International Journal of Engineering & Technology, 7 (3.29) (2018) 514-518

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Critical study of software models used cloud application development

Dr. Yogesh Kumar Sharma [1] *, P. C. Harish Padmanaban [2]

[1] *Associate professor JJT University Dist Jhunjhunu, Churela, Rajasthan*
[2] *Research Scholar JJT University Dist Jhunjhunu, Churela, Rajasthan*
**Corresponding author E-mail: pchp348@gmail.com*

Abstract

These days, each merchant and IT specialist co-op needs to switch into a cloud domain for better Quality of Service (QoS), Implacability, Performance and sensible Cost. Numerous product engineers attempting to get the advantages of distributed computing and need to get to the cloud conditions with ease and simple access. For this method of reasoning and constant cloud benefits, a solid virtual stage is required. Numerous issues are experiencing being developed and organization of these stages in regards to programming models, application design, APIs and administrations it gave. Then again, there are excessively numerous issues on the customer side, including the constraint of instruments, the cooperation amongst customer and specialist organization and client prerequisites in a particular cloud. As the cloud is intrinsically circulated condition, so it creates holes in correspondence and coordination between stack holders. To adapt to these obstructions and conquer challenges amid programming devel-opment in Cloud Computing, it is important to have a structure which settle the issues and build up the product procedure demonstrate which meet the client necessity and give nature of administrations inside a period and spending plan. In this paper, the writing survey fundamentally centers around the product procedure show with their quality and shortcoming. The writing audit likewise dissects a few traits for programming life cycle includ-ing cost, time, versatility and QoS.

Keywords: *Software Models; Cloud Computing; Cloud Application Development.*

1. Introduction

The intersection of cloud advancements, Internet of Things (IoT) sensors, and enormous information investigation has been perceived as the key ena-bling blend of advances for development in a wide scope of divisions, including military applications, the rising brilliant matrices, savvy structures, applications for e-wellbeing, or cataclysmic event prevention.

A typical normal for every one of these applications is that they are information concentrated, with information being created ceaselessly and originating from heterogeneous sources, for example, sensors or logical gadgets. Moreover, information age rates can shift altogether, and the applications may frequently need to process information in an opportune way ena-bling frameworks to take restorative/vital operational activities, or respond at circumstances [1] critically. Therefore, such applications make frequently utilize computational assets. We will allude to this sort of uses as consistent, information stream applications (CDFA), in-volving an extensive variety of utilizations, for example, logical work processes, pipelines, spilling applications, or some other dataintensive applica-tion where information conditions and simultaneousness play a vital perspective.

It is hard to have one arrangement substantial for every one of these applications in any situation. An approach for their origination can make utilization of any of the current, empowering system. Some illustrative cases can be found in business mists (e.g. Amazon, Google, or Microsoft), or in any of the in excess of 40 ventures of the Apache Big Data Stack [2], which incorporate the pioneer MapReduce computation system, or others, for example, Flume, Spark, Storm, or Flink .

Distributed computing involve administrations situated design. It gives incredible adaptability and on request administrations to clients and lessened in-arrangement innovation overhead with decreased aggregate cost of proprietor dispatch [3].

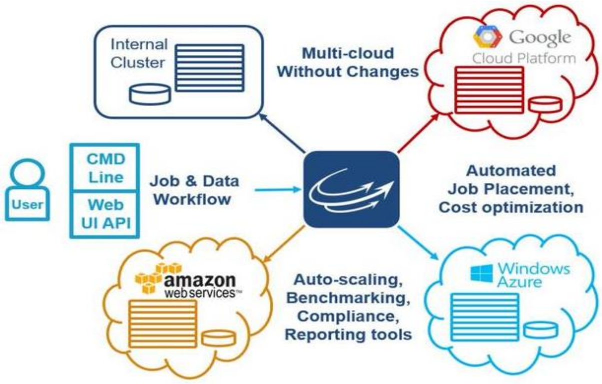
Internal Cluster
Multi-cloud Without Changes
Google Cloud Platform
User
CMD Line
Web UI API
Job & Data Workflow
Automated Job Placement, Cost optimization
amazon webservices
Auto-scaling, Benchmarking, Compliance, Reporting tools
Windows Azure

Fig. 1: Flow of Client and Supplier of Distributed Computing.

Distributed computing alludes to both the application conveyed as serindecencies over the web and the equipment with the framework programming in the server farm, that give those administrations. The administrations are alluded to as SAAS (programming as an administration), and server farm is alluded to as cloud [2].

1.1. Cloud service models

Cloud computing administrations are changing relying upon necessities.

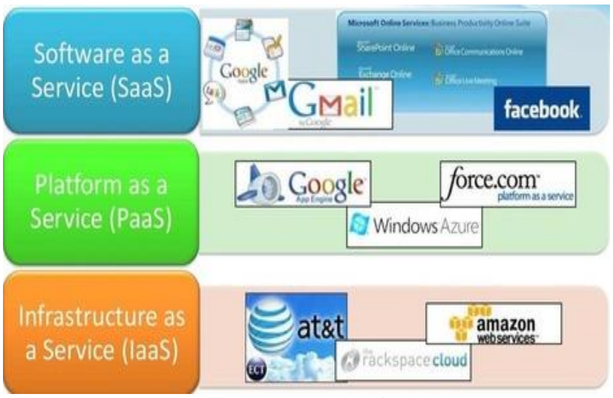
Software as a Service (SaaS)
Google
Gmail
Microsoft Online Services
facebook
Platform as a Service (PaaS)
Google App Engine
force.com platform as a service
Windows Azure
Infrastructure as a Service (IaaS)
at&t
amazon webservices
rackspace cloud

Fig. 2: Service Models of Cloud.

Software as a Service (SaaS)

In Software as a Service (SaaS), application is conveyed as a serbad habit to the clients. Clients can utilize the application gave by the administration creator, and facilitated in cloud.

Platform as a Service (PaaS)

Platform as a Service (PaaS) empowers purchaser to send their own particular programming and applications in the cloud. In that administration consumer can't get to the working frameworks and system. PaaS has two principle purposes:

1) Conveyance of use: Delivery of utilization execute inside a distributed computing condition.
2) Advancement of utilization: All the capacities that are expected to construct, test, and send applications for distributed computing environment.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) gives processing, stockpiling, and systems administration as an administration

2. Review of literature

Various investigations have been done in the writing demonstrate that dexterous approachs and distributed computing has construct its highlights with respect to the necessities for little and expansive associations and also on a comprehension of the task gathering. In the accompanying segment of the paper, two distinct techniques for distributed computing and spry systems that are broadly utilized as a part of business are talked about. Coordinated advancement techniques and distributed computing are a magnificent mix. A short audit of essential research thinks about did in most recent two decades has been introduced in following area of the paper. Boehm and Turner (2005). By and large, coordinated strategies are lightweight procedures that use short iterative cycles. It energetically draws in clients to set up, organize, and guarantee that necessities are dealt with and depends on a group's unspoken information as various to accreditation (Boehm and Turner, 2005).

As per Cho (2008), he features the distinctions interface ing lithe and customary programming advancement strategies and derecorders the scrum structure, which is an outstanding light-footed technique. The creator likewise finds issues and difficulties in scrum in a top to bottom contextual analysis. The distinguished issues are involved confirmation, client interest, and operational condition and scrum ceremonies. The issues and difficulties uncovered by the creators on the establishment of one holder ponder and the discourse of nine representatives is insufficient to set-up an announcement concerning the strategy (Cho, 2008).

As expressed by Cao et al. (2009), the spry headway is predictable for considering the difficulties associated with the engineers, which could impact its accomplishment extensively. The technique by and large relies upon the contained learning of the engineers. The designers won't not archive indispensable choices, and the deficiency of recommended record-continuing with respect to the task may possibly make things more confounded for the groups to portray and appreciate the structure (Cao et al., 2009).

According to Hneif and Ow (2009), the dialog on three spry methods of extraordinary programming, scrum, and light-footed demonstrating represent the distinctions among them. The creators additionally exhibit various recommendations concerning how these techniques ought to be trailed by the experts (Hneif and Ow, 2009).

As per Pallis (2010), in perspective of present request from employment, the individual prerequisites for online arrangements, and the development of the net, distributed computing could be a showing of an imaginative worldview of an extensive scale circled registering viability for business and social request arrangements (Pallis, 2010).

Predictable with Hochmuller (2011), in deft programming create ment, the buyer or customer envoy is in charge of clarify ing supply. For some time, customers did not contain the fundamental abilities to show their necessities. To transcend this subject, the essential designer should co-work with the engineers and clients. The creator additionally features a few repayments that can be accomplished if the architect works bear by bear with the client. This move towards client engineer connections is the establishment for the spry programming advancement process. To accomplish genuine advantages, the designer ought to be an authority in different regions of programming engi-neering, for example, cost estimation, prerequisites affirmation, and certification (Hochmuller, 2011).

As indicated by Mazni et al. (2011) the coordinated strategy importantly affects programming advancement. The strategy positively impacts the designers since it can be brought and included into the administrations offered for the colleagues (Mazni et al., 2011).

According to Khajeh-hosseini et al. (2012), improvement approachs, for example, deft strategies, underline nitty gritty practices that may maybe convey issues of worry as non-specialized and specialized issues identified with distributed computing. The client and developer convey the limitations, and the programming environ-ment fuses cases of non-specialized and specialized issues, separately (Khajeh-Hosseini et al., 2012).

In perspective of the current request from work and individual require-ments for online game plans and the development of the web, distributed computing could exhibit another worldview of an extensive scale dis-seminated registering viability for business (Venkatraman and Wadhwa, 2012). Distributed computing has numerous advantages, however it additionally has challenges, for example, security concerns; information proprietorship concerns; secure and interoperability concerns; undertaking backing and administration improvement issues; condition for online network; and there is worry among engineers about another distributed computing stage without appropriate supervision, which is centers around how to progress completely make utilization of commonplace distributed computing design (Venkatra-man and Wadhwa, 2012).

Reliable with Werfs et al. (2013), on a basic level, the cloud computing condition encourages lithe advancement groups likewise include their skill by consistently creating programming made merchandise (Werfs et al., 2013).

3. Cloud development processes

Cloud improvement forms system, three wide kinds of procedures which exist at introduce are

- Waterfall Processes
- Agile procedures
- Iterative procedures

Most generally utilized lithe procedures in cloud application creation are Scrum and Extreme Programming (XP).

3.1. Agile software methodology

Coordinated programming technique (ASM) gives assistance to some key prob-lems in programming advancement: that product takes excessively cost to create, takes too extensive stretch for improvement, and s ome time does not work legitimately. The most analysts and experienced peo-ple have considered it as precise approach. General research and customary methodologies has outlined that nimble improvement should center around four center qualities (Fowler and Highsmith, 2001).

- Working programming over full documentation.
- Customer participation over contract arrangement.
- Replying to adjustment over arrangement.

Outrageous Programming is a product advancement philosophy that is most broadly utilized as a part of programming improvement industry and it has been developed in the ongoing past years. Outrageous programming accentuates collaboration and enhances programming ventures; for the most part it is certain prescribed for little tasks. Our exploration centers around building up an appropriate approach for consumer loyalty that how XP will be fit for their expansive undertakings and how groups as well as ganizations dependably incline toward client use on extraordinary programming in their tasks, however client are constantly confounded to embrace the outrageous programming.

The most essential cloud element, and the vital quality driver is client [1], and client is most critical in XP. Before begin ing coding, designers lead meeting with client and all through the task do acknowledgment tests and offer criticism to engineer.

Clients every day sit with software engineers and do work with them. On account of every day gatherings and acknowledgment test engineer get right prerequisites of clients and they comprehend what client needs, and they noted in beginning periods of programming advancement life cycle (SDLC), if clients necessities are changed Extreme programming most appropriate for client. Associations dependably favor client utilize extraordinary programming in venture yet client has befuddled to receive the outrageous programming. On the off chance that the designers using our proposed system then client will be fulfilled on his side Different kinds of fulfillment holes distinguished by Brown and Swartz:

- Customer expectations and client encounters hole
- Customer desires and laborer perception of client trusts hole
- Customer encounters and specialist consciousness of client encounters hole

Our principle center around consumer loyalty and feel him that Extreme Programming will be solid in their product advancement process in distributed computing.

3.2. Extreme programming

Extreme programming XP [7, 8] has exceptionally request in programming improvement industry due to its lightweight technique. Despite everything it has a few guidelines however they are been anything but difficult to take after. Around then 31% programming businesses are utilizing deft preparing and 27% [2] are utilizing XP and 4% know about others. XP hones gives a correct shot at perfect time to entire group and client, that the item can be discharge with few deformities. Outrageous programming stresses cooperation and enhances programming venture in five essen-tial ways correspondence, straightforwardness, input, regard, and bravery. The most bewildering XP rules are steady for the client and Programmer that toward the end they can see finish sensible picture of programming. We have condensed XP rules and their exercises.

4. Cloud development life cycle

Being developed of utilization for cloud condition the conventional programming improvement life cycle [7] is taken after alongside cloud advancement life cycle. As advancement requires to two gatherings i.e. Programming Development group which is in charge of the create ment of use for the clients and the Cloud Service Provider which is in charge of giving equipment foundation to have ing the application on his cloud and looking after it. Figure 1. Delineate the cloud advancement life cycle alongside great programming devel-opment life cycle. Give us now a chance to complete a near investigation of create ment in cloud alongside customary improvement demonstrate.

1) Requirement Vs Cloud Requirement: Requirement stage for improvement group is worried about elicitation of client necessity as plainly as conceivable likewise to assemble useful and non useful prerequisite identified with the task, abnormal state deliberation and framework prerequisites, execution and outside interface necessities, outline limitations. Necessities tell what the framework ought to do and characterize limitations on its task and execution [8]. Requirement depictions should cover the business area, applications reconciliation, innovation, information and data design [9].

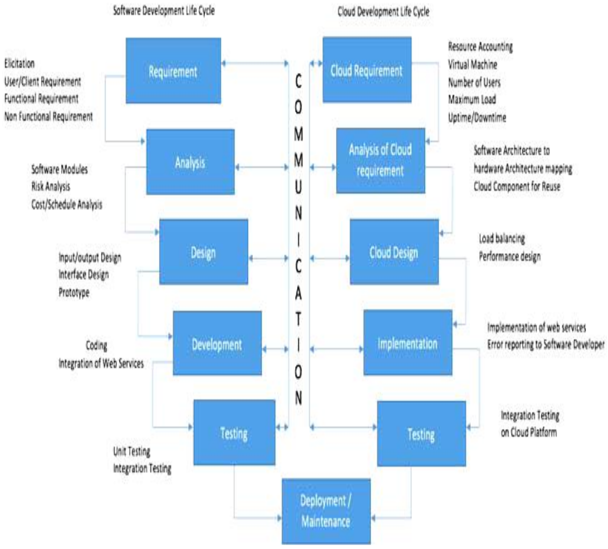

Fig. 3: Development in Cloud Environment.

In light of the underlying prerequisite got from the improvement group the cloud specialist co-op produces its own particular necessity like asset accessibility, most extreme number of client that can get to the application at once, greatest load which will required by their server to deal with, and how much downtime/uptime their customer needs.

Cloud Requirement can likewise be arranged based on cloud administrations. For instance, in SaaS the selection of administrations is extremely im-portant as it will be hard to suit changes in later phas-es. Subsequently it is basic to express the client necessity obviously. In PaaS we require remember the non useful prerequisite as it is the stage which is offered as an administration and is should have been customized as indicated by the client require. In IaaS it is the foundation which is being given as administration the rest like regular web applications, the framework administrations ought to fulfill the non functional prerequisite.

2) Analysis versus Analysis of Cloud Requirement: The particular and engineering points of interest are dissected to make an arrangement to build up the application. The arrangement is utilized as diagram for the advancement procedure. An examination of hazard is improved the situation the whole undertaking and checking, administration, alleviation designs are additionally created. These designs are utilized for deciding the cost of the venture and to set up sched-ule, course of events for each procedure. The CSP utilizes this detail intend to delineate engineering to equipment architecture.They additionally utilize this intend to discover the cloud parts assuming any, which can be reused.

In SaaS, the group should get ready for the customization of administrations air conditioning cording the client requirement for the cloud programming. In PaaS, the group should get ready for application improvement on cloud stage and send it. In IaaS, the group should get ready for application advancement and arrangement on indicated cloud stage and foundation.

3) Design versus Cloud Design: The improvement group change the prerequisite accumulated from the past stage into a plan which will the coders to effortlessly change outline into code. The outline is to be made remembering the plan objectives of the web applications [10-11], a point by point configuration is made which incorporate between confront subtle elements all things considered, input/yield outline. In cloud outline the CSP is worried about planning a suitable strategy for stack adjusting and execution outline. On the off chance that appropriate load adjusting isn't done it might bring about low execution.
4) Development versus Implementation: being developed stage the development group is centered around executing the outline demonstrate into code. The code composed should interoperable and versatile. The coupling between the module ought to be low and union inside the module ought to be high. The cloud specialist organization execute the code got by the advancement group onto web benefit. Any mistake found amid execution of use ought to be accounted for back to the improvement group for redress.
5) Testing: Testing is a standout amongst the most essential stage in applica-tion's lifecycle, any undetected mistake can bring about disappointment of the web application. The product improvement group needs to do broad testing of utilization to make it blunder free. Unit testing ought to be done on the module as when they are prepared.

After every one of the modules are prepared devel-opment group and specialist organization finish the reconciliation test. The improvement group does the test in house where as the specialist organization does it on the cloud foundation. Any blunder found are accounted for back to advancement group for amendment.

In SaaS the group should center around relapse testing the same number of extra administrations are include later as customization. In PaaS the emphasis ought to be on execution testing in light of the fact that for web application the non practical necessity like execution, versatility, accessibility is essential. In IaaS the attention ought to be on recuperation testing as framework is for all intents and purposes possessed and customer has no physical control over the equipment [18].

6) Deployment and Maintenance: In the sending stage the cloud specialist organization takes control as it conveys, convey, bolster and keep up the application. In SaaS the sending is done on the understanding chose the two gatherings, the specialist co-op bolster and keeps up the application. In PaaS, the arrangement is done on the stage and the stage is designed according to the solu-tion supplier. The upkeep and support is chosen in view of the understanding between the specialist organization and arrangement supplier. In IaaS, the sending in view of the framework like number of assets, in particular, processor, working framework, stockpiling limit, observing and metering. Again support and upkeep is done on assention between the arrangement and specialist organization The cloud venture is like the customary undertaking however include three noteworthy gatherings Client, Solution Provider, Service Provider. The communi-cation is one of the key factors in accomplishment of any cloud venture, absence of correspondence can bring about deception of necessity which influence the whole undertaking. The primary concentration in cloud task ought to be on prerequisite and configuration stage. The cost in create ment stage in a cloud venture is not as much as that of customary undertakings. It is obvious from the above representation that the improvement in cloud venture is like customary task however it is very stage and seller particular, and need norms. Different process model can likewise be actualized into cloud ventures

Challenges and open issues

Programming building in the cloud period has made its underlying advances demonstrating potential to give solid usage and execution situations for cloud-based applications. Be that as it may, various critical difficulties should be tended to for this way to deal with be feasible. We list a couple of them:

1) Programming designing perspectives

As it was made apparent from the investigation of the past parts, albeit current answers for cloud-based IDEs give vital new highlights and capacities, they utilize a subset of the highlights show in modern level work area based situations. Along these lines they can address an expansive number of advancement needs, focusing on particular dialects and application requests, yet the integration of the full arrangement of abilities and adaptability existing in their work area partners should be slowly consolidated. Troubleshoot ging and runtime inspecting should be additionally bolstered while a full arrangement of dialects and segments (e.g. databases) should be made accessible to the designer. This requires a significant push to in-corporate usefulness that is accessible to the designers for over 10 years. Additionally, expanded and unpredicted latencies in the advancement procedure that can't be effectively overseen by the envi-ronment itself as they rely upon the system abilities (Jackson et al. 2010) can cause critical issues in the advancement master cess. Fundamentally, in spite of the fact that cloud based conditions can de-liver extraordinary throughput for the gathering and testing of expansive scale extends by utilizing vast ranches of figure hubs, little scale, newborn child activities may experience the ill effects of substantial latencies, a reality that can disappoint designers and debilitate them from moving their workplace to the cloud. Half and half arrangements, where neighborhood situations coordinate with cloud IDEs could give an underlying beginning stage to handle the extreme issues of dormancy.

In addition, current arrangements have incompletely exploited the cloud abilities to use joint effort (Graham 2011) and in this manner programming profitability. Code sharing and forming are total requirements, yet they don't advance efficiency past what is at present upheld in customary conditions, nor do they make utilization of the favorable circumstances gave by cloud situations. Joint effort among improvement groups can be drastically en-hanced by multilayer programming, i.e. by creating parts in various layers in the meantime on a similar task.

Moreover, current cloud-based programming building environments take after the conventional patterns in application advancement. In any case, with question situated methods having achieved a state of depletion, Model-Driven Engineering (MDE) constitutes the most recent change in perspective in programming building, as it raises the level of deliberation past that gave by third era programming dialects. MDE deteriorates framework outline and operational rationale from execution points of interest by using suitable deliberations communicated as models. This decay significantly rearranges delicate product advancement and can computerize generous parts of the procedure.

Cloud advancement has changed in an opposition of mixes with different dialects and programming, leaving outside the fundamental es-sence of the cloud thought, which is the capacity to configuration, actualize, test and convey an application specifically to the cloud. Up to this point there is a propensity to either make a situation with an editorial manager and numerous inserted compilers/mediators, or a domain where the improvement group can transfer their work area made code. In spite of the fact that this possibly a very straightforward arrangement, one can without much of a stretch comprehend that the main decision isn't productive and the second one is simply not cloud execution. A programming domain ready to utilize multi-layer programming and MDE would offer the required levels of deliberation and would just need a straightforward programming dialect to stick things together. This may appear as a radical change, yet we have seen it working in different spaces.

2) Interoperability

The current scene in cloud-based programming advancement platshapes has given adequate answers for exchanging an extensive number of uses to cloud foundations gainfully. Be that as it may, they are intensely in light of specially appointed arrangements, which in a few cases nearly join the created applications to particular improvement situations and facilitating frameworks. Relocating ventures starting with one stage then onto the next or reusing segments be-tween stages is in no way, shape or form clear, as in a few cases some of the consolidated parts are restrictive. Similar holds for segments that need to collaborate with a specific end goal to convey a larger amount item. The circumstances turns out to be considerably all the more difficult in MDE based methodologies that furthermore consolidate more elevated amount of ideas and instruments like models, space particular dialects, and apparatuses for computerized display administration (change, approval, correlation, blending, refactoring and so on.). The utilization of open gauges can give a strong base to the advancement of interoperable mod-ules, while constructing the improvement with respect to open-source segments can limit re-building endeavors.

3) Security

Information security is a standout amongst the most basic issues in cloud based appli-cations (Kaufman 2009; Sangroya et al. 2010; Jensen et al. 2009). The lion's share of clients and ventures are hesitant to confide in delicate information to cloud conditions, and this is the primary explanation behind the de-velopment of private mists. Programming is in no way, shape or form a special case in this run the show. Programming ventures are acknowledged by huge speculations and constitute a basic capital of programming building organizations. Obviously, transferring source code to outside conditions isn't a simple choice, regardless of whether this worries code that will be at long last discharged as open source, yet ensured by one of the accessible licenses.

4) Asset administration

One of the key points of interest of distributed computing is its capacity to use unified assets with a specific end goal to convey fantastic administrations in a "pay as you go" mold. Thusly, it brings down expenses by discharging clients from the weight to contribute on nearby framework, while econ-omy of scale empowers extra advantages, as various little scale and commonly underutilized server farms are supplanted by bigger infrastructures that objective higher asset proficiency. Notwithstanding, this cre-ates another test to oversee assets in these substantial scale envi-ronments that host administrations with various attributes, application requests and measurements for nature of administration (Beloglazov and Buyya 2010; Delimitrou and Kozyrakis 2014; Younge et al. 2010). To get a thought on the difficulties on the difficulties included, we may investigate the power utilization pertinent to server farm task facilitating cloud frameworks: Cloud activity is quick advancing as a standout amongst the most eager for power human exercises engrossing enor-mous and relentlessly expanding measures of vitality, with noteworthy effect on the earth and the nursery discharges (Aravanis et al. 2015). In 2011, server farm's aggregate vitality utilization was around 271 billion kWh: enough to control up all private house-holds in industrialized nations, for example, UK or France, equivalent to the aggregate sum of vitality devoured by Italy (Data Center Dy-namics 2011), or roughly 7 % of the US add up to vitality con-sumption (Index Mundi 2011).

In light of this reality, propelled asset administration motors should be joined at all levels of a cloud environment, from equipment, to cloud administration programming and up to the cloud applications themselves. In a perfect world, every one of these layers ought to work together in an effective method to limit asset utilization (a worry for cloud professional viders), without abusing QoS as communicated in the significant [21].

Administration Level Agreements—SLAs (a worry for cloud clients). In this way, all cloud segments including applications and application advancement conditions would incredibly profit by insightful administration motors that can screen asset utilization, examine the present status, anticipate future requests, settle on more asset effective arrangements and implement/ask for those new designs inside the facilitating cloud condition.

5. Conclusion

This article give a diagram of different lithe procedures with a review on the writing of this philosophy, which iden-tified the key practices of nimble philosophies and nearly dissected the five techniques based on distinguished practices. This paper gives an investigation of one of a kind practices for lithe meth-odologies, a correlation between the procedures, a synopsis of the practices that are basic to light-footed techniques, and gives a comprehension of fitting philosophy mixes.

References

[1] Bandana, H., Harshitha, T.M., September 2013. An effective way to apply agile manifesto in development of cloud application through agile service networks. 3 (9).

[2] Boehm, B., Turner, R., 2005. Management challenges to implementing agile processes in traditional development organizations. Softw. IEEE 22 (5), 30e39.

[3] Bogdan, Ghilic-Micu, Marian, Stoica, Cristian R azvan, Uscatu, 2014. Cloud computing and agile organization development, 18 (4).

[4] Cao, L., Balasubramaniam, R., Kannan, M., Xu, P., 2009. A framework for adapting agile development methodologies. Eur. J. Inf. Syst. 18, 332e343.

[5] Cho, J., 2008. Issues and challenges of agile software development with Scrum. Issues Inf. Syst. IX (2).

[6] Gangadhar, P.V.S.S., Shrivastava, A.K., Shukla, Ragini, Apr-2015. To implement cloud computing by using agile methodology in Indian E-Governance. 02 (01).

[7] Hasaba, Seyed Amin Mousavi, Faraahi, Ahmad, 2014. An overview of applying cloud computing in addressing agile global software development challenges. 40 (1).

[8] Hneif, M., Ow, S.H., 2009. Review of agile methodologies in software development. Int. J. Res. Rev. Appl. Sci. 1 (1).

[9] Hochmuller, E., 2011. The requirement engineer as a Liaison Officer in agile software development. In: Proceeding of AREW.

[10] Jain, Abhishek, Rani, Reena, 2011. Analytical Study of Agile Methodology with Cloud Computing. RTMC.

[11] Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I., 2012. The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. Software Practice Exp. 42 (4), 447e465.

[12] Khan, Ahmad, Shaikh, Parez, Dhembre, Chetan,Gawali, Sushant,November 2011. Cloud services for collaborative web based project management system 8 (6), No two.

[13] Kumar, Gaurav, Bhatia, Pradeep Kumar, August 2012. Impact of agile methodology on software development process.2 (4).

[14] Mani, Pavithra, Deebitha, S., March 2014. Analysis of agile software development utilising cloud-computing capabilities.3 (10), ISSN: 2278e3075.

[15] Mazni, O., Sharifah-Lailee, S., Azman, Y., 2011. The impact of agile approach on software engineering teams. Am. J. Econ. Bus. Adm. 3 (1), 12e17.

[16] Misra, Subhas C., Kumar, Uma, Kumar, Vinod, Grant, Gerald, 2006. The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development Organizations. IEEE.

[17] Moniruzzaman, A.B.M., Hossain, Syed Akhter, Jul 12, 2013. Comparative Study on Agile software development methodologies.

[18] Dalal S., Dahiya N., Jaglan V. (2018) Efficient Tuning of COCOMO Model Cost Drivers Through Generalized Reduced Gradient (GRG) Nonlinear Optimization with Best-Fit Analysis. In: Saeed K., Chaki N., Pati B., Bakshi S., Mohapatra D. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 563. Springer, Singapore

[19] Nawrocki, Jerzy, Jasiski, Michal,Water, Bartosz,Wojciechowski, Adam, 2002. Extreme Programming Modified Embrace Requirements Engineering Practices. IEEE.

[20] Nazir, Ambreen, Raana, Ayesha, Fahad Khan, Muhammad. Cloud computing ensembles agile development methodologies for successful project development, doi: 10.5815/ijmecs.2013.11.04.

[21] Seth B., Dalal S. (2018) Analytical Assessment of Security Mechanisms of Cloud Environment. In: Saeed K., Chaki N., Pati B., Bakshi S., Mohapatra D. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 563. Springer, Singapore.