ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Comparative Study of Traditional Software Development and Development on Cloud

Manish Joshi,
M. Tech(S.E)
Department of Information Technology
Babasaheb Bhimrao Ambedkar University
Lucknow, Uttar Pradesh 226025, India

Dr. Dhirendra Pandey
Assistant Professor
Department of Information Technology,
Babasaheb Bhimrao Ambedkar University
Lucknow, Uttar Pradesh 226025, India

Abstract:Tim Berners-Lee's has said that "the original idea of the web was that it should be a collaborative space where you can communicate through sharing information" and with his vision of the Web 3.0 the transformation of the World Wide Web into an intelligent Web system. With the advent of new technology new business models are being incorporated to host the applications on the Web as services. These Web services are being designed to be automatically discovered by software agents and exchange data among themselves. Cloud computing model, which has gained a lot of attention, in which hardware, software, tools, and applications are given out as services to users across the globe over the Internet for very low cost or no cost at all. There are many advantages of this cloud computing business model, as it requires very low capital expenditure, it has higher speed of application deployment to market, low cost of operation, and easy maintenance of the software services. Because of these advantages, cloud may soon become the most prevalent computing platform. The traditional engineering practices were not made keeping cloud computing platform in mind. To get the best of the all the advantages of cloud computing, we need to change the traditional way of software engineering. This paper, focus on how cloud computing, is going to influence the traditional software engineering processes to develop quality software and how we adapt the traditional software engineering process to cloud computing environment.

Keywords:Software engineering; Cloud computing platform; Cloud computing platform; Cloud Services; Cloud Development life cycle

I. Introduction

The World Wide Web (WWW) came into existence in 1990 by [1] Tim Berners-Lee, since then the World Wide Web has been a container of data in form of document, images, graphics, etc., and the amount of data is increasing very rapidly. The amount of data present on the Internet is so large, that it is impossible for humans to comprehend the data all by themselves and unless the information from the data can be comprehend and inferred quickly, the data is of no use. Thus, [2] Tim Berners-Lee envisioned Semantic Web or Web 3.0; this technology will transform the data into linked structure which no longer will be only for human but also for machine processing, thus will help in faster processing of data by machines. This linked data will be used by software agents on behalf of human users and will help in produce intelligent information. The transition of the World Wide Web where data integration, inference, and data exchange among different applications is possible, new business models have been conceptualized for application deployment and delivery over the Internet. The applications are being hosted on web (Web Applications or Web Services) and can be accessed over the Internet by users all over the world.

In another business model, where cloud provider focus on hardware side of the application development like processors, storage, memory, operating system, and application development tools and the service provider on the software which can be delivered as utility or application to the indented user over the Internet. This business model is what is called as Cloud Computing where software services are hosted on the web which is shared by multiple clients as and when required. It can also be summarized as the renting hardware and software from Cloud Service Provider to deliver an application service over the Internet. The users can access files and application from device which have access to the Internet.

There are many benefits of this business model like it require no initial capital investment, speed of application deployment to market, low cost of operation and easy maintenance of software services. In past few years this business model has gained a lot of popularity over the world, the main reason for this popularity is its distributive nature. In traditional computing environment a company willing to provide some type of service to its customers need to focus on hardware side as well as on the software or application development side which makes the software development process complex. The hardware side require capital investment which otherwise could have been used in refining the software. What cloud does is that it distributes these responsibilities to different parties which specializes in their respective areas. For example, the hardware part is looked after by the cloud service provider which specializes in hardware capability such as providing the server space, firewall, network infrastructure, managing the load balance and the software development is done by the application service provider, which specializes in application development and they work together to provide service to customers. What it does is it reduces the burden from the software development team so they only focus on software development and does not have to manage and maintain the infrastructure or the application stack (operating system, application server, database, and programming language). Another advantage is that the two parties need not to be in same geographical location, the cloud provider may be providing its hardware service from China and a team of software developer from America developing their application software to be deployed on hardware based in China for its customer throughout the world

A. *Types of Cloud Computing*

The cloud model can be classified into following four categories [3] from their deployment perspective.

1) *Private Cloud:* In private cloud, the cloud infrastructure is provided for exclusive use by a single client/organization. It

is not shared with anyone and the organization have full control over the hardware infrastructure. It is similar to dedicated data center provided in cloud. This can be on premise or off premis.

2) Public Cloud: In private cloud, the cloud infrastructure is provided for use to the general public. The whole computing infrastructure is located on the premises of a cloud computing company that offers the cloud service. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider, thus, spate from the customer has no physical control over the hardware infrastructure.

3) Hybrid Cloud: Hybrid cloud is the combination of both private and public cloud. It is used in situation where public cloud is required for its services to the customers and private cloud is used to secure their data through private cloud.

4) Community Cloud: The community cloud infrastructure is shared by several organizations and supports a specific community that has shared common concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise

B. *Cloud Computing as Services*

If we look at cloud computing from a service point of view the cloud can be categorized into [4] three categories:

1) Infrastructure as a Service (IaaS): Infrastructure as a Service (IaaS), in cloud computing environment provides the physical infrastructure for hosting solution and are self-service models for accessing, monitoring, and management of remote data center infrastructures, such as operating system, servers, storage, networking, virtualization and networking services (e.g. firewalls). Instead of having to purchase hardware which require capitol investment, client can purchase IaaS based on its usage.

IaaS users are responsible for managing applications, data, runtime, middleware, and OSes. Cloud providers still has control over virtualization, servers, hard drives, storage, and networking. Many IaaS providers also offer databases, messaging queues, and other services above the virtualization layer as well.

2) Platform as a Service (PaaS): Cloud platform services, or Platform as a Service (PaaS), provide its users infrastructure to develop, run, and manage applications without the complexity of building and maintaining the hardware usually required to develop and launch an application.

The main advantage of PaaS is that it provides them a framework they can build upon to develop or customize applications. PaaS helps makes the development, testing, and deployment of applications quick, simple, and cost-effective. With this technology, enterprise operations, or a third-party provider, can manage OSes, virtualization, servers, storage, networking, and the PaaS software itself. Developers, however, manage the applications.

3) 1.2.3 Software as a Service (SaaS): In SaaS the client uses web to deliver application to its customer, but the applications are managed by the cloud service provider, as a client, they deliver their application to the Cloud Service Provider who in turn launches application on their server and they provide management, maintenance and support for applications, runtime, data, middleware, OSes, virtualization, servers, storage and networking, as it is all managed by Cloud Service Provider

II. COMPARATIVE STUDY

A. *Software System Life Cycle*

Information systems help organisation to capture and manage data to produce information useful for its employees, customers, partners and suppliers. Each information system has a life of its own. [5] Engineering such a software system involves process, methods and tools that facilitate systematic, disciplined and quantifiable approach to the overall software development. Process is the foundation layer and comprises a framework of activities to be carried out regardless of domain, size and complexity, methods indicate how-to of each activity and tools support process and methods. The generic process framework for software engineering comprises communication phase, planning phase, modelling phase, construction phase and deployment phase. Typical set of activities carried out in each of these phases are listed below:

- Requirement phase: Requirements gathering, focus on what requirements, specify the requirements and project initiation.
- Analysis phase: Prepare project schedule, estimate of efforts and task duration. Tracking of the schedule happens in parallel with the rest of the subsequent phases.
- Design phase: Contains analysis phase and design phase. Activities of analysis phase are model the requirements, build prototype and evaluate alternate options. Similarly, design phase activities are translating the requirements into a blueprint for software construction and iterate to a fine grain level details needed for the coding.
- Development phase: Construct the code, in other words implement the design.
- Testing Phase: Unit wise, after code integration and finally the system.
- Deployment phase: Deliver, support and maintain the deployed software.

B. *Switch form traditional to Cloud Computing*

The traditional way of computing is not able to cope up with cloud computing, as the usage of cloud computing grows daily. Some of the primary reason for that are:

- Flexibility: The traditional services are not able to meet up with the changing demands of the customers and as the customer base grows the business are forced to expand their bandwidth to meet the customer demands. Whereas cloud-based services are ideal for businesses with growing or shifting bandwidth demands. If customer needs increase it’s easy to scale up cloud capacity, drawing on the service’s remote servers.
- Automatic updates: One of the benefits of cloud computing is that there is no need worry about

updating the software. The cloud service provider is responsible for the upgrade of the software.

- Reduced Cost: Cloud cuts down on the hardware cost. The client can select or subscribe its choice of hardware from the cloud provider. This significantly reduces the cost of owning hardware. Hence providing more effective use of hardware.
- World-wide access: Cloud provide access to the applications, files and services. This helps in increasing the productivity of the business as file are available on the go and business can work from anywhere around the globe.
- Increased teamwork: Teams can access, edit and share documents, files anytime, from anywhere, hence they're able to do more together. It helps teams make updates in real time and gives full visibility of their collaborations.

Because of the above advantages and the distributive nature of cloud computing more and more people are switchingover to cloud computing environment.

C. *Classic Life Cycle Model*

The classic life cycle model is also called waterfall model. In this model, the workflow is sequential in nature from communication/ requirement phase to deployment/ maintenance phase and the result of each phase act as inputs for the next phase. [6] It is a theoretical and sequential model, not adaptable directly. However, the classic life cycle model has a definite and important place in software engineering work. It provides a template for other software development models, other model more or less are based on classical life cycle model with some modification but the phases in each model remain the same. The classic life cycle remains a widely used procedural model for software engineering. But other process models in use today are more or less based on waterfall model but are basically iterative in nature, which is more practical. We know that customer requirements change over time resulting in extension of life cycle phases. In particular, the design phase needs to able to adapt to change with the changing requirements.

D. *Cloud Service Umbrella Model*

Consider any application software being used. The software needs a platform, which is resident on suitable infrastructure. The functional aspects of the application are dealt with at the software level, with the non-functional aspects spread beyond, reaching up to infrastructure including the platform. Cloud services follow the same pattern. SaaS is existent because it shadows a PaaS, which further wraps an IaaS. At each service level of the cloud, the non-functional facets are dependent on the underlying support. The scope of controls that can be exercised follows the same path and varies from low to high from outermost shadows to innermost. This shadowing of services can be shown as an abstract Umbrella model. This model would help understand the variants of `the classic life cycle phases better. The Umbrella model shows an IaaS shadowed inside a PaaS that in turn is shadowed inside SaaS. SaaS and PaaS are coexistent i.e. they alone cannot exist; similar is the case with PaaS.

E. *Cloud Development Life Cycle*

In development of application for cloud environment the traditional software development life cycle [7] is followed along with cloud development life cycle. As development requires to two parties i.e. Software Development team which is responsible for the development of application for the users and the Cloud Service Provider which is responsible for providing hardware infrastructure for hosting the application on his cloud and maintaining it. Figure 1. Depict the cloud development life cycle along with classic software development life cycle. Let us now do a comparative study of development in cloud along with traditional development model.

1) Requirement Vs Cloud Requirement: Requirement phase for development team is concerned with elicitation of user requirement as clearly as possible also to gather functional and non functional requirement related to the project, high-level abstraction and system requirements, performance and external interface requirements, design constraints. Requirements tell what the system should do and define constraints on its operation and implementation [8]. Requirement descriptions should cover the business domain, applications integration, technology, data and information architecture [9].

Based on the initial requirement received from the development team the cloud service provider generates its own requirement like resource availability, maximum number of user that can access the application at a time, maximum load which will required by their server to handle, and how much downtime/uptime their client wants.

Cloud Requirement can also be classified on the basis of cloud services. For example, in SaaS the choice of services is very important as it will be difficult to accommodate changes in later phases. Hence it is essential to state the customer requirement clearly. In PaaS we need bear in mind the non functional requirement as it is the platform which is offered as a service and is needed to be customized according to the user need. In IaaS it is the infrastructure which is being provided as service the rest similar to typical web applications, the infrastructure services should satisfy the non functional requirement.

2) Analysis vs Analysis of Cloud Requirement: The specification and architecture details are analysed to create a plan to develop the application. The plan is used as outline for the development process. An analysis of risk is done for the entire project and monitoring, management, mitigation plans are also developed. These plans are used for determining the cost of the project and to establish schedule, timeline for each process. The CSP uses this detail plan to map software architecture to hardware architecture.They also use this plan to find the cloud components if any, which can be reused.

In SaaS the team should plan for the customization of services according the user need for the cloud software. In PaaS, the team should plan for application development on cloud platform and deploy it. In IaaS, the team should plan for application development and deployment on specified cloud platform and infrastructure.

3) Design vs Cloud Design: The development team transform the requirement gathered from the previous phase into a blueprint which will the coders to easily transform design into code. The design is to be made keeping in mind the design goals of the web applications [10][11], a detailed design is made which include interface details of all modules, input/output design. In cloud design the CSP is concerned with designing an appropriate method for load balancing and performance design. If proper load balancing is not done it may result in low performance.

4) Development vs Implementation: In development phase the development team is focused on implementing the design model into code. The code written should interoperable and portable. The coupling between the module should be low and cohesion within the module should be high. The cloud service provider implement the code received by the development team onto web service. Any error found during implementation of application should be reported back to the development team for correction.

5) Testing: Testing is one of the most important phase in application's lifecycle, any undetected error can result in failure of the web application. The software development team needs to do extensive testing of application to make it error free. Unit testing should be done on the module as when they are ready.

After all the modules are ready the integration test is done by development team and service provider. The development team does the test in house where as the service provider does it on the cloud infrastructure. Any error found are reported back to development team for correction.

In SaaS the team should focus on regression testing as many add-on services are add later on as customization. In PaaS the focus should be on performance testing because for web application the non functional requirement like performance, scalability, availability is important. In IaaS the focus should be on recovery testing as infrastructure is virtually owned and client has no physical control over the hardware.

6) Deployment and Maintenance: In the deployment phase the cloud service provider takes control as it delivers, deploy, support and maintain the application. In SaaS the deployment is done on the agreement decided between the two parties, the service provider support and maintains the application. In PaaS, the deployment is done on the platform and the platform is configured as per the solution provider. The maintenance and support is decided based on the agreement between the service provider and solution provider. In IaaS, the deployment based on the infrastructure like number of resources, namely, processor, operating system, storage capacity, monitoring and metering. Again support and maintenance is done on agreement between the solution and service provider.

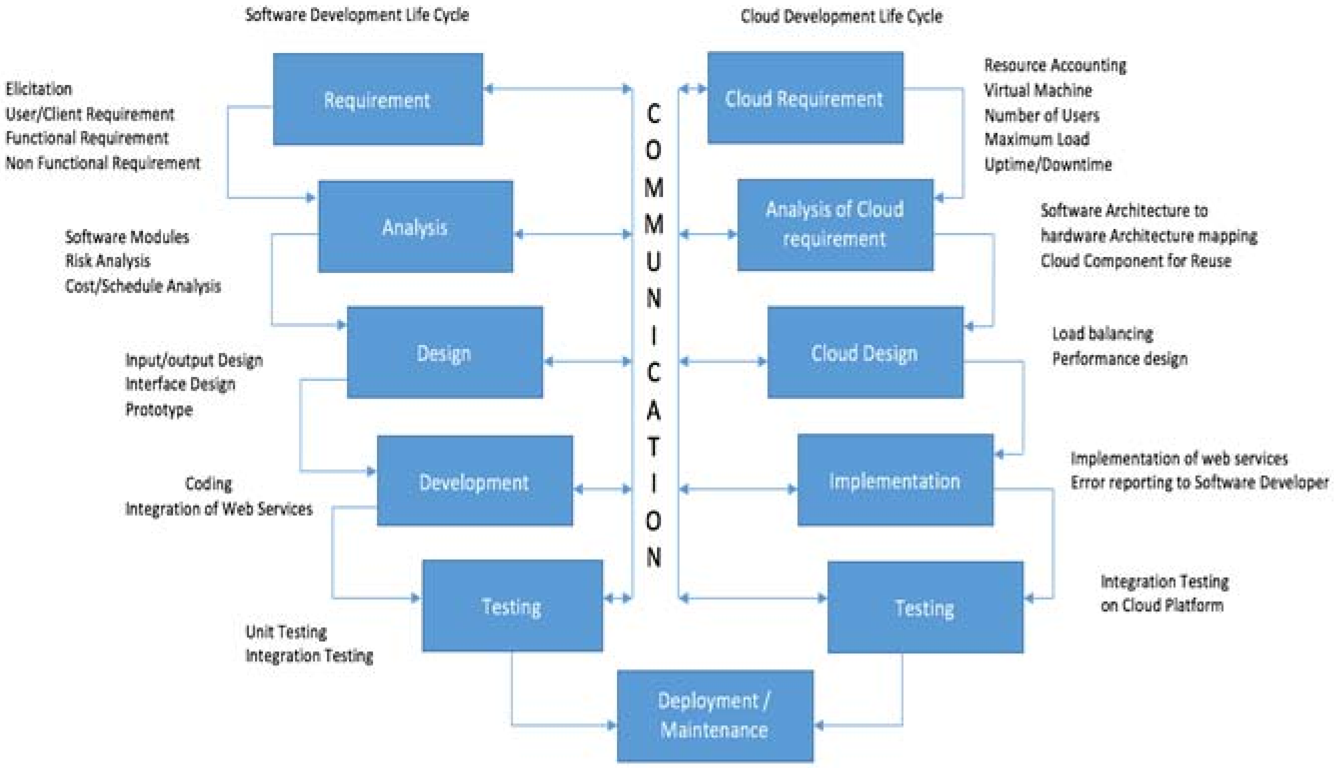

Figure 1: Development in cloud environment

The cloud project is similar to the traditional project but involve three major parties Client, Solution Provider, Service Provider. The communication is one of the key factors in success of any cloud project, lack of communication can result in misrepresentation of requirement which affect the entire project. The main focus in cloud project should be on requirement and design phase.

The cost in development phase in a cloud project is less than that of traditional projects. It is clear from the above illustration that the development in cloud project is similar to traditional project but it is highly platform and vendor specific, and lack standards. Other process model can also be implemented into cloud projects.

III. CONCLUSION

Cloud Computing being an emerging technology lacks in standards, there is need to define standards which can help in streamline the software development process in cloud. Software engineering process models need to be researched which can help transform the traditional developments into cloud and reap all its benefits. Also, safety and privacy issues of data in cloud computing platform need to be considered seriously so that cloud computing is truly accepted by all.

IV. REFERENCES

[1] Berners-Lee: Future of the web. http://dig.csail.mit.edu/2007/03/01

[2] Radha Guha. "Impact of Semantic Web and Cloud Computing Platform on Software Engineering", Software Engineering Frameworks for the Cloud 3 Computing Paradigm, Computer Communications and Networks.

[3] Cloud Computing: What is Infrastructure as a Service https://technet.microsoft.com/en-us/library/hh509051.aspx

[4] K. Chandrashekran “Essential of Cloud Computing”, CRC press M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[5] Suchitra Ravi Balasubramanyam “Cloud-Based Development Using Classic Life Cycle Model”, Software Engineering Frameworks for the Cloud 3 Computing Paradigm, Computer Communications and Networks,DOI 10.1007/978-1-4471-5031-2_1

[6] Radha Krishna and R. Jayakrishnan “Impact of Cloud Services on Software Development Life Cycle”, Software Engineering Frameworks for the Cloud 3 Computing Paradigm, Computer Communications and Networks,DOI 10.1007/978-1-4471-5031-2_1

[7] Pressman, R.S.: Software Engineering: A Practitioner’s Approach, 7th edn. McGraw Hill, New York (2009)

[8] Sommerville, I.: Software Engineering, 8th edn, pp. 131–164. Pearson Education, Harlow (2009)

[9] Jalote, P.: An Integrated Approach to Software Engineering, 3rd edn, pp. 67–211. Narosa Book Distributors, New Delhi (2008)

[10] Pressman, R.S., Lowe, D.: Web Engineering: A Practitioner’s Approach. Tata McGraw Hill, New Delhi, (2011).

[11] Suh, W.: Web Engineering: Principles and Techniques. Idea Group, Hershey, pp. 1–22, 81–82 (2004)