requirement. SaaS concentrates on dividing the ownership and responsibility of software from its usage. Many existing restrictions in software use, distribution, and development can be reduced by providing software's functionality as services that can be configured at delivery time. Such a model of software delivery has exposed fresh markets for both the small-scale services providers and larger organizations [2] [3].

The increasing interest in the upswing of SaaS is because SaaS service provider installs, runs and maintains all the required software and hardware. Service users only access it through internet. They don't need to worry about installation maintenance of the software being used like traditional software users [4]. A large number of organizations consider SaaS important [5]. The survey [5] concluded that 15% organizations consider SaaS IMPORTANT and 5% of the organizations consider it VERY IMPORTANT. Research statistics [6] describe the comparison and analysis of costs spent on purchasing and placement of in-house software and software used as a service. The statistics show that hiring a software saved

45% of the customer's savings and cut out 63% of its IT expenses. The major attraction towards software rental is if one can rent a service then why buy it? Cloud Computing and Software as a Service (SaaS) models are changing the way in which software is designed, developed, tested, delivered and used. This is a time when the results and potential expenses of errors and faults are climbing quickly for organizations that handle cloud services. Software experts must create better methods for creating and keeping up practices of cloud applications. SaaS needs to be as Research-Directed and informed software development as traditional software development is and more organized and well-structured concerning Software Engineering processes, Requirements, Development and Design procedures [7].

The difference between traditional software applications and SaaS applications gives a rise to a different direction of requirements in the sense of operations and management, architecture, security and privacy, compliance and quality. Altogether this implies that new requirements that have never been experienced before in software evolution have to be tackled i.e. scalability, SLA compliance, multi-tenancy support, security and usage monitoring etc. The principle goal of requirements Engineering process is to develop and as well as manage the requirements and to identify changes and irregularities between the project outcomes and the requirements [8]. If you want your SaaS to be better in quality you have to focus on these key requirements to be considered and discussed at early stages, when requirements engineering team struggle to accomplish improved quality, shorter delivery deadlines and reduced requirements engineering costs etc. [9]. Survey [9] results also show that with improved requirements engineering, debugging cost cut down can be increased by 18% and debugging rate may be compressed by a margin of 44%. Insufficient gathering of requirements may result in design changes at later development stages. These design changes may result in additional requirements management tasks because changing in design of SaaS applications is a challenging job. With the coherent participation of cloud service providers, the software development complexity is to climb further [10]. Therefore, in the ensuing cloud era, existing software process models are simply insufficient unless the remote interaction with cloud providers is a part and parcel of the whole process. This qualitative study focuses on the software process concerns over technological concerns, addressing the two basic research questions related to Requirements Engineering of SaaS cloud applications.

**RQ1.** *What are the specific requirements that should get priority while requirements Engineering of SaaS?*

**RQ2.** *How should we change the existing user-based requirements engineering practices to appropriate service-based requirements engineering?*

## II. RELATED WORK

Many researchers have been working in the field of cloud computing and requirements engineering from the last decade. A few of them have also worked on the software process improvement topics but they have not discussed any framework designed specifically for SaaS requirements engineering process. Major Software engineering challenges in cloud environment include: Software Composition, Query Oriented vs. API oriented programming, Availability of source code, Execution Model and Application Management [11]. These challenges need to be addressed in various Software Engineering processes and methodologies so as to make the cloud concepts more beneficial to all sections of the world.

Cloud computing being the most up to date buildup of the IT business, the challenges of Software Engineering on the Cloud computing platform have not been examined yet and no service improvement process model for Cloud computing platform has been recommended yet [9]. In the study [9] extreme programming methodology is extended for cloud environment and claimed that each stage of the cycle should also include cloud provider. Software Engineering Frameworks for the Cloud Computing Paradigm [13] is a complete book in which concepts related to Software Engineering procedures and practices that best suit cloud model are focused. The aim of the book is to present current research and development in the field of software engineering as relevant to the cloud paradigm. Various studies regarding Impact of Cloud Paradigm on Software Engineering, Software Development Life Cycle for Cloud Platform, Software Design Strategies for Cloud Adoption and Performance of Cloud Based Software Applications are included in the book showing a fertile area of research. Emphasis on enterprise requirements of cloud environment have been made in a study [12] in the book and the framework proposed discusses enterprise requirements in context of business oriented analysis. Three categories of enterprise requirements as mentioned by the authors in [12] are customer Requirements, market analysis and investment analysis. Their research question was “how do we change our existing user-based requirements capturing methodologies to a suitable service based business requirements engineering?”

The development of cloud based applications as SaaS implies an additional task that need to be planned as part of requirement analysis [14]. This additional task is performing Cloud assessment at initial level. The assessment also helps to determine the business case and return on investment. Typical assessment questions listed in the study [14] are: Does cloud architecture fit the requirements for the application? How interconnected is this application with other application in the enterprise—for public cloud, can these interfaces be exposed for access from external networks? Is the enterprise comfortable with public cloud, or should it focus only on private cloud

among other options? and How would applications be monitored after they are hosted on public cloud? This study [14] was based on traditional requirements management and addressed the issue in context of business and return on investment. No technical aspect have been discussed in the study.

## III. PROBLEM STATEMENT AND MOTIVATION

The problem is that while dealing with cloud applications where you never know where the data is, who is using it and from where he/she is using it; software processes get more and more complex to adopt. Even requirements Engineering process becomes so complex. The major concern for a SaaS application is security than compliance requirements, monitoring billing and usage of services and so many other special types of requirements. Cloud separates the data and the high level operational features of the application from the infrastructure and hides the low level technical details, like how is the data replicated and where is it stored etc., hence differentiating it from the old style traditional outsourcing which is still quite standalone computing. These contrasts offer climb to an exclusive set of concerns that not just effect the security, risk management practices and privacy, yet have additionally empowered a new rise of legal issues in areas of consistence, compliance and examining and auditing. SaaS vendors need to address a significant number of non-functional application concerns that are essential for the success of the service. For example, traditional software vendors were not concerned with issues like metadata management, multitenancy, tenant customization and configuration, scalability, fault tolerance to meet SLAs, metering, monitoring, robust security in distributed environments, and a host of other concerns. The key questions are: “What are the specific requirements that should get priority while requirements Engineering of SaaS?” and “How do we change our existing user-based requirements engineering practices to an appropriate service-based requirements engineering?”. It is an interesting problem area to work in because cloud computing is an emerging field and attracting market due to its large benefits. From the literature we have not found any study that addresses this issue for SaaS or has proposed a framework of requirements engineering for SaaS cloud environment.

## IV. SAAS REQUIREMENTS ENGINEERING PROCESS

Software engineers need to develop SaaS applications that backup the multitenant architecture which support sharing single instance of the service/application among multiple users. Software processes must be applied in order to get better results in the form of quality of the products, more customer satisfaction and trust leading to more revenues. The main idea of this paper is to study the traditional user based requirements engineering process and practices and transform it into a services based requirements engineering process for SaaS software applications. Our first research question details out the specific requirements of SaaS applications and their importance for SaaS. The second research question focusses on the practices for SaaS requirements engineering and quality of the process. As a solution to answer the second research question we have

proposed a modification in the CMMI Level 2 processes that may result in a well-structured and research directed development of SaaS applications.

#### A. SaaS Requirements: Answer to RQ1

There are three major stakeholders that must be involved in the requirements engineering process as shown in Figure 1. So requirements related to all of these users of Cloud Services have been explored and discussed. The requirements are mostly non-functional in nature.

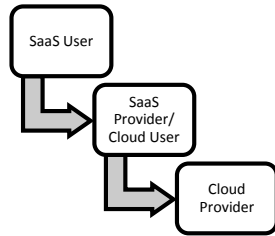


Figure 1. SaaS users

Major categories include Architecture Requirements, Operations/ Behavioral Requirements, Management Requirements, Technical/ Implementation Requirements, Quality of Services (QoS), Security and Privacy Requirements and Compliance Requirements. Although requirements under these categories may overlap. Figure 2 explains the relationship among various SaaS stakeholders and the requirements for those stakeholders.

**Architecture Requirements:** Architecture is very important for any application. As SaaS customers are more concerned with what services are being provided rather than how the service will be provided so the cloud provider can choose from the various cloud model options based on the cost and performance. Multitenancy adds complexity in data restoring and upgrades so the architecture should permit restoring data of one user without effecting other’s data and minimum interruptions during upgrades. The suitable Provider delivery model (SaaS, PaaS, IaaS) and tenancy model choice will depend on customer requirements such as security, privacy, high availability and customizability. Varying levels of service agreements (SLA) are also part of SaaS architecture requirements. SaaS architecture should be stateless as it can provide the benefits of lowest cost, best reliability and highest performance. The architecture must have features like fault tolerant, comprehensive redundancy and uptime and fail over strategies. Other architectural requirements for SaaS may also include details about the reusable SaaS services, integration with legacy elements, API requirements, Server side requirements and multiple, secure, disaster tolerant data centers.

**Operations/ Behavioural Requirements:** Easy customization and extension of user interfaces, data and business processes, User experiences, Design requirements, Security and Privacy of functionality or transactions, Availability, Performance, Interoperability, frequent and non-disruptive upgrades, Capacity planning, Data migration, Data locality management and Data governance are the key operational issues for SaaS applications.

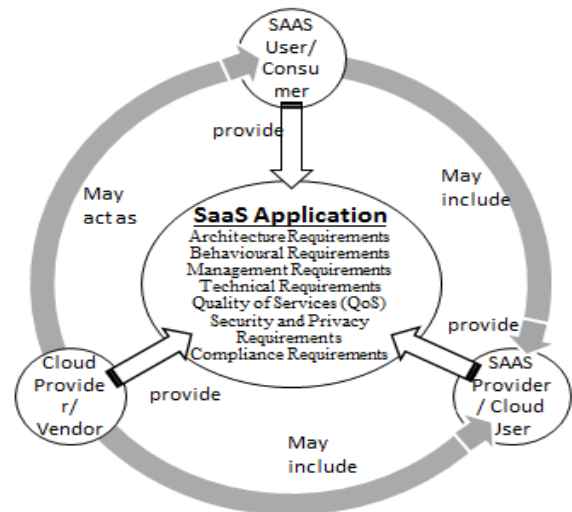


Figure 2. SaaS users and requirements relationship diagram

**Management Requirements:** Management requirements may include Centralized reporting and monitoring of service levels, usage and billing, hosting/ user and tenant management, monitoring and auditing tools Reporting, SLA management, Capacity Planning, Provisioning, Monitoring, Data management, storage and processing, Virtualization management, Load balancing, systems management, bug fixes and upgrades with minimal disruption in service, Patch management and Process management.

**Technical/ Implementation Requirements:** Recruitment, role changes, Customer/ SLA/ Technical Support, Migration of data, Cloudonomics, Service centric issues, cost of operating SaaS solution, delivering services, upkeep and maintenance, administering and upgrades, Administrative cost, Quality assurance and control/ comprehensive test strategy, Third party engagement, Transferable skills, Better testing (penetration testing, “white hat hacking”) can be considered as technical requirements for SaaS.

**Quality of Services (QoS):** Uptime requirements for high availability and fault tolerance, Redundancy (track record), Performance, Interoperability, Scalability, High availability, Reliability (restoring data), Adoptability and Usability requirements (Easy to procure, install, integrate, use, update and configure) should be focused so as to increase quality.

**Security and Privacy Requirements:** One of the major concerns for cloud based applications is the security and privacy of user data. Customer acceptance and adaptation/ trust, Audited data security controls, Device and user authentication and access control, Code reinforcing and reviewing, Malware detection, Rights to audit the SaaS vendor’s operations and access log info, Physical and personal security are the requirements that must be handled in order to deal with security and privacy issues. Furthermore, multi-tenancy require additional security and privacy like authentication and

authorization; cross domain authentication, user provisioning and identity administration; managing accounts etc.

**Compliance Requirements:** Service related protocols, Regulatory compliance, Licensing requirements (service level management and compliance SLA's), Compliance focus (Industry specific, Geopolitical and Enterprise specific compliance requirements) should be addressed to meet application compliance.

SaaS requirements engineering process should include detailed information about all these requirements [14]. Detailed requirements under these categories may include availability of applications at regular intervals, the strategy for application downtime and uptime, limits of data loss in case of application crash, I/O volume required to satisfy the business process, level of concurrency required across time periods and hence amount of parallelism that can be applied to improve the performance of the system. Furthermore, requirements on identifying number of concurrent users accessing the system, volume of data that is required to process the functionality, network bandwidth and expected delay in response while processing heavy load functionality, Identify and analyze country-level regulations to handle the load and different failure scenarios and respective handling mechanisms, and Cost of maintaining application on cloud should also be considered as important during SaaS requirements capturing and analysis process as these requirements directly have impact on the services provided by the SaaS cloud [15] [16] [17] [18] [19] [20].

Requirements under all these categories come from three different dimensions of users as shown in Figure 2. For example under the most important category of requirements that is security and privacy requirements, SaaS user/ consumer will provide requirements like customer acceptance and adaptation/trust and device authentication. SaaS provider/ cloud vendor will provide requirements such as audited data security controls, user authentication and access control, malware detection, protection of access by other tenants collocated in the same system and rights to audit the SaaS vendor's operations and access log info. Cloud provider/ vendor will provide requirements related to privacy even in multitenant environment, cross domain authentication, identity administration and additional security required by Multitenancy.

### B. Enhanced CMMI Level 2 Processes: Answer to RQ2

The biggest challenge in Cloud Computing is the lack of a standard that has been designed specifically for SaaS. For any organization transforming from a standalone business style into a collaborative business trend by reusing and sharing processes and services with the help of cloud, it is very difficult to understand the clear requirements and the relationship between those requirements [21].

Traditional software Engineering process include activities like Elicitation, Analysis, Specification and verification during requirements development phase and traceability and change management while management phase [22]. At each phase

software engineer and the user is involved for requirements capturing, analysis, documentation and even during requirements verification. Requirements engineering until now involved end users, and software engineers. But for SaaS it has to involve cloud service providers as well, as they will be providing the computing infrastructure, software development, management, maintenance platforms, etc.

As the cloud providers are only conversant with the infrastructure utilization details, their experts can do the capacity planning, risk management, configuration management, quality assurance, etc., well. Similarly, Requirements management activities should also include Cloud Service Providers, who can chip in with some decision-enabling details such as software-development cost, schedule, resource, and time. Figure 3 shows the enhanced Requirements Engineering Lifecycle for SaaS cloud based applications where Cloud vendor have been involved at each stage.

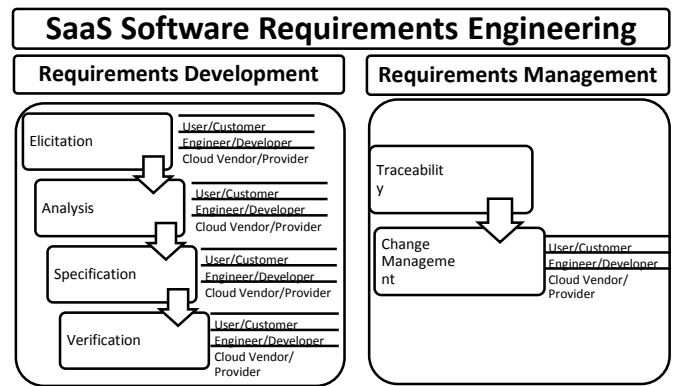


Figure 3. SaaS Requirements Engineering Framework

A report [23] showed that using a standard like CMMI for process improvement can help in reducing different costs and improvement of the process as shown in Table 1.

TABLE I. SOFTWARE PROCESS IMPROVEMENT USING CMMI STATISTICS [23]

Improvement attributes	Improvement rate
Development cost decrease	73%
After delivery deficiency rate cut down	80%
Rework cost decrease	96%
Reduce project risk	92%
Shorten average development cycle time	37%
Increase return on investment	21 times

Considering CMMI as a criterion for improvement of the procedures involved in the evolution of the software applications, CMMI level 2 has the five basic sub processes for the REQM as shown in Figure 4.

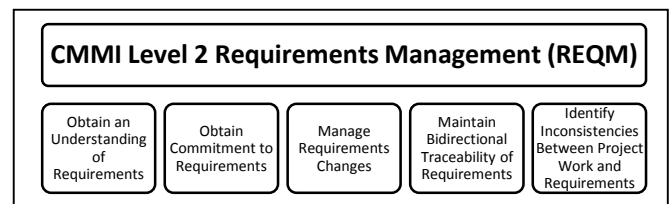


Figure 4. CMMI level 2 REQM Framework [24]

From CMMI Process Evaluation Checklist [24] we have extracted Key Process Areas (KPA's) of CMMI level 2 that have to be considered to improve the software process for any software and particularly first KPA that is Requirements Management (REQM) as it is the main focus of this study. Table 2 shows the sub processes of REQM and the activities in each sub process. The development of cloud based applications as SAAS implies an additional task that need to be planned as part of requirement analysis. This additional task is performing Cloud assessment at initial level. Cloud assessment will help to evaluate the cloud readiness and applicability for an enterprise. The assessment also helps to determine many requirements for the services being built on that cloud.

TABLE II. CMMI LEVEL 2 REQM CHECKLIST [25]

CMMI L2 Process Evaluation Checklist for REQM	
Requirements Management (REQM)	Obtain an Understanding of Requirements
	Document the requirements
	Identify the stakeholders
	Distribute the requirements for review
	Allow time for adequate review
	Encourage feedback
	Obtain Commitment to Requirements
	Identify appropriate approver groups
	Incorporate feedback
	Set a time limit
	Ensure that commitment allows for future change
	Seek signatures
	Manage Requirements Changes
	Know that requirements will change
	Control with baselines
	Honour your customers' needs
	Assess proposed changes
	Incorporate changes in an orderly manner
	Maintain Bidirectional Traceability of Requirements
	Trace to plan
	Trace to anticipate
	Trace to know
	Identify Inconsistencies Between Project Work and Requirements
	Harmony with plans
	Harmony with work products

Reporting, SLA management, Capacity Planning, Billing and Metering, Provisioning, Monitoring, Data management, storage and processing, Virtualization management and Load balancing requirements and issues must also be assessed at the stage of requirements engineering in the newly added subprocess of cloud assessment. Figure 5 shows the enhanced CMMI level 2 REQM Framework for SaaS applications with the modification of adding a new element.

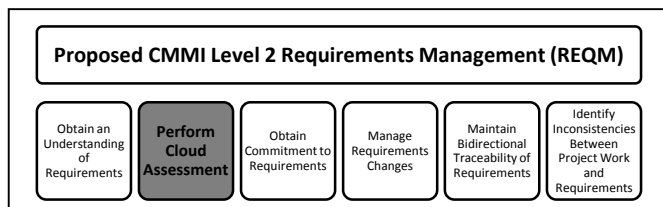


Figure 5. Proposed CMMI level 2 REQM Framework for SaaS

**Cloud Assessment:** It is important to note that cloud assessment guidelines are defined at enterprise level. The enterprise can optionally create tools to aid the projects and new initiatives to perform cloud assessment. There are much more requirements that need to be planned at Requirements

Engineering process for better quality and avoid problems at later stages of SaaS development.

TABLE III. ENHANCED CMMI LEVEL 2 REQM CHECKLIST

Activities under the newly added sub process in CMMI Level 2 Process Evaluation Checklist for Requirements Management (REQM)	
Perform Cloud Assessment	Assess cloud architecture for initial requirements identified
	Make decision for public or private cloud
	Identify suitable cloud service provider
	Assess the cost of using cloud
	Document the cloud requirements

In the newly added subprocess of CMMI level 2 five activities has been added as shown in Table 3.

- *Assess cloud architecture for initial requirements identified:* Does cloud architecture fit the requirements for the application? Either the SaaS under development can be deployed on the cloud infrastructure available in case of already available cloud? If a new cloud setup has to be build than this activity may assess the requirements for the cloud i.e storage capacity, processing speed, security levels etc.
- *Make decision for public or private cloud:* Is the enterprise comfortable with public cloud, or should the enterprise focus only on private cloud option among other options? According to the SaaS requirements identified initially one has to decide either to go for private cloud or use public cloud. If the application is not security or privacy critical or client has not explicitly mentioned the requirements for using cloud model than one can use a public cloud from an appropriate cloud vendor. If the application is subject to the critical security and privacy issues than deploying a private cloud can be a better option. Defining the strategy in adopting cloud for future project.
- *Identify suitable cloud service provider:* How the cloud vendor plan to restore data for one single customer without impacting others? How do cloud vendor provide varying levels of services (SLA's)?
- *Assess the cost of using cloud:* Cloudonomics, work on the economics and financial considerations of Cloud.
- *Document the cloud requirements:* What type of physical and personal security cloud provide? How would applications be monitored after they are hosted on public cloud? How much scalable the service will be? Document all requirements.

## V. CONCLUSION AND FUTURE WORK

Clear understanding of requirements and their importance for SaaS cloud environment is a critical for any enterprise system transforming the user centered requirements engineering process into a service centered engineering process. This paper presents the requirements for SaaS applications in detail and the major categories identified include Architecture Requirements, Operations/ Behavioral Requirements, Management

Requirements, Technical/ Implementation Requirements, Quality of Services (QoS), Security and Privacy Requirements and Compliance Requirements. We investigated the characteristics of Cloud Computing and SaaS software applications and characterized these requirements accordingly. These requirements may overlap in terms of categories and users of SaaS. Importance of including cloud vendors during the requirements engineering activities has been also a focus of discussion in the paper.

One of the major challenges of Cloud Computing is the absence of a standard or a particular procedure, which can meet the organization's cloud requirements. A new sub process has been suggested as part of the CMMI level 2 Requirements Management (REQM) KPA that is cloud assessment. Further the activities under the newly added sub process have been identified and explained.

In future we will be working on a case study project named Allwebid [26] and map the suggested process on a real project to check the effectiveness of the findings. We will map the REQM KPA of CMMI level 2 for Allwebid and try to fill up the gaps in the Requirements Engineering process to improve the quality of the case study project. In future we will also explore the other KPA's of CMMI level 2 for SaaS cloud environment and try to find out other SAAS focused activities rather than Requirements Engineering.

#### REFERENCES

- [1] T. W. E. S. Bernd Grobauer, "Towards a Cloud-specific Risk Analysis Framework," in *Siemens IT Solutions and Services Monograph*, 2010.
- [2] D. B. P. B. Mark Turner, "Turning Software into a Service," in *IEEE*, 2003.
- [3] A. A. K. Anum Tariq, "Framework supporting team and project activities in Global Software Development (GSD)," in *IEEE ICET*, 2012.
- [4] V. Choudhary, "Software as a Service: Implications for Investment in Software Development," in *40th Hawaii International Conference on System Sciences*, 2007.
- [5] R. Z. W. X. W. Q. A. Z. Minqi Zhou, "Security and Privacy in Cloud Computing: A Survey," in *Sixth International Conference on Semantics, Knowledge and Grids*, 2010.
- [6] "Software as a Service: Strategic Backgrounder," Software & Information Industry Association, 2001.
- [7] L. O.-E. Arrieta, "From Software as a Good to Software as a Service: Preparing the Evolution of Software Products into the Cloud," in *IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, 2012.
- [8] J. J. a. Y.-S. Lin, "Research and Development of a CMMI-Complaint Requirements Management System for Software Engineering," in *Springer-Verlag Berlin Heidelberg*, 2008.
- [9] R. Guha, "Impact of Semantic Web and Cloud Computing Platform on Software Engineering," *Software Engineering Frameworks for the Cloud Computing Paradigm (Springer)*, pp. 3-24, 2013.
- [10] B. V. D. T. B. Chhabra, "Software engineering issues from the cloud application perspective," *Int. J. Inf. Technol. Knowl. Manage.*, vol. 2, no. 2, p. 669-673, 2010.
- [11] V. V. a. R. A. Pethuru Raj, "Envisioning the Cloud-Induced Transformations in the Software Engineering Discipline," *Software Engineering Frameworks for the Cloud Computing Paradigm (Springer)*, pp. 25-54, 2013.
- [12] M. Ramachandran, "Business Requirements Engineering for Developing Cloud Computing Services," *Software Engineering Frameworks for the Cloud Computing Paradigm (Springer)*, pp. 123-144, 2013.
- [13] S. S. Zaigham Mahmood, *Software Engineering Frameworks for the Cloud Computing Paradigm*, London: Springer, 2013.
- [14] R. K. a. R. Jayakrishnan, "Impact of Cloud Services on Software Development Life Cycle," *Software Engineering Frameworks for the Cloud Computing Paradigm, Computer Communications and Networks*, Springer, 2013.
- [15] M. Thanawala, "Oracle SaaS Platform: Building On-Demand Applications," in *An Oracle White Paper*, 2008.
- [16] I. SoftServe, "Building Successful Enterprise SaaS Apps for the Cloud; How to Develop Powerful, 'On-Demand' Software Solutions In An Increasingly Competitive & Mobile Marketplace," THINKstrategies, Inc., 2011.
- [17] D. Key, "(done) Software Development Strategies for Building SaaS Products.Cloud Strategies - Making SaaS Businesses Work," Cloud Strategies, 2013. [Online]. Available: <http://www.cloudstrategies.biz/why-software-development-for-saas-is-different/>. [Accessed 29 3 2014].
- [18] S. V. TILL, "7 Requirements for SaaS," SecurityInfoWatch, 11 2 2011. [Online]. Available: <http://www.securityinfowatch.com/article/10537058/7-requirements-for-saas>. [Accessed 29 3 2014].
- [19] D. Key, "Software development for SaaS," Cloud Strategies, 2013. [Online]. Available: <http://www.cloudstrategies.biz/practice-areas/product-development/>. [Accessed 29 3 2014].
- [20] D. Key, "Requirements for building enterprise saas applications," cloudstrategies, 2013. [Online]. Available: <http://www.cloudstrategies.biz/requirements-for-building-enterprise-saas-applications/>. [Accessed 29 3 2014].
- [21] A. J. D. K. Y. G. Bhaskar Prasad Rimal, "Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach," *J Grid Computing, Springer*, 2011.
- [22] K. E. Wieggers, "When Telepathy Won't Do: Requirements Engineering Key Practices," *Cutter IT Journal*, pp. 1-8, May 2000.
- [23] "US Department of Defence: A business case for Software Process Improvement revised," in *State-of-the-art report*, 2000.
- [24] D. D. F. Rico, "CMMI Process Evaluation Checklist," [Online]. Available: <http://davidfrico.com/s-cmmi-checklist.pdf>. [Accessed 5 March 2014].
- [25] J. R. Persse, "Goals and Practices," in *Requirements Management Under CMMI*, O'Reilly Media, 2007, pp. 11-31.
- [26] "Allwebid," Allweb Technologies, Inc., [Online]. Available: <http://www.allwebid.com/>. [Accessed 25 March 2014].