The International Journal Research Publication's
ISSN:2251 1563
Research Journal of Science & IT Management

Software Development Life Cycle for Developing Cloud Based Software Suit

Prof. Mukund A. Kulkarni
Research Scholar,
Bharati Vidyapeeth Deemed University
Institute of Management, Kolhapur.
Mob. 9890166249
Email: my_mukund@yahoo.com

Prof. Dr. S. S. Gulvani
Research guide
Bharati Vidyapeeth Deemed University
Institute of Management, Kolhapur.

ABSTRACT:

Cloud application development requires a different approach than the traditional software development life cycle (SDLC), as the cloud provider becomes a critical success factor of the overall application development and implementation. In a traditional software engineering, more emphasis is put on the functional aspects of development because it is deployed after development on client infrastructure with implicit security, compliance, control, operational transparency and perceived service level requirements. Another important factor is the cost of operations. Its cost can be easily esteemated as entire software engineering appraoch is clearly defined. The main objective of this paper is to focus on the different factors to be considered in SDLCfor cloud based software suit and outline the motivation, inputs and deliverables of each activity.

Paper also explores apps, web service, plug-ins collectively called as cloud based software suit. They all are to be developed on a foundation based on certain principles of cloud computing at a fundamental level. These principles are Elasticity, Discoverability, Reachability, Scalability, and Supportability which are briefly discussed in paper. Major steps in SDLC includes designing, coding and Testing and all these steps are different up to some extent for cloud application development.

Key words:

Cloud computing, Cloud SDLC, cloud software engineering, Cloud Applications,SaaS.

Cloud Computing:

Cloud computing is the new era of computer world. It is changing existing way of computerization very rapidly. It is being applied in different sectors. To understand cloud computing, we can divide it in two categories

1. Understanding Architecture (Hardware setup)
2. Understanding its applications (Resource sharing)

A huge research is going on cloud computing and its applications. Basically cloud computing provides everything in the form of service. It is divided in three parts

1. **IAAS** (Infrastructure as a Service): It provides hardware resources as a service when required. e.g. Storage space.
2. **PAAS** (Platform as a Service): It provides Operating Systems as a service when required. e.g. Widows hosting service
3. **SAAS** (Software as a Service): It provides application software as a service when requested. e.g. Google Adds

Traditional SDLC:

In traditional software development life cycle we follow different models of software development life cycle that contains data analysis, Designing, coding, Testing and deployment phases as a major phases. These stages are almost same in all areas of development. However some cutomisation is required in these phases as per need of technology used for development. This change in implementaion varies developer to developer. Cloud based software development also need some changes in SDLC

Concept of Cloud Based Software Suit:

Software suit is a collection of services, plu-ins, APIs to connect with other applications, web services, web-base applications and mobile apps. All of them can work together or independently but they are similar in behaviour so that it will be user friendly. Now-a-days, various services are made available in the form of software suits. For example now banking is possible by using web application through browser and same can be possible using mobile apps. All these facilities shares same data stored on cloud. While developing such software suit more focus is to be given on two things. One is common resources and their sharing capabilty and second is connectivity with other apps through APIs or plug-ins. For this purpose we need a different approach in implementaion of SDLC for Cloud based software suit.

SDLC for Cloud Based Software Suit:

Some of the key features of cloud based applications are elasticity, Discoverability, Reachability, Economic Feasibility, Scalability and supportability. These keypoints makes cloud computing so popular and now most of the applications are diverting towards this technology. It has made a revolution in the field of IT development. At the time of development we have to keep in mind above key points so that customer/user can get real benefit of cloud computing. Therefore there is need to slightly modify SDLC phases for cloud application (SaaS) development. (Zack, 2011)

Points to be considered for Cloud Applications:

- **Elasticity:** It is the ability to scale resources. To the consumer, the cloud appears to be infinite, and the consumer can purchase as much or as little computing power as they need. This is one of the essential characteristics of cloud computing. This features reflects relative changes in desinging and coding phases of SaaS development. Here developer must provide facility to utilises resources as per requirement.
- **Discoverability:** Discoverability helps service consumers by reducing the distance between the vision and the deployment. The main benefit of service discoverability for a provider is that it is a force multiplier in reducing the cost of sale due to the network effect created through the automation engines supported by service architecture. Essentially, discoverability will enable the creation of a service engine. Service implementation has to incorporate discoverability at its core through rigorous service architecture, subscription and provisioning automation, and investments in sales and marketing engines. It leads a major change in deployment and maintainance phase.
- **Reachability:** A reachable service is one that is available globally and serves up customers spanning large enterprises and small business alike. Since cloud computing removes infrastructure obstacles for a global deployment, the service can be consumed by a Long Tail of customers given that the service functionality can be automatically adapted to the given customer context. The aspects of a service that affect reachability are global deployment, configurable functionality, local compliance, language, currency, invoicing and supportability. Cloud platform provider selection and the service architecture will play an important role in

enabling the reachability of a SaaS service. Application service must be desinged in such a way that it will improve reachability of service.

- **Economic Feasibility:** A cloud-hosted service has to be economical to operate at varying scales of customer usage. The service subscription fee should be greater than the sum of the cloud usage cost, both direct and indirect. The variable part of the service cost model is the cloud resource usage that can eat into the profit margins if not reflected in the cloud pricing model. Independent Software Vendors (ISVs) main business is to build and sell software and services for enabling their customers. In the traditional deployment of a shrink-wrapped software package, perpetual licenses may work as the ISV is only responsible for the software bits; the dynamics of the resources like network bandwidth, server licensing, and storage capacity is entirely up to the customer. With this implicit assumption of the resource availability and the absence of any direct consequences of usage economics, the architecture may be less rigorous in clamping down the resource consumption. In a cloud setting, the architect has to be extremely diligent about every architecture decision that impacts the usage of resources such as compute, storage and networking. Cloud Platform Providers need to help SaaS architects with cost oriented service architecture (CSA) models that incorporate economical resource usage as the fundamental tenet.
- **Scalability:** A cloud service has to support the multi-tenant platform components required to deliver consistent performance across varying conditions of usage. This is fundamental to the creation of a service ecosystem that is self-sustaining and embodies the other tenets like discoverability, reachability and supportability. The service architecture must enable the attainment of near linear scalability by eliminating resource bottlenecks through rigorous application state management. In a shrink-wrapped software installation, the high water mark of the scalability is pre-established due to the decisions made at the time of infrastructure deployment in terms of network topology, schematics of the server farm and the storage fabric interconnects and the configuration. Storage, for example, is often in the critical path of scalability. In a cloud deployment, the virtualized storage removes such scalability constraints; an architect can be liberal in storage partitioning and placing them on different physical volumes and eliminate storage as the bottleneck. Similar observations are true for compute and network resources.
- **Supportability:** Supportability often takes a backseat relative to functionality in shrink-wrapped applications because there are ISV services organizations, customer technology support teams, and customer and ISV helpdesk teams to jump in and help fix the application problems. They often have full access to the bare metal infrastructure (compute, networking and storage) and application state at run time. This is really expensive and a cloud hosted service, designed to optimize economics, can't afford helpdesk calls as frequently as in the case of a shrink wrapped application. A cloud hosted service needs to devise support strategies with full awareness of the greater lack of control of the deployment infrastructure and the application's run time state. The cloud service architecture should adopt supportability as one of the fundamental tenets that guide the solution design and implementation. In addition to application issues, supportability characteristic of a SaaS service should give top priority to performance, availability, back up/recovery and disaster recovery due to the external hosting environment. End user support processes need to consider online forums as one of the viable support options in addition to the more expensive support options that require human intervention.

The SDLC phases for SaaS development needs to support the creation of cloud hosted services that reflect the above tenets at a fundamental level.

SDLC for Cloud-Based Software Development

SaaS development requires a rigorous approach towards cloud provider assessment from the platform capabilities as well as the operational enablement perspective. Due to the explicit list of detailed activities during initial evaluation of the cloud provider, acquisition of production subscription and

integration of operational frameworks and processes, these respective phases have been added to the traditional software development lifecycle.

The SaaS service being developed is not the first one within the organization, the Evaluation, Subscribing and Operations phases will be less detailed due to leveraging the work that has already been done during the previous project.

The resultant SDLC for Cloud Software suit Developmnent is as shown in fig.1. While preparing this model, traditional waterfall model (Pressman., 2009) of SDLC is modified by adding some new phases.

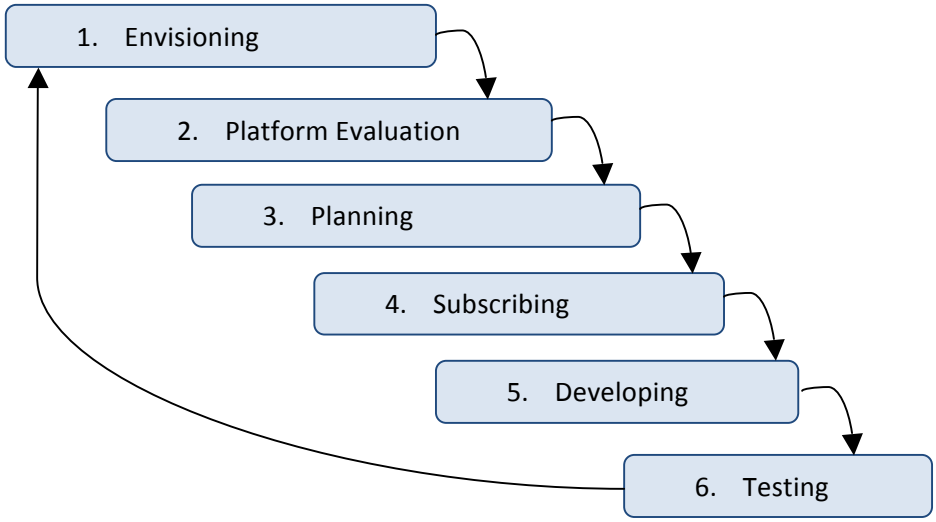

Fig. 1 SDLC for Cloud-based Software Development

1. **Envisionning:** At this phase existing cloud services are analysed and new requirements are generated. As per requirement a new cloud services are visualised. Developer will identify new business opportunities, how to upsell to existing customers, how to expand the customer base by reaching the Long Tail, and they will strategize to extract more value from their intellectual property
2. **Platform Evaluation:** The evaluation phase will identify a cloud service provider for the visualised cloud service. Architecture proof points will be intersected with a cloud provider's platform capabilities in arriving at the decision of "fit for purpose".
3. **Planning:** After having identified a cloud platform that is feasible for building the SaaS service, the planning phase will help plot the course of action for a predictable delivery of the service. Planning largely depends on the type of service and the organizational culture. The rigor of the activities and the resulting deliverables depend on the complexity and the size of the service. The activities in this phase are pretty much similar to those of the traditional software development lifecycle.
4. **Subscribing:** Subscribing is an important phase of the SaaS Development Lifecycle during which a production quality subscription is acquired. In this phase, based on the previous trial experience, the cloud provider is subjected to more scrutiny, driven by the production deployment and operational needs of the service being planned. The architects and IT Professionals will revisit the deployment models; subsequently upgrading schemes, support processes, business continuity and disaster recovery. The procurement team will work with the cloud provider in identifying the right level of subscription, either IaaS or PaaS, with full awareness of the pricing models, what-if analysis, and support costs.
5. **Developing:** This is the service construction phase during which design specifications are refined and translated into code artifacts and supporting documentation. The Developing phase is composed of a series of iterations built on top of the core architecture and high level design specifications. The architecture and detailed design may change based on the discovery of new

functionality and refinement of existing functional specifications. The resource allocation, scope of functionality and project schedule determines the granularity and number of iterations. Developers and solution architects will work hand in hand in designing the iteration make up and involving end users throughout the phase for a quality service delivery.

6. **Testing:** Ther service is continuously monitored and tested for service level agreements, support contracts, compliance, security, and shared infrastructure; activities during the operating phase. The insights acquired during the evaluation, subscription and development phases are blended with the operational characteristics of the cloud platform in creating deployment and operational processes for running the cloud hosted service with the best possible systemic qualities. The evaluation and subscription phases of the SaaS development lifecycle would have given a detailed knowledge of the cloud platform operational aspects.

Conclusion and future scope

For cloud based software development we cannot use traditional software engineering as it is. Hence we need to upgrade the same by considering some key featues of cloud computing technology. Accordingly we need to implement a separate software development life cycle(SDLC) for SaaS Applications/Services.

References:

[1] Guha, R. (n.d.). SOFTWARE ENGINEERING ON SEMANTIC WEB AND CLOUD COMPUTING PLATFORM. *http://www.editorialmanager.com/ijseke/download.aspx?id=5281&guid=3065f3cc-bbd1-4f26-8134-20438154876d&scheme=1* .

[2] Dr. Rahul Malhotra, P. J. (Jul-Aug 2013). Testing Techniques and its Challenges in a Cloud Computing Environment. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol 1 No.3* , 88-93.

[3] Ravichandran, S. (2012). Innovative Method of Software Testing Environment using Cloud Computing Technology. *International Journal of Communication and Networking System, Vol. 01, No. 01*.

[4] Shehla Afzal, M. F. (Sep 2013). A Review on Green Software Development in a Cloud Environment Regarding Software Development Life Cycle: (SDLC) Perspective. *International Journal of Computer Trends and Technology (IJCTT)* , 3054-3058.

[5] CloudBees Inc. Cloudbees.com. [Online]. (n.d.). Retrieved from CloudBees Inc. Cloudbees.com. [Online]

[6] MÜNCH, D. J. (2013). Cloud-Based Software Engineering. Dept. of Computer Science, Helsingfors University.

[7] Pressman., R. (2009). *Software Engineering: A Practitioner's Approach.7th Edition.* McGraw-Hill Higher Education.

[8] Zack, H. K. (2011, October 03). *The SaaS Development Lifecycle(https://www.infoq.com/articles/SaaS-Lifecycle)*. Retrieved from InfoQ: https://www.infoq.com/articles/SaaS-Lifecycle.