Cloud Computing Development Life Cycle Model (CCDLC)

Muhammad Babar(✉), Ata ur Rahman, and Fahim Arif

National University of Sciences and Technology, Islamabad, Pakistan
babarkhan66@gmail.com, atabcs_87@yahoo.com,
fahim@mcs.edu.pk

Abstract. Cloud Computing is getting reputation as the standard approach for designing and organizing software applications over the internet, especially for distributed and e-commerce applications. In recent times, Cloud Computing has emerged as a new opportunity that how software and other resources can be provided to the consumers as a service. However, applying the existing traditional software engineering life cycle models to cloud computing, we identify some inadequacies like they do not concentrate on engineering activities, they lack the fundamental description of cloud services using traditional requirement engineering process, they do not deal with proper modeling, and they suffer from good development and testing processes. We propose a cloud computing life cycle model (CCDLC) for development of cloud to deal with the mentioned deficiencies faced in the existing traditional life cycle models.

Keywords: Cloud computing · Software engineering · Life cycle model · Process model

1 Introduction

The need to access data from any location transformed World Wide Web into an intelligent web which can display quarried data from any location. As the time pass the technologies have improved and need increased day by day. So the experts feel that there is need of some more improved version of business oriented model where hardware, software, tools and applications provided to customer on lease, which can be accessible across the globe via internet. Now this idea of experts and companies implemented through cloud computing where users can get hardware software, tools and applications on lease. Cloud computing has brought the innovative period of potential computing, change a huge branch of IT trade, redesign the buy and use, and get significant consideration worldwide and local society of IT, national and intercontinental organizations [3, 5].

Cloud computing benefits software engineering concepts like agility, availability and cost efficiency. These need to be well engineered for cloud platforms using the software engineering methodologies particularly modeling the cloud aspects to provide logical tested solution prior to implementation to improve the quality. The models assist in representing the problem and its solution (logically) in a methodical mode. They also demonstrate the precise details for different viewpoints and at various stages of development [2]. The goals of modeling and design of cloud paradigm, are to

J. Ferreira and M. Alam (Eds.): Future 5V 2016, LNICST 185, pp. 189–195, 2017.
DOI: 10.1007/978-3-319-51207-5_19

support all the activities of cloud modeling and design to fit into an overall model-based development from both business and IT perspectives. Unfortunately, there is not yet a consensus on the right set of process models.

The rest of the paper is organized as follow: Sect. 2 illustrates the detail background with regard to the development of the cloud, Sect. 3 introduces the proposed cloud computing development life cycle model, and finally Sect. 4 includes the conclusion.

2 Motivation and Background

Software models have been directing the practice in the domain of software engineering. The process model is actually a set of logically associated activities whose implementation guides to the construction of a specific software product [6]. There are frequently precise specifications and in order to obtain them there may be need to combine different cloud services [6].

Generally, we can have two possible ways to utilize the software engineering in the paradigm of cloud computing; (i) to improve the software development methods to suite designing software applications for the cloud (ii) the utilization of cloud to support the process of software development [1]. Cloud computing benefits software engineering concepts like agility, availability and cost efficiency. These need to be well engineered for cloud platforms using the software engineering methodologies particularly modeling the cloud aspects to provide logical tested solution prior to implementation to improve the quality. Modeling of cloud computing can be used to generate code and offer code generation as a service which is the attractive and demanding activity [4].

There is a huge anxiety that what type of service and requirements of the organization should be moved to the cloud and this migration may take place based on the QoS [8, 9]. To identify and categorize the services of being provided by the cloud, the services should be ranked [11]. The scalability of the cloud should also be kept in mind and should be a very obvious part of the cloud development process because the nodes which are part of the cloud may be located in different regions of the world [10]. The Service Level Agreement (SLA) is also a major activity involved in the cloud computing paradigm, thus a proper management of the SLA is also important [12].

The cloud development is a very sensitive issue in terms of its flexible nature. The development of cloud should be taken into consideration throughout the entire life cycle f the cloud development. Global Software Development could be one of the ideal techniques to utilize for the development of cloud computing [13, 14]. Risk mitigation, monitoring and management activity could play a vital role in the development of cloud.

3 Cloud Computing Development Life Cycle Model (CCDLC)

This section includes the detail description of the proposed life cycle model. Software development model can be different for cloud computing than traditional development because cloud need to consider services to internal consumptions as well as external

users. The target market of developing cloud application is small and most of them are not fully aware of the development life cycle on cloud computing. The services to external users the development life cycle can be different because it is integrated with internal consumption of cloud service.

There is need to integrate other significant processes with SDLC, because during cloud development there is need of other processes such as feasibility analysis, risk management, security and privacy checks, scalability, efficiency, SLA and other QoS factors. Therefore, we propose, a specific life cycle model for the cloud development which tackle the above mentioned characteristics of the cloud computing. The abstract description of the proposed life cycle model is depicted in Fig. 1. The figure highlights and visualizes the overall proposed model and depicts all the phases which are proposed as part of the proposed model. The detailed description of each and every phases of the proposed cloud computing life cycle model (CCDLC) is given below:

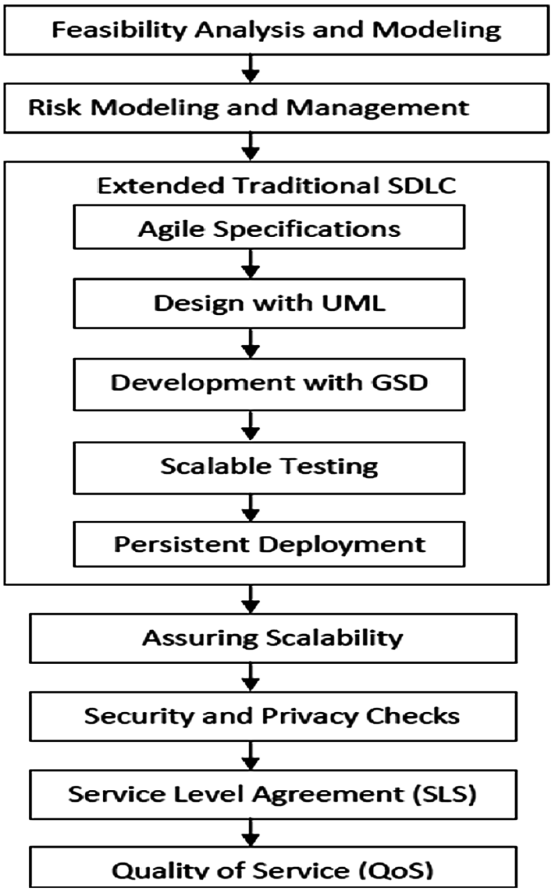

Fig. 1. Proposed cloud computing development life cycle model (CCDLC)

3.1 Feasibility Analysis and Modeling

This phase is about the decision making by the organization to migrate cloud paradigm. Now a days the cloud computing has become the most demanding paradigm and every organization is moving to the cloud. There are a number of reasons for the reputation such as extreme scalability, accessibility and the reduction in the IT cost. However, several organizations still get it tough to migrate their application to the cloud. The complexity of migrating to cloud could be because of different reasons. Therefore, the first step we propose in the proposed model is to clearly take a clear decision to migrate to the cloud. The decision should be based on obvious advantages and pros.

3.2 Risk Modeling and Management

Once the decision is made to migrate to the cloud by organization, and then the risk modeling and its mitigation and management should be taken into account. Risk is the opportunity that an incident will happen and unfavorably influence the results. This phase plays an important role in the overall designing of the cloud. Cloud may affect by a number of risks such as security risks, privacy issue, technical risks, social risks, and accessibility issues. Therefore, these issues are needed to be identified and mitigated properly before going to develop cloud.

3.3 Extended Traditional Software Development Life Cycle (SDLC)

The traditional SDLC include requirement, design, development, validation, and deployment. In the cloud development, these phases are taken into account with some modifications and extensions. The detail description of these phases is given below:

3.3.1 Agile Requirement Engineering

The traditional requirement engineering process can be extended in the way to continuously manage the requirements with the agility. The requirement phase includes the identification and elicitation of the cloud services; it may include the subscription to an existing service in the cloud, developing in-house service using the cloud environment, or outsourcing service.

3.3.2 Modeling and Design Using UML Extension Mechanism

Architecture and design is the next phase of traditional generic SDLC. In this phase the proper architecture (the logical solution) of the system is built and mapped to the requirement. Unified Modeling Language (UML) can be extended to model the cloud aspects in the overall process of the cloud development.

3.3.3 Development with GSD Techniques

The next phase of traditional SDLC is the development. The cloud development is a very challenging and complex task. The traditional SDLC development phase may extend to Global Software Development (GSD) to achieve the required services according the specifications. Now days, we are practicing an explosion of cloud

computing as a novel and innovative generation of the internet. Every organization is migrating to the cloud. Therefore, the development of the applications should be taken into consideration with the experience and skills of international experts using the GSD techniques.

3.3.4 Scalable Testing

The next phase of traditional SDLC is the testing. In this phase the developed systems is tested, verified and validated thoroughly. As the cloud is a complex distributed system, therefore, the traditional tasting techniques will be extended to have scalable testing methods and tools to validate and test the cloud services. The scalable testing may include automated testing procedure and automated tools to test the services automatically.

3.3.5 Deployment with Persistency

Deployment is the very last phase of traditional SDLC. Traditional deployment is one big bang activity which is taken place at the end of the development. Cloud is a complex distributed system, therefore, it requires a proper continues deployment process in the form of persistent deployment. Persistent deployment is a kind of extension to continue integration whose purpose is to minimize the load time between the development and deployment. To get the persistent deployment the development squad relies on infrastructure that mechanize and appliance a variety of steps leading up to deployment.

3.4 Assuring Scalability

The capability to scale on demand is the greatest benefit of cloud computing. When taking into consideration the variety of advantages of cloud, scaling on-demand is one of the all time great advantage in the cloud computing paradigm. Thus, after the development of the cloud, the cloud should assure the scalability and it should be scalable to provide the services on demand at any stage. This scaling facility should be auto by the vendor.

3.5 Security and Privacy Checks

Security and privacy have consistently been a big issue in IT applications. In the cloud computing paradigm, it becomes principally severe because the services situated in diverse locations across the globe. Security and privacy are the two major features of user's anxiety in the cloud computing. Therefore, the security and privacy should be handled properly after the development and deployment of the cloud.

3.6 Service Level Agreement (SLA)

Security and privacy have consistently been a big issue in IT applications. In the cloud computing paradigm, it becomes principally severe because the services situated in

diverse locations across the globe. Security and privacy are the two major features of user's anxiety in the cloud computing. Therefore, the security and privacy should be handled properly after the development and deployment of the cloud.

3.7 Quality of Service (QoS) Management

Due to the numerous use of cloud computing, the QoS of cloud computing has turn out to be a significant and necessary matter as there are numerous open issues which required to be resolved associated to the trust in cloud computing paradigm. QoS management involves guaranteeing the level of service along with all its attributes such as availability, performance and reliability. In this phase proper QoS modeling approaches are needed to be used assure and manage the QoS.

4 Conclusion

Cloud computing development life cycle (CCDLC) model is proposed for the designing and developing of cloud from both vendor and consumer perspectives. The proposed model overcomes the shortcomings and deficiencies found in the traditional existing software engineering process models. The proposed process model includes the modifications and extension of the traditional SDLC and other important processes.

References

1. Whittle, J., et al.: The state of practice in model-driven engineering. IEEE Softw. **31**(3), 79–85 (2014)
2. Brunelière, H., et al.: Combining model-driven engineering and cloud computing. In: 4th Workshop on Modeling, Design, and Analysis for the Service Cloud, Paris, France, June 2010
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
4. Crocombe, R., Kolovos, D.: Code generation as a service. In: Proceedings of ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, Ottawa, Canada, 29 September 2015
5. Silva, G.C., Rose, L.M., Calinescu, R.: Cloud DSL: a language for supporting cloud portability by describing cloud entities. In: Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, 30 September 2014
6. Sommerville, I.: Software Engineering, 9th edn. Addison-Wesley, Boston (2011)
7. Patidar, S., Rane, D., Jain, P.: Challenges of software development on cloud platform. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 1009–1013 (2011)

8. Saripalli, P., Pingali, G.: MADMAC: multiple attribute decision methodology for adoption of clouds. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), 4–9 July 2011, pp. 316–323 (2011)
9. Beserra, P.V., Camara, A., Ximenes, R., Albuquerque, A.B., Mendonca, N.C.: Cloudstep: a step-by-step decision process to support legacy application migration to the cloud. In: 2012 IEEE 6th International Workshop on Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 24–24 September 2012, pp. 7–16 (2012)
10. Gibson, J., Rondeau, R., Eveleigh, D., Tan, Q.: Benefits and challenges of three cloud computing service models. In: 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), 21–23 November 2012, pp. 198–205 (2012)
11. Garg, S.K., Versteeg, S., Buyya, R.: A framework for ranking of cloud computing services. Future Gener. Comput. Syst. **29**(4), 1012–1023 (2013). http://dx.doi.org/10.1016/j.future.2012.06.006
12. Undheim, A., Chilwan, A., Heegaard, P.: Differentiated availability in cloud computing SLAs. In: 2011 12th IEEE/ACM International Conference on Grid Computing (GRID), pp. 129–136 , September 2011
13. Smirnova, I.: Impact of cloud computing on global software development challenges. In: Proceedings of Cloud-Based Software Engineering, pp. 71–78. University of Helsinki, Helsinki (2013)
14. Haig-Smith, T., Tanner, M.: Cloud computing as an enabler of agile global software development. Issues Inf. Sci. Inf. Technol. **13**, 121–144 (2016). http://www.informingscience.org/Publications/3476