GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: B. Tech

Course Code:BCSE3096

Course Name: Cloud Application Development

Cloud Development Life Cycle

- In development of application for cloud environment the traditional software development life cycle 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

Let us now do a comparative study of development in cloud along with traditional development model.

i)Requirement Vs Cloud Requirement :

•Based on the initial requirement received from the development team the cloud service provider generates its own requirement .

•These are 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.

ii) *Analysis vs Analysis of Cloud Requirement:*

- An analysis of risk is done for the entire project and monitoring, management, mitigation plans are also developed.

- 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.

iii) *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 , a detailed design is made which include interface details of all modules, input/output design.

iv)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 coupling between the module should be low and cohesion within the module should be high.

v) *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.
- In SaaS the team should focus on regression testing as many add-on services are add later on as customization.

Cont…

- 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.

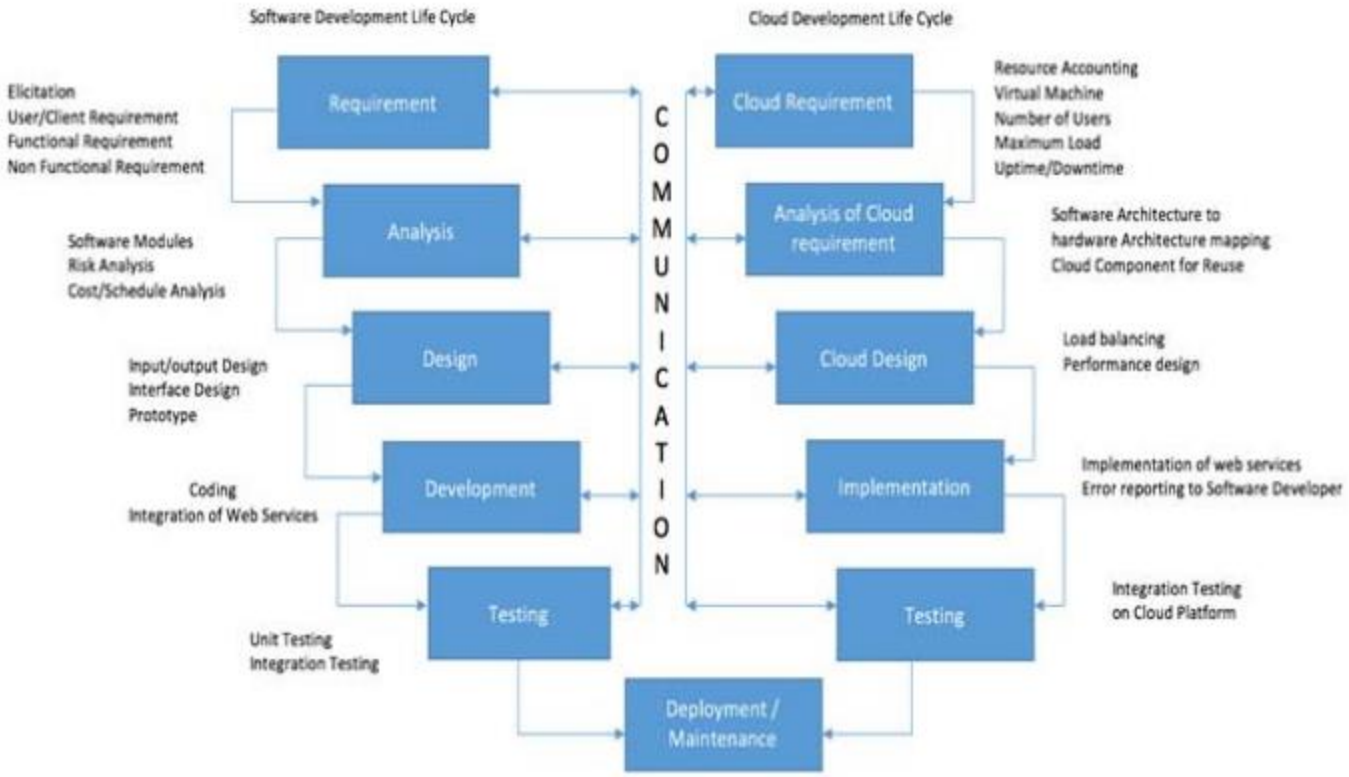
v) *Deployment and Maintenance:*

- 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.

Contd…

•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.


Software Development Life Cycle
Cloud Development Life Cycle
Elicitation
User/Client Requirement
Functional Requirement
Non Functional Requirement
Requirement
Cloud Requirement
Resource Accounting
Virtual Machine
Number of Users
Maximum Load
Uptime/Downtime
Analysis
Analysis of Cloud requirement
Software Modules
Risk Analysis
Cost/Schedule Analysis
Software Architecture to hardware Architecture mapping
Cloud Component for Reuse
Design
Cloud Design
Input/output Design
Interface Design
Prototype
Load balancing
Performance design
Development
Implementation
Coding
Integration of Web Services
Implementation of web services
Error reporting to Software Developer
Testing
Testing
Unit Testing
Integration Testing
Integration Testing on Cloud Platform
Deployment / Maintenance
COMMUNICATION


Thank You