Software Engineering for Cloud Computing Role of Secure SDLC

Dhruvil Shah

Institue of Technology

Nirma University

19BCE248

Jurin Vachanni

Institue of Technology

Nirma University

19BCE286

***Abstract*—Recent NEWS revealed that the Ransomware attack on a hospital in Europe could be held responsible for patients' death. It is the first recorded death in human history due to a cyber attack. Investigation revealed that potential security loopholes had been identified and reported earlier but the team ignored this warning; which later had severe consequences. This incident has encouraged us to carry out detailed research and analysis in this area and how one's approach at early stages of SDLC plays roles in various aspects. The rapid development of the cloud computing IT based industry and its desire to use it is clearly seen now-a-days. The use of cloud computing in software engineering is not a new thing. But according to new cloud computing technology and their requirements, there is a requirement to have different methods to deal with it. Cloud computing faces several issues and challenges at all the stages of development in software development life cycle (SDLC). In this paper we have described the Life cycle of SDLC and also provided challenges faced during cloud computing in Software development life cycle.**

***Index Terms*—software engineering, cloud computing, Software Development Lifecycle**

I. Introduction

When the enterprise needs to increase their computation efficiency they had two choices either buy additional hardware or make the IT operation more efficient. Cloud computing is a completely different approach in providing resources to organizations in which computing resource maintenance will not be a concern for the organizations. Cloud Computing is providing services such as storage, software, analytic, intelligence over the internet. It allows the developer to focus more on the business product rather than worrying more about side services. Moreover, It also benefits financially by just paying for the services used within the duration.

If we talk in terms of Software Engineering then developing a particular software from scratch is a big challenge, as we need to choose a suitable model, following a set of activities throughout the project mainly termed as umbrella activities. The above-mentioned things are common to each project but for large scale projects, there are always big challenges on the way. Particularly talking about Cloud services plays a major role as the majority of the services are provided by platforms such as AWS, IBM, Google is not directly implemented by software developers and they need to integrate inside the software.

There are many challenges associated with using conventional software development methods in the cloud. So, before you start developing any application or software in the cloud, you should consider the challenges and issues in the software development lifecycle. This article explores the challenges of the cloud software development lifecycle.

Now the question arises why such things are needed, so a few instances can clarify more like, In 2010, Microsoft experienced a breach within its Business Productivity Online Suite that was traced back to a configuration problem. The issue allowed non-authorized users in their offline address books to access employee contact info. It affected only a small number of users, but it is worth noting. In 2012, the DropBox accounts were hacked to more than 68 million. Robbed credentials have gone to a dark web marketplace allegedly. The credentials price at the time was around $1,141 in Bitcoins. Dropbox responded by asking the user base to reset the sitewide password. Hackers robbed 167 million LinkedIn email and passwords and placed them on the dark web in May 2016.LinkedIn, in response, performs a 2 step authentication, which is the option of entering a pin code on your mobile device.An agile software development model, considering the software development challenges from the developer and provider view. Although the study is comprehensive, but it's focused on role categorization and there is a small amount emphasize on development challenges. Taking about these paper we have in total 5 sections which consists of Introduction part, Related Work part which depicts the work done so far, brief about cloud computing, SDLC Life cycle, Implementation of Secure Software Development Life Cycle in Cloud Computing, and last one concludes with conclusion.

II. Related Work

So Far much insightful work has been done on the defined topic. Many researchers have proposed different works related to it. Some of them have been depicted here. In[1] the author

Decade	Summary of trends and R&D efforts
1950s	Emphasis was adoption of other branches of science to aid in the evolution of software processes and methodologies. Introduced increasingly unpredictable and diverse range of fast-growing problems.
1960s	Application of low-risk, fast, out-of-the-box processes for approaching and solving software development problems. This method often introduced defects, needing patches or rework.
1970s	Use of structured and formal methods in a bottom-up approach towards goal-oriented and purposeful software projects, with priority given to early elimination of errors. Often required pre-determination of system's purpose and domain understanding. Resulted in issues with compliance, scalability, reusability, and process overheads.
1980s	Adoption of various approaches towards increasing productivity in software projects. Mainly directed towards aspects such as: staffing, architecture, compliance, component reuse, process improvement and maturity. Productivity increased, but skepticism flourished, whilst the rate and scale of innovation reduced.
1990s	More emphasis on reducing the time to market software, improving usability and usefulness of software, as well as maximizing returns on investment. Popular methods of this decade include agile methods, product reuse, concurrent processes and rapid composition methods. This sometimes resulted in overambitious and unrealistic milestones and deadlines; incomplete specifications, incompatibility, and lots of time spent on rework and integration
2000s	Use of adaptable methods e.g., model and plan-driven methods, hybrid agile methods, as well as service-oriented architectures. These were adapted towards addressing dynamic increases and changes in business and stakeholder needs. More attention was paid towards integrating systems and software engineering. Negatives include issues with scalability, and clashes in models and methods used
2010s	Emphasis on creation and adoption of value-based methods, enterprise architectures, enabled by emerging paradigms. These were geared towards: ease of use for end users to build own systems, scalability, global connectivity, agility, and use of diverse processes, components, platforms, skills, and practice and on-demand resources as services. These have introduced: complexity, distribution, proliferation of incompatible tools and inadequate methods, time zones differences, development culture and practice clashes, problems with continuous synchronous integration, compatibility with legacy applications and traditional processes and practices.

Fig. 1. Summary of a review of seven decades of RD efforts and effects on the development process

has proposed a solution moreover related to services of cloud which is most probably concern to software developers rather than audience using it. Also, there is as such no role in SDLC mentioned throughout the paper. In [2] the author mentioned the importance of SDLC but in a very generalized form not a part of deep research. In[3], author considering the provider and software development challenges from the developer's point of view.

The study is more often a comprehensive one, but more light are given on role categorization and there is a little emphasize on development challenges.To date, the focal point of the majority of RD efforts withinside the location of Cloud-primarily based totally software program improvement is at exceptional imbalance. Most deal with particular factors of the improvement process,ensuing in inadequate interest being paid to different factors similarly undermining collaboration. An assessment of related literature exhibits that little or no RD efforts were devoted closer to improving collaboration in software program improvement in general, as properly as, within the Cloud.

III. Cloud Computing

Cloud computing is the technology which is used to distribute the larger computing resources into smaller subprograms through larger networks. This subprogram is provided to a huge system having capabilities of calculating, analysing and searching to get needed information. By the use of this technology billions of information can be accessed easily.

Cloud computing allows users to use application service at location. The requested resources are not stored in the fixed physical entity but rather it is on cloud. Applications which are running on cloud can be accessed by any device like laptops or cell phones through the network. Cloud computing is not for particular applications under cloud rather it is the same cloud for different applications. Cloud computing is an on demand shared pool that can be rapidly released with minimum management efforts. With recent trends in cloud computing, technology has become more flexible and scalable than ever. This helps industries to have better control over their data. It also helps provide better levels of security in each data center. The most integral components of the latest cloud computing technology are highly organized.

This means that potentially however your information gets processed is improved. In fact, this is often what will increase the possibilities of cloud computing opportunities. Below are a number of the foremost spectacular and promising predictions on the long run of cloud computing 2020. Most of those predictions facilitate businesses decide if investment their time in new technology in cloud computing is worthy or not.

There are three most commonly used models of cloud services:

A. IaaS (Infrastructure-as-a-Service)

- IAAS is used by network architects.
- IAAS gives access to virtual machines and storage.
- It provides visualized computing resources over the internet.

B. PaaS (Platform-as-a-Service)

- PAAS is used by developer.
- PAAS give access to run time environment to deployment and development tools for application.
- It is a cloud computing model that delivers tools that is used for development of applications.

C. SaaS (Software-as-a-Service)

- SAAS is used by end users and accessed by end users.
- It is a service model in cloud computing that host software makes available for clients.

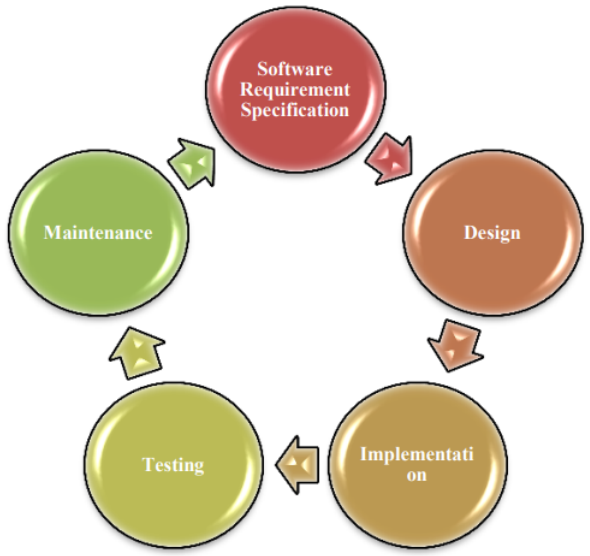

Fig. 2. Green model

IV. Life Cycle of Software Development in the Cloud

To utilize new technology with highest productivity there is a need to adapt related models and methodologies. To increase productivity in cloud based software engineering there should be synchronization between cloud and software development life cycle. There are several software development life cycle models. In this paper, a software development model that has been created for cloud computing has been selected as base.Software development lifecycle in Green method is shown in Fig. 1. Various levels of the model are:

A. Determining software requirements

Normal software requirements are classified into functional requirements and non functional requirements. Functional requirements are like requirements of the software. Non functional requirements are security, performance, reliability. Determining software requirements in cloud includes few issues to be resolved:

- Converting user oriented requirements to service oriented requirements.
- Security and communication requirements
- Appropriate cloud architecture

B. Design

Designing phase changes a user's requirements into a form like structure which helps the programmer to code and implement. There are three levels of designing phase in software development:

- Architecture design - At this level designers can have a clear picture of the proposed system.
- High level design - It focuses on the implementation of components into the system.
- Detailed design - implementation of what we see.

C. Implementation

Software implementation stage involves transformation of technical data into ready software which is to be tested. In cloud environments software implementation includes few of the service models. Few of the necessary are listed below:

1) Data Transfer:

- Data transfer is the information to be transferred from server to cloud service provider to local device and vice versa is true.

2) Data storage:

- Data storage is information to be stored on the cloud by a cloud service provider.
- Storage data is managed and operated by service providers.

3) Application access:

- Application access can be done from anywhere in the world and at any time.
- It can be accessed by any type of device like laptops, mobile, PCs, etc.

4) Backup:

- Cloud backup is to send the physical document or virtual file to the cloud to overcome data loss by device failure.

5) Authentication:

- Authentication is to verify who you are.
- With a single signal on all data access can be provided.

6) Data recovery:

- Data recovery is to get the information lost due to device failure from cloud.

7) Risk handler:

- Risks like security breach, threat, cyber attack are being handled by risk handlers.

D. Test

Testing in software development is a very important part of software development life cycle. 40 percentage of resources are spent on testing to find bugs in software which cost high. Software testing in the cloud is very difficult. Cloud-based testing is done with latency, performance, security, integrity and adaptability. Complexity of testing and repairing bugs in a cloud environment needs more time and energy. Different tools and methods are developed to test cloud applications. The authors of [18] presented an overview of these tools and methods.Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.

E. Support and Maintenance

The application must be supported efficiently while the software is running. Software maintenance depends on selective service models and methods. In a cloud environment, service providers are responsible for the infrastructure of the software. Most of the users are mainly attracted by the services

Fig. 3. Cloud testing approaches

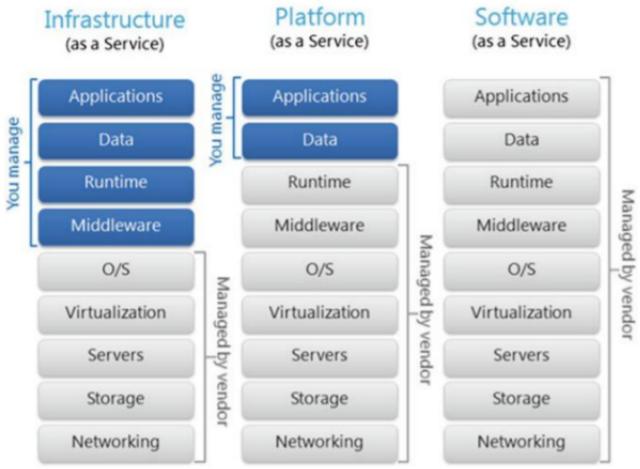

Fig. 4. Services Provided at each interface

of the respective organization. If we take an example of the most used cloud technologies like google drive, dropbox, or AWS-like technologies we can see the features are much restricted by the numbers but the services they provide to their client are more often attractive to the users. The servers are well trained to give responses in milliseconds when requested. Also, free services are provided to make users use particular services. So it can be said that the importance given to the implementation phase should be in an equal amount for the support and maintenance phase.

V. Software Development Challenges in Cloud

So Till now in broader aspects, we have analyzed some of the methodologies to overcome the problem. Now talking in terms of the Software Development lifecycle we can examine each stage and act accordingly. Mainly the stages considered here are Software requirement challenges, Design challenges, Implementation phase, Testing, Support, and Maintainance.

A. Determining software requirement challenges

Before developing any software we need to analyze at the first phase why we need to develop such a thing? What are the requirements? , Is it feasible or not? So Mainly requirements are categorized into three parts Function, Non-Functional and other requirements. Let's look into each phase one by one.

1) Functional requirement : Functional necessities square measure related to applications and business model. however considering the utilization of cloud computing, there square measure challenges with useful necessities:

- Prioritizing specific requirements: specific SaaS
- Requirements should be thought-about and prioritized

2) Non-Functional Requirements: In addition to the useful necessities and package-related non-functional necessities, cloud computing desires to consider other non-functional necessities like security and privacy, reliableness, delay, expandability, and availability. so as to grasp these necessities, first the parameters below the management of client and marketer in A different cloud services model should be known. Cloud service models square measure shown in figure three thoroughly. This model shows the responsibility level of security and privacy and reliability for clients and marketers. different necessities The other necessities that square measure nor useful are neither nonfunctional, and it's necessary to contemplate them in cloud software development, demand analysis part square measure as follows:

3) Lack of Standards for development: Cloud technology is in the early stages of its development. therefore there's no unique commonplace for its API and applications development.

4) Vendor Lock-in: relating to completely different cloud service vendors, customers ought to be ready to opt for acceptable services from completely different vendors. There should be no vendor lock-in for patrons.

5) Cloud Evaluation: Includes enterprise readiness assessment to maneuver toward the cloud. This assessment helps to see the business cases and come of interest

6) Service Level Agreement : The shoppers ought to be assured regarding computing resources within the cloud and the quality of service and performance. These square measures are typically considered in a sort of Service Level Agreement

B. Designing Challenges

Designing doesn't make a UI of software or some templates it has a broader concept which we are going to see in the below description. This phase plays an important role because

everything is nearby future is going to be on virtual platforms. Heterogenous resources are used everywhere so compatibility within software can't be compromised. A few of the challenges are mentioned below:

- The first and foremost thing to be considered is compatibility because the cloud is everywhere and it should provide services to each type of virtual platform. Let's take an example of google drive, where billions of users store their information, and also it can be integrated with multiple platforms.
- Reusability should be taken into action.
- Parallelism should be achieved which means the cloud should be at least as powerful as the hardware configuration of the user and smooth services should be provided.
- Apart from all the functionality it primary focus should be on user privacy and error detection techniques. If the service provided to the user is not satisfying then the above-performed actions are in vain.

C. Implementation Challenge

After verifying the design, the implementation/coding phase begins. The main Challenges associated are :

- If you want a large number of users to be using your services then they should be reliable as well as cost-efficient. The transfer cost for data or storage should be minimal. The concern associated is that world s changing at a rapid rate so if it is not developed keeping future technologies in consideration then redesigning costs will be much higher.
- The next thing is topological problems as the things are moving from status to dynamic network architecture where dependency changes. Accessing things all over the globe should be the primary concern of any cloud technology.
- Billing must be done in terms of usage, not the static one which attracts more amount of users and is also beneficial to both parties. Some amount of usage should be costless which would bound the user to use that service more frequently and become used to it like Google photos, drive, email, and many more.

D. Testing challenges

Challenges concerned with this phase are:

- Expandable and Performance test which tests on the efficiency of services when the product is in the market. Severs should be made available to give service to billions of users at the same time. The second thing is Expandable which means when new features are added it should not get slow down or shouldn't lag.
- Security Test is the primary thing needed in any cloud environment. When your product is dealing with major audience which has not to have that much technical background then these things shouldn't be avoided in any situation. The privacy should always be given more priority than some penny of money.
- Integrity testing should be concerned about connections, communication with legacy systems and their connections to the applications, appropriate test scenarios and criteria must be used.

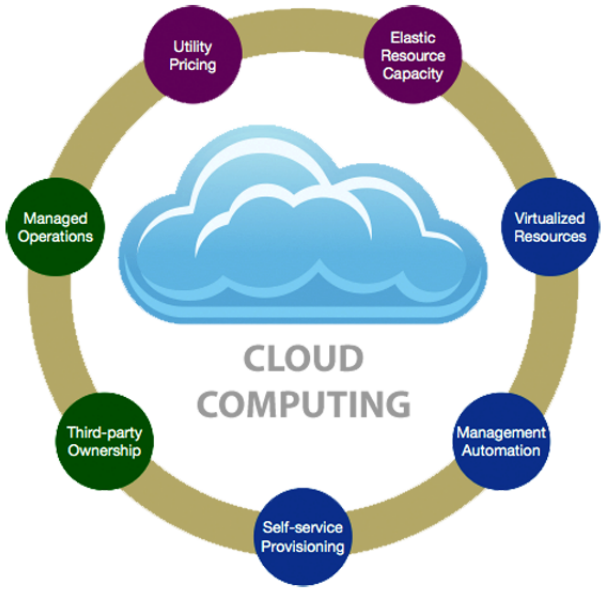

Fig. 5. Overall pictorial representation of cycle

E. Support and maintenance challenges

The issues of cloud software development and maintenance in the SaaS model are examined in this section. The following are the challenges:

- Support for development: If software is created and more resources, such as storage or bandwidth, are required, it is the provider's job to meet the demand. The provider is having issues with resource allocation.
- Service Level Agreement: To be able to meet SLA standards, the service provider must effectively manage resources.
- Cloud software support must be able to estimate cost and resource solutions, as well as cut maintenance expenditures.

VI. Future Scope

The proposed cloud-based SDLC approach focuses on which stages of the process may be done outside of the cloud and which phases can be implemented in the cloud. However, there are other problems that are unique to each stage of the life cycle. As a result, there is plenty of room to continue the work in terms of phase-by-phase evaluation to identify and address difficulties in the future. Cloud computing is also closely linked to concepts such as the internet of things. It becomes easier for IoT to ensure performance, security, and

functionality when data is stored in the cloud. The network's speed, which regulates the rate at which data is acquired and processed, would be the only constraint. Everything else about cloud computing will fall into place if the network is fast.

VII. Conclusion

Cloud Computing can be said as an emerging technology at the present stage. It has been developed to a great extent but though there are many things to be explored that have been under-roof for a while. The main concern associated with such services is how it ensures great satisfaction to the users using them. If more of a quantity is being focussed rather than quality then it doesn't make any sense. So in this paper, we have established a step-by-step procedure about how a transformation from the SDLC phase can help in achieving such goals. We have provided in-depth each step-by-step procedure for implementing it in a more specific way. The majority part of this paper deals more with user benefits from the service rather than the profit barrier of an organization or developer point of view. This paper also covers a small talk about how existing cloud services can improve more to overcome bad instances occurring nowadays very frequently.

REFERENCES

[1] Grundy, J., Kaefer, G., Keong, J., and Liu, A., "Guest Editors' Introduction: Software Engineering for the Cloud", IEEE Software, vol. 29, pp. 26-29, 2012.

[2] Shan, T., "Smart Cloud Engineering, Nomenclature, and Enablement", In Proceedings of the 1st International Conference on Cloud Computing and Services Science, Noordwijkerhout, Netherlands, 7-9 May, 2011.

[3] Silva, E.A.N., and LucreÌ•dio, D., "Software Engineering for the cloud: A Research Roadmap", SBES-Software Engineering Brazilian Symposium, 2012.

[4] Pramod, N., Muppalla, A.k. and Srinivasa, K.G., "Limitations and Challenges in Cloud-Based Applications Development", In Software Engineering Frameworks for the Cloud Computing Paradigm, Springer, pp. 55–75, 2013.

[5] Harman, M., Lakhotia, K., Singer, J., White, D.R. and Yoo, S., "Cloud Engineering is Search Based Software Engineering Too", Journal of Systems and Software, Vol. 86, Issue. 9, pp. 2225–2241, Sept 2013.

[6] Patidar, S., Rane, D., and Jain, P., "Challenges of Software Development on Cloud Platform", World Congress on Information and Communication Technologies (WICT), pp. 1009-1013, 2011.

[7] Goncalves, R., et al, " A Multi-Criteria Approach for Assessing Cloud Deployment Options Based on Non-Functional Requirements". SAC '15 Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp. 1383-1389, 2015.

[8] Tariq, A., Khan, S.A., Iftikhar, S., "Requirements Engineering Process for Software-as-a-Service (SaaS) cloud Environment", International Conference on Emerging Technologies (ICET), Islamabad, pp. 13-18 2014.

[9] Shahzad, F., "State-of-the-art Survey on Cloud Computing Security Challenges, Approaches and Solutions", The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN), Vo. 37, Pages 357–362, 2014.

[10] Tapangarg, "SAAS, PAAS and IAAS – Making Cloud Computing Less Cloudy", 2010, Online Available: http://cioresearchcenter.com/2010/12/107/

[11] Bhuvaneswari, T., Prabaharan, S., "A Survey on Software Development Life Cycle Models", International Journal of Computer Science and Mobile Computing, Vol. 2, Issue. 5, pp. 262 – 267, May 2013.

[12] Chauhan, N. and Saxena, A., "A Green Software Development Life Cycle for Cloud Computing", IT Pro, pp.28 -34, 2013.

[13] Krishna, R., Jayakrishnan, R., "Impact of Cloud Services on Software Development Life Cycle", Software Engineering Frameworks for the Cloud Computing Paradigm. Part of the series Computer Communications and Networks, pp. 79-99, 2013.

[14] Sharp, J., et al., "Cloud Design Patterns", Microsoft Patterns and Practices, 2014.

[15] "Recommendations for Implementation of Cloud Computing Solutions", Technical Report of Federal Bureau of Investigation, 2012.

[16] "A Guide to Implementing Cloud Services", Department of Finance and Deregulation Australian Government, 2012.

[17] Jovanovic, I., "Software Testing Methods and Techniques". Transactions on Internet Research, 30–41, Jan. 2009.

[18] Murthy, N.M.S., Suma, V., "A Study on Cloud Computing Testing Tools". The 48th annual convention of computer society of India, pp. 605– 612, 2014.

[19] Cloud Testing, Wikipedia, OnlineAvailable: https://en.wikipedia.org/wiki/Cloud$_{t}esting$.

[20] Nirmala, A.P., Sridaran, R., "Cloud Computing Issues at Design and Implementation Levels – A Survey", Int. J. Advanced Networking and Applications,Vol. 3, Issue. 6, pp.1444-1449, 2012.

[21] Kar, S., "Implementation Challenges and Higher Costs Than Expected", 2013. OnlineAvailable: http://cloudtimes.org/2013/02/13/kpmg-reportcloud-implementation-challenges-higher-costs/ .

[22] Gao, J., Bai, X. and Tsai, V., "Cloud TestingIssues, Challenges, Needs and Practice", Software Engineering: An International Journal (SEIJ), Vol. 1, No. 1, pp. 9-23, 2011