CLOUD'S LEAN EXSCRUM (CLXS): A METHODOLOGY FOR LARGE PROJECTS IN CLOUD COMPUTING ENVIRONMENT

Anum Liaquat[1], Ayesha Siddiqui[1], Muhammad Shafique[2]
[1,2]Department of Computer Science and Engineering, University of Engineering & Technology, Lahore, Pakistan
Corresponding Author's Email. anumliaqat83@gmail.com

ABSTRACT*—Traditional software development has always been expensive, slow, less flexible and highly unreliable. Cloud computing has emerged over the last decade to address these problems. Now, most software industries are switching their development to cloud infrastructures. As software companies focus on building high quality software, so there is a need to update software development models according to cloud platforms. We developed a software model compatible in cloud computing. We preferred agile for getting higher customer satisfaction as agile methodology introduces flexibility. Unfortunately classical agile methods are not suitable for large projects. But one problem arises again that is mismatch risk of services provided by cloud service provider and development house or the organization which is using cloud infrastructure. So spiral model is the only classical model which has risk management phase description explicitly. We are introducing a risk management phase in our hybrid proposed methodology to mitigate any type of mismatch risk of service. In fact this concept is borrowed from spiral model and it is helpful to eliminate deficiency of proposed methodology. The proposed hybrid agile methodology in this paper addresses this issue and targets to large projects.*

Keywords— agile methodology, cloud computing, eXscrum, XP, lean software development, large projects, mismatch risks

I. INTRODUCTION

Traditional software development is now anachronous due to their immutable nature and they do not provide quality product on time. Organizations need an ability to adapt unpredictable change in more rapid and appropriate ways than their competitors [13]. A new emerging trend in software industry is agile development. Agile development practices have resolved many problems e.g. low quality product, accommodation of changes during development and increased focus on documentation, by supporting customer interaction and change amalgamation. Some dominant agile methods include Extreme Programming (XP), Scrum, Crystal Methods, Feature Driven Development (FDD) and Test Driven Development (TDD).

One of the ubiquitous computing that needs attention these days is cloud computing. Now world of traditional software development is moving towards the path of cloud computing which has promising power of innovation. It compromises on the concept of parallel computing, distributed computing and grid computing. Cloud computing refers to the use of centralized, highly available virtualized computing resources (hardware and software) that may be accessed as a service through network, mainly internet. User only needs little knowledge of system's infrastructure, installation, configuration and maintenance etc. National Institute of Standards and Technology (NIST) defines cloud computing as: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]". Large scale problems can easily be solved through cloud computing as it has the power of supercomputers [2]. It provides virtualization, scalability, high performance and support with low costs. There are three types of clouds in the market namely public, private and hybrid clouds.

Projects differ in size, complexity and mission criticality. Cloud computing mostly deals with large projects and agile methods cope with small and medium projects. Agile methods lack the customization for large sized projects; also, they are

not reusable and sometimes need excessive maintenance. Keeping in consideration both technologies many new agile methods have been proposed e.g. Extended XP and eXscrum that can be suitable for cloud computing environment. Our main focus is to design a methodology that best suits for complex project development in cloud environment. Agile methods are selected to use in cloud environment to meet changing requirements of technology because it introduces feedback mechanism. Agile with cloud computing provide marvelous results as user response is very quick and distribution cost no longer exists.

A brief overview of agile methodology and cloud computing is given in Section II and Section III of this paper respectively. Section IV covers literature survey showing that cloud environment supports agile methodologies well for software developments. In Section V we devised an agile methodology for cloud computing that deals larger projects. Discussion with mathematical proof about the proposed methodology is given in Section VI. Finally, Section VII and Section VIII are about future recommendations and conclusion.

II. AGILE METHODOLOGY

A. *Agile Manifesto*

Agility means just enough. Agile is n iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders. All agile models follow agile manifesto proposed [23] by a team of innovative developers. Some of the points of agile manifesto are given as under:

- Customer satisfaction through timely collaboration
- Simplicity
- Good design and fast coding
- Welcome change even at the end of the project
- Delivery of working product

- Performance check of team on regular intervals
- Providing necessary facilities needed by the developers
- Face-to-face discussion
- Speed of work and collaboration should be constant
- Working software
- Inter-communication between teams.

B. *Agile Models*

There are many agile methods in use today. Some of the well-known agile models are Extreme Programming, Scrum, Rational Unified Process, Crystal, and Feature Driven Development etc. Detail of some of them is given below:

1) *Extreme Programming (XP):* It enhances quality of software through frequent releases and customer collaboration. It follows pair programming technique.
2) *Scrum:* uses project management techniques like meetings, time-boxed sprints and team management.
3) *Test Driven Development (TDD):*Two times effort needed in TDD. First test cases are written for new features and then code is implemented, afterwards that code is again tested against those test cases. Therefore, TDD is best suited in the cases where team members are real experts to foresee the requirements. Modifications in the code are real headache later in the project. Furthermore, user interface testing cannot be done by TDD.
4) *Crystal:* is a family of different agile models like crystal clear, crystal orange etc. It is used to implement unique functionality through some terms, conditions and specific techniques
5) *Feature Driven Development (FDD):* does features based development. Focuses on single feature given and deliver it within a specific time span.
6) *eXscrum:*It is a proposed model combined by introducing XP in scrum development cycle. It provides us an efficient methodology by eliminating the flaws of both models.

C. *Traditional verses Agile*

Some of the main differences between agile and traditional software development methods are given below

- Traditional software development does not allow immediate change while agility is all about accommodating changes.
- In agile methods exhaustive documentation is discouraged but traditional development is based on documentation.
- Traditional development is reusable but in agile methods all code has to be written from scratch.
- All the requirements are gathered before starting the project in traditional development but in agile methods requirements are continuously changing.

D. *Benefits of Agile Models*

There are many benefits of using agile methods over traditional software development due to which it is getting popular day by day. Few worth listing points are as follows:

- High Quality Product
- Greater value to customer interaction due to which requirements are accommodated as soon as they are encountered.
- Prototyping provides an overview of the software by which customer knows about how end product will look like.
- Less documentation saves time
- Cost is also affected because resources are not wasted on unnecessary tasks.

E. *Considerations for applying agile in a project*

Before applying agile methods to develop any project, a project manager has to consider following points:

- Whether agile method will be supported in the provided environment or not?
- Will all team members of a project remain same throughout the project?
- Organization which is going to use this method accepts the iterative nature of the projects?
- Will it be possible for the organization to deliver frequent incremental prototypes?
- Customer interaction during the project on short notice will be possible or not?

If all an organization fulfills all the requirements listed above then it is ready to implement agile development methods for their project.

III. CLOUD COMPUTING

There is no proper definition of cloud computing. It is the new technology that is going to exist everywhere. Cloud computing is actually the efficient use of web as a resource space which has the computing power (the cloud). All services of the clouds are provided by internet. Cloud computing system consists of two parts: front end and back end. Front end comprises of what the user sees and the application which make cloud's services available to user while back end is actually the cloud itself which rely on strong computers, huge servers and large data storage units.

A. *Architecture of Cloud Computing*

Cloud's computing architecture is divided into five layers: client, application, platform, infrastructure and server.

1) ***Client Layer:*** It consists of client's computer which is configured to access cloud's services.
2) ***Application Layer:*** This layer delivers software as a service through internet thus eliminating the need to install it on every client's pc.
3) ***Platform Layer:*** It provides computing platform as a service to developers by using cloud's infrastructure. Thus, developers don't need to install software and make environments for development, deployment and testing.
4) ***Infrastructure Layer:*** It consists of data centers, network resources, computing machines etc. which are delivered as a service to clients on pay per use basis. Hence, customer has much faster system with less cost.
5) ***Server Layer:*** Servers consist of computer hardware/ software that make sure the delivery of above mentioned services of all layers.

Layers in a sequence are shown in fig.1

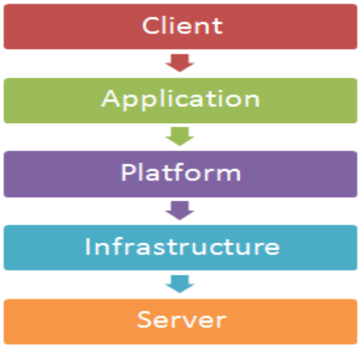

Fig. 1 Cloud computing layers

B. ***Characteristics of Cloud Computing:*** Some characteristics of cloud computing are as follows:

- Less maintenance cost
- Ultimate performance
- Virtualized resources
- 24 hours business continuation
- Efficient usage of infrastructure
- Easy disaster recovery
- Maximum support of scalability and flexibility
- Pay per use facility
- High security and reliability than traditional systems

C. ***Deployment of Cloud Computing:*** Deployment of cloud computing can be categorized into private, public and hybrid clouds.

1) ***Private Clouds:*** As the word suggests, private clouds are deployed by organization itself and are accessed only within.

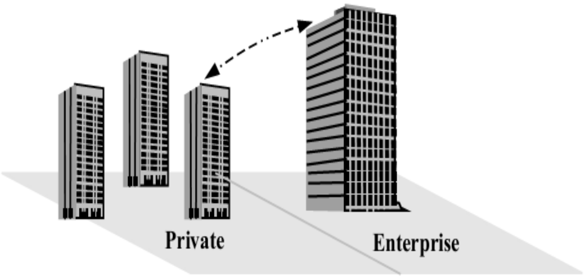

Fig. 2 Private cloud

2) ***Public Clouds:*** These are less secure clouds as many organizations can use the same resources but cheaper ones.

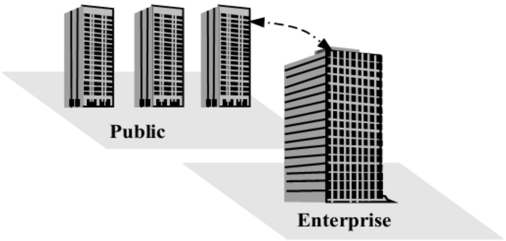

Fig. 3 Public cloud

3) ***Hybrid Clouds:*** If an organization allocates highly critical work to private clouds an also uses public clouds for less critical work then the concept of hybrid clouds arise.

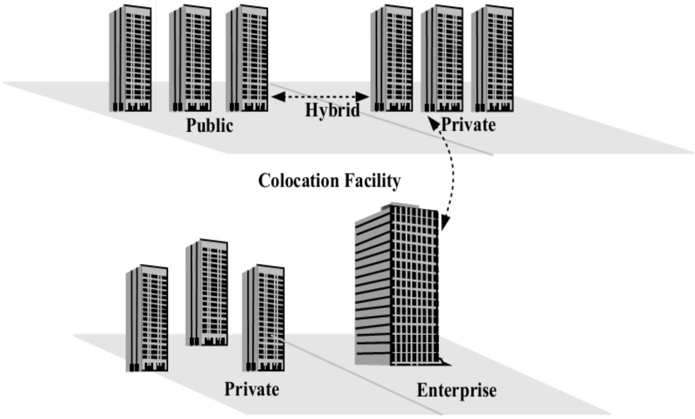

Fig. 4 Hybrid cloud

D. ***Traditional Computing Vs. Cloud Computing*:**Some differences between traditional computing and cloud computing are as follows:

- Costs are high in traditional computing while cloud computing offers low costs in every respect.
- Traditional computing is slow but cloud computing is very fast as it has much powerful computing resources.
- Traditional computing is not scalable according to project's size. On the other hand, cloud computing is flexible and scalable.
- Traditional computing is not reliable and secure while cloud computing is just its opposite case.
- Traditional computing does not guarantee availability of computing resources all the time as errors/failures may occur. Cloud computing makes sure that resources are highly available even they are subject to crashes.

IV. RELATED WORK

Agile methodology is the only one which facilitates rapid change acceptance and accommodation. It supports on-premises development more than distributed one as communication costs are low [14]. Some popular agile methods include Extreme Programming (XP), Scrum, Crystal Methods, Feature Driven Development (FDD) and Test Driven Development (TDD).Agile scrum method is adopted in large projects where there is a lot of uncertainty. Face to face communication is basic need for agile methods but Distributed Scrum and Distributed XP have been developed. To introduce agility in distributed project unofficial meetings were being held and there was also a centralized version control server which can be accessed by all the team members to check their codes through VPN [18].

Rizwan proposed an extended XP model which performs cyclic steps and remains evolutionary till the completeness of software. This model was applied to three cases studies including small, medium and large size projects and results show an increased quality as compared to existing XP due to changes introduced in the lifecycle of model[16].

Scrum is the best method because it is scalable from large to small projects [15].Scrum is more high level, focusing on the management of the project (e.g. the requirements or features are managed) rather than specifying or defining engineering practice such as pair programming or test driven development [17].

Qureshi *et al* proposed eXscrum method by introducing XP method in scrum coding phase [19].

Large projects can be done by creating scrums of scrums but it is necessary to have the same vision of project to all team members. Obstruction in doing large projects with scrum includes limitations of distributed developments, large teams, safety critical systems and developing large complex software due to issues in face to face communication and location differences. When we set some goals to achieve then objectives must be specific and measurable. Scrum can be used in large project by utilizing a goal oriented approach by improving requirements [20].

Cloud computing is a new field in computer sciences which is gaining popularity these days. It is an emanating concept of computing in which virtualized resources are retrieved as a service through web browsers or web based applications with internet as a connection between server and clients [5]. Tools like cloudo.com [7] and eyeos.com [8] can be used to access services of cloud environment. Data and software applications are placed on servers. Third parties offer cloud services to enterprises and charge costs as pay per use. There are many advantages at the user end like reliable system architecture, huge memory, large computing resources such as supercomputers, significant cost reduction as the overhead of maintenance of resources(both hardware and software) is not there, flexibility and scalability, interoperability, high availability of services, data recovery etc. Jadeja and Modi discussed cloud computing concept in detail with its characteristics, architecture and challenges [3]. Similarly Mollah et al discussed cloud computing layers, service models, types of deployed clouds and open source cloud platforms with comparisons in great depth [4].

Peng *et al* proposed three layered architecture of cloud environment. First layer being Infrastructure as a service (IAAS), second Platform as a service (PAAS) and third layer categorized is Software as a service (SAAS) .They further devised a way to test cloud computing applications. In their viewpoint traditional testing do not provide large scale computing power, unaffordable to purchase resources for test environments, involve complex configuration for testing, provides difficulty in making test cases. Cloud testing on the other hand provides rapid response, easier expansion, and improved efficiency and above all it is affordable [2]. Jadeja and Modi further introduce two more layers in architecture of cloud that is client layer above SAAS layer and server layer below IAAS layer [3].

Kalagiakos *et al* introduced the idea that cloud platform can be used for distant learning effectively. The user has availability of learning objects around the clock while he has to worry less about installing environments compatible for running those objects. Also user can access learning material from anywhere [6].

Hashmi *et al* identifies that cloud is the best technology for global software development. GSD provides reduced time to market delivery, access to large skill pool, 24 hour development and high economic benefits. The challenges introduced by GSD can be effectively addressed by cloud computing [10].

Jain and Rani gave an idea of integrating agile methods with cloud computing as distribution cost will decrease and feedback of clients will be more prompt [21].

Cocco *et al* made system dynamic model that can be used as a tool for the comparison of global software development on cloud environment and on traditional systems. This model can be used to estimate costs of cloud computing which generally shows low costs. They prefer to use agile methodology for both developments [11].

Patidar *et al* highlighted the problem of software development in cloud computing. He suggested that for getting high quality softwares, software development models must be reviewed and updated by keeping cloud providers in mind. Cloud providers should be considered at each and every step of decision making [12].

We found just a vague idea of agile methodology concept introduced in cloud computing. Many researchers said that no doubt, agile methods have many benefits but for larger projects it is still not a good practice. Furthermore, obstructions in doing large projects with agile scrum include limitations of distributed developments, large teams due to issues in face to face communication and location differences. eXscrum was proposed for large projects, but the procedure to introduce it in clouds was not mentioned.

The idea of GSD using cloud computing is introduced in the literature, but there is a need to mention the steps clearly that developers may follow in a cloud environment for getting engineered software product. Similarly we found a possibility of combining two technologies that is agile methodology and cloud computing to get maximum benefits from the two but the issue still remains that how these technologies can be run simultaneously. Further, we encountered a system dynamic model for GSD on cloud environment and on traditional systems. The model just addressed small project development using agile methods. There is a question mark for developing large projects in cloud environment. Literature also suggested that all traditional software models must be revised keeping cloud providers in mind at each and every step in order to avoid any failure, but there are no such appropriate software development methods formulated up til now. Also, there is no proper agile method that can be followed by software developers for dealing with large projects. This thing encourages us to design a methodology which has agility for complex large projects.

V. PROPOSED AGILE METHODOLOGY FOR CLOUD COMPUTING

We proposed a methodology which is based on lean principles. Lean principles focus only on a single piece or feature. It eliminates all unnecessary code, superfluous functionality, ambiguous requirements and communication delays. Using these principles we utilized eXscrum method to be used with cloud computing. EXscrum combines the benefits of both extreme programming method and scrum retiring their flaws.

In sprint cycle of eXscrum, we have to consider cloud providers at four steps which are plan, design, coding and testing as cloud providers now play major roles. In plan stage cloud providers should be kept in mind as how much they are giving resources and major planning decisions should be made accordingly. Design issues should be resolved according to cloud providers as different platforms may affect the project whenever switching between platforms occur. Coding should be done by proper standards which are negotiated between organization and cloud providers. Test cases should be made according to cloud environment which differ from traditional environment. So, how we used eXscrum in cloud computing is shown in fig.5.

In our proposed CLXS model, first requirement gathering is carried out by an organization. Negotiation of coding standards and software development model to be followed is made between cloud developers and local developers so that uniformity can be maintained. Expert personnel will sort the requirements into functional and non-functional. After distinguishing the requirements, project manager will divide functional requirement to organizational cloud developers and non-functional requirements to developers provided by cloud providers as shown in fig.6.

Hence two backlogs are maintained:

1. Functional requirements backlogs
2. Non-functional requirements backlogs

Non-functional requirements are completely or partially handed over to cloud services provider division of non-functional requirement is based upon its nature which is identified by Project manager, after its identify its nature it is defined who is responsible to fulfill it either development house or cloud service provider.

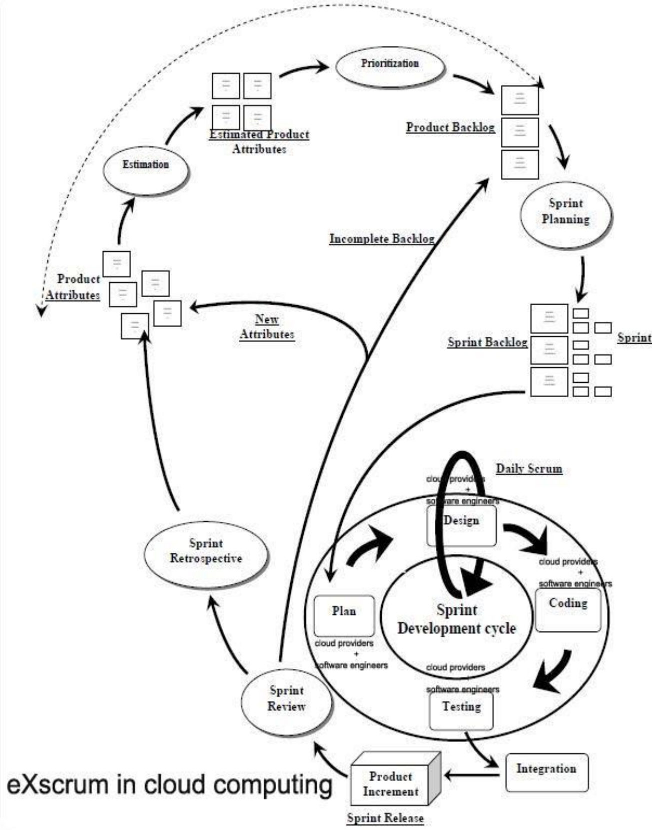

Fig. 5 eXscrum in cloud computing

Suppose client need about development technology e.g. if client asks to develop a web site, then there are many technologies like PHP, ASP.Net Ruby on Rails to develop it but if client explicitly mention the technology then this requirement is put into Non-functional requirement backlog and handed over to cloud service provider so they set platform layer accordingly because platform layer it belong to cloud architecture as shown in fig.1.

An administrator team will communicate with both types of providers to have an idea of the overall working and makes sure that the development on both sides is going in parallel particular to the same task. According to eXscrum a time boxed sprint will be produced after certain time period. Sprints from both organizational developers and cloud developers will next be taken and integrated by an integration team. An integration testing will be done to make sure the proper functioning of the product. Now the product is ready for the release. The release is given to the customer to approve and give opinion for further changes. New changes from the customer feedback are incorporated in the cycle before the beginning of next sprint cycle. The cycle keeps continuing until the final product delivery. We named our proposed methodology as cloud’s lean eXscrum (CLXS).

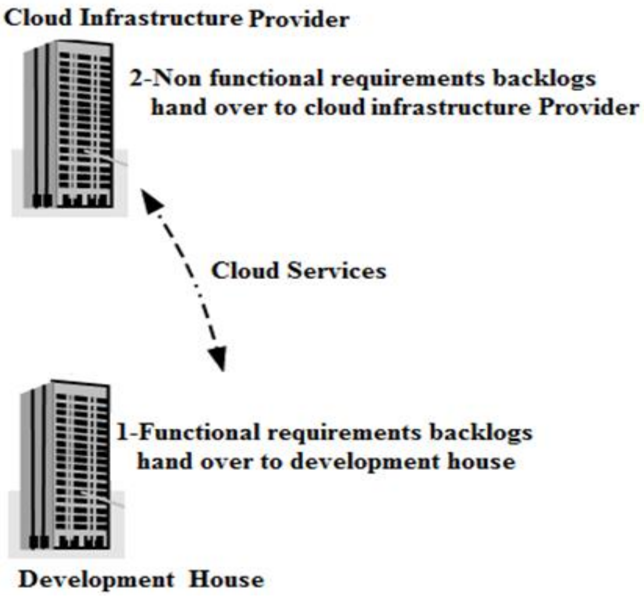
Cloud Infrastructure Provider
2-Non functional requirements backlogs hand over to cloud infrastructure Provider
Cloud Services
1-Functional requirements backlogs hand over to development house
Development House

Fig.6 Division of Requirement logs

We know that agile is very powerful in accepting rapid changes, but still there is problem that it do not have any explicit phase to manage risk because when you are working on big project and any other organization like cloud service provider then its bit tricky to keep things synchronize and smooth to avoid nay miss match of required services for any sprint. Suppose we are working on a legacy system to up gradation and adding more use cases to extend it so it could meet the current requirements of the client. Legacy system is built in older version of .net and SQL server. Before start of development cloud service provider has set the platform layer according to our requirement. Some requirements of the client has been fulfil in previous sprints and now a new sprint is going to be start .In Sprint meeting client raise the requirement to develop new functionality in newer version of SQL server and .net framework. Now miss match risk is ensue and as a result, until the cloud service provider set platform layer to provider services of new version we have to wait and halt development which not cost affective. To cater this problem fig 5 is slightly modified and new phase is introduced in sprint which is risk management phase. In this phase clients new requirements for next sprint is asked and communicated to cloud service provider so when development team from development house start development for new sprint they find development environment ready for development. This concept is borrowed from spiral model where risk management phase is explicitly defined. Fig 6 is showing the modification having a new step of Risk analysis and mitigation.

VI. CASE STUDY

As mentioned above that there are also other agile methods like crystal, FDD and TDD. Crystal family emphasis on the fact that project development should be crystal i.e. all the programmers should be at the same place.In cloud computing this fact restricts us not to use crystal methods.

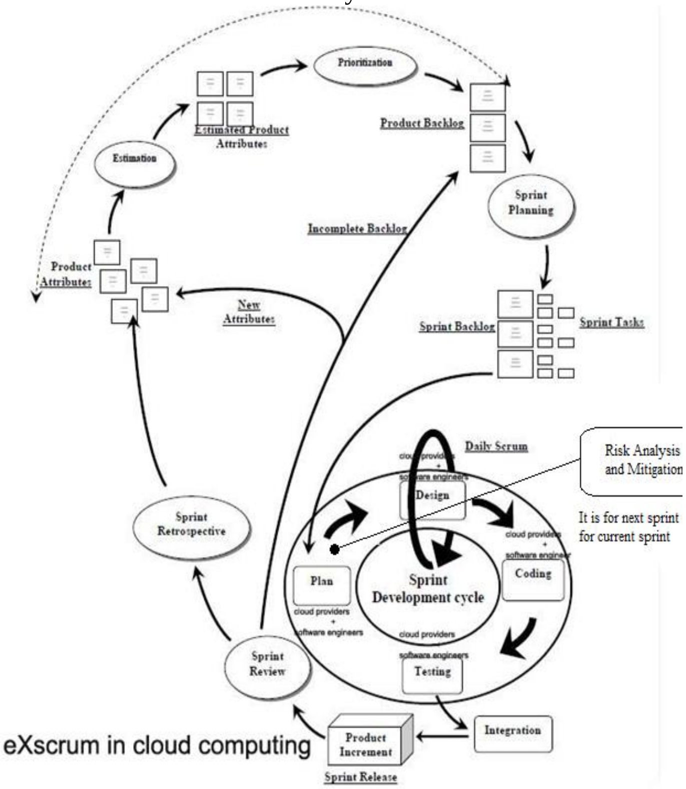
Prioritization
Estimated Product Attributes
Product Backlog
Estimation
Sprint Planning
Incomplete Backlog
Product Attributes
New Attributes
Sprint Backlog
Sprint Tasks
Daily Scrum
Risk Analysis and Mitigation
It is for next sprint for current sprint
cloud providers + software engineers
Design
cloud providers + software engineer
Coding
Sprint Retrospective
Plan
Sprint Development cycle
cloud providers + software engineers
cloud providers software engineers
Testing
Sprint Review
Integration
Product Increment
eXscrum in cloud computing
Sprint Release

Fig. 7 eXscrum in cloud computing with risk analysis and mitigation

FDD method reflects traditional software development as it has detailed designing and analysing capabilities. Therefore, it takes more time than other agile methods. Major disadvantage is that we cannot develop software which is not feature driven. So, the choice of FDD to implement with cloud computing is also eliminated. In TDD test cases are prewritten and are checked after implementation of code. It is best suited in the cases where team members are experts enough to foresee the requirements. Modifications in the code are real headache later in the project. Let's discuss a case study to validate the efficiency of our proposed methodology. We are using two ways to manage the problem described in case study. First is based upon our current methodology and second is based on our newly proposed methodology.

Let's have a look on following case study:

Our organization decides to upgrade a legacy system based on Vb.net framework version 2 and Sql server 2000 with some additional functionality. Organization is signed an agreement with client and organization started project on agile methodology. We put development platform requirement in front of cloud service provider and signed an agreement with them they provide us required platform after two sprints of development client raise a Non-functional requirement to build new functionality on .net framework 4 and SQL server 2008.When this thing is communicated to our cloud service provider they charged us more and we have to revise our agreement with them and parallel we have to wait for new plat form setup, It increases our development cost in terms of time. It's because of deficiency to cater miss match risk of development platform in current eXscrum methodology. Because development platform is set in start of the project and so on in case any such sudden addition in client requirement rises then sprint length may extend.

A. *Steps Current eXscrum methodology :*

After analysing the client current requirement regarding the technology and development environment is communicated to cloud service provider and they set the platform layer accordingly.

Suppose client requirements are divided into 4 sprints

Each sprint has following steps

1- Plan→Meeting in which client, project manager, Cloud service provider are involved to select and add functionality to develop for current sprint
2- Design→Functionalities are selected from product functional backlogs to develop.
3- Coding→Work is assign to developers in daily scrum meeting
4- Testing→Testing of work done in current sprint

Two sprints have passed by following above mention steps and in the start of third sprint client raise a non-functional requirement to upgrade the version of platform. Now development of third sprint is dependent upon the platform requirement. Now a developer has to wait for time ***Tw*** until service providers set cloud environment for development. Because development is directly dependent on development environment

Time Variables & Time calculation:

In following equations n>0 (n=1, 2,3,4,5...)

- ***tc***→*Time for development of single functionality*

$$\sum_{i=1}^{n} tci = tc1 + tc2 + tc3 + \cdots + tcn$$

- ***Tc***→*Total time to develop all selected functionalities in a sprint*

$$Tc = \sum_{i=1}^{n} tci$$

- ***tp***→*Time for setting of one cloud service as a development platform*

$$\sum_{i=1}^{n} tpi = tp1 + tp2 + tp3 + \cdots + tpn$$

- ***Tp*** → *Total time to set cloud environment for development*

$$Tp = \sum_{i=1}^{n} tpi$$

- ***tw*** → *Waiting time for developer to wait for one cloud service required as a development platform*

$$\sum_{i=1}^{n} tw = tw1 + tw2 + tw3 + \cdots + twn$$

- ***Tw*** → *Total waiting time for developer to wait for one cloud service required as a development platform*

$$Tw = \sum_{i=1}^{n} tw$$

$$Tp \geq Tw > 0$$

- ***Ts*** → *Estimated development time for each sprint in current methodology*

$Ts = Tw + Tc$ ……….Eq(1)

B. *Steps proposed eXscrum methodology :*

In our proposed methodology we keep cloud service provider one step ahead and in each sprint any type of miss match risk of cloud platform layer for next sprint is analysed and mitigate with client involvement so that when developer move to work on next sprint then they found a cloud platform layer ready for development so they do not have to wait ***Tw*** to start development.

Each sprint has the following steps in proposed methodology

1- Plan→Meeting in which client, project manager, Cloud service provider are involved to select and add functionality to develop for current sprint

2- Risk Analysis and mitigation for the next sprint→It is managed with the collaboration of client, project manager, Cloud service provider

3- Design→Functionalities are selected from product functional backlogs to develop.

4- Coding→Work is assign to developers in daily scrum meeting

5- Testing →Testing of work done in current sprint

These strategies keeps both end parallel and do not let create dependency in sequence. Hence there is not ***Tw*** therefore Equation .1 is modified to Equation .2 for proposed methodology.

- ***Tm*** → *Estimated development time for each sprint in proposed methodology*

$Tm = Tc$ ……….Eq(2)

From Eq(1) and Eq(2)

$Tm = Ts - Tw$ ……….Eq(3)

From Eq(3)

$$Ts > Tm > 0$$

Our proposed methodology is efficient as compared to existing agile methodologies for cloud computing and it is mathematically proved by Equation (3).Efficiency is achieved in three terms that is on time product delivery by dividing work load between organizational and cloud development teams, fast product development and instant user feedback. We introduced eXscrum model to make sure optimal development by using lean principles. In this way, we introduced engineering practices as well as project monitoring in order to ensure the development of client's satisfactory large projects on cloud environment.

VII. FUTURE RECOMMENDATIONS

In our proposed methodology traditionally a projects management team of an organization divides project between an organization's developers and for cloud developers. In future the strategy could be changed by giving the whole project to be developed to cloud providers' coders. An organization's administration team can maintain a check and balance on development made by cloud developers. This creates an opportunity of free lancing and developers now can work at home easily. The idea if implemented will also provide a room for cloud providers to grow their business at high scales by charging extra amount for providing development teams.

VIII. CONCLUSION

Agile methods replaced traditional software development due to their adaptable nature. Agile utilizes engineering practices to gain high customer satisfaction. It implements small to average scale projects on premises. A world needs products in short time due to which high speed computing is the focus of today's computing paradigms. These days cloud computing comes at the top of fastest computing. Also, customers need costs effective resources to get more profit. As projects do not conform in their size and complexity, so there must be some software model which should deal with project diversity. To take advantage from agile compliance to changing requirements and fast computing, on clouds we proposed a new method. This new methodology makes use of XP, scrum, lean principles and cloud computing environment. Proposed method is highly scalable from small to large projects.

ACKNOWLEDGEMENT

We are in-debt of our teachers for their valuable guidance and support. National Library of Engineering Sciences also played a great role in order to complete this research. Lastly, we all are really indebted to University of Engineering & Technology, Lahore because it has facilitated us to access marvellous research content which is subscription based.

REFERENCES

[1] J. National Institute of Standards and Technology. "Computer Security Resource Centre" Internet: www.csrc.nist.gov , September, 2011.

[2] Zhenlong, P., Zhonghui, O.Y., Youlan, H.,"The Application and Development of Software Testing in Cloud Computing Environment," *Computer Science & Service System (CSSS), 2012 International Conference on* , vol., no., pp.450,454, 11-13 Aug. 2012

[3] Jadeja, Y.; Modi, K., "Cloud computing - concepts, architecture and challenges," *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on* , vol., no., pp.877,880, 21-22 March 2012

[4] Mollah, M.B.; Islam, K.R.; Islam, S.S., "Next generation of computing through cloud computing technology," *Electrical & Computer Engineering (CCECE), 2012*

25th IEEE Canadian Conference on , vol., no., pp.1,6, April 29 2012-May 2 2012

[5] "Cloud Computing."Internet:www.investopedia.com

[6] Kalagiakos, P.; Karampelas, P., "Cloud Computing learning," *Application of Information and Communication Technologies (AICT), 2011 5th International Conference on* , vol., no., pp.1,4, 12-14 Oct. 2011

[7] Internet: www.cloudo.com

[8] Internet: www.eyeos.com

[9] Hossain, E.; Babar, M.A.; Hye-young Paik; Verner, J., "Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework," *Software Engineering Conference, 2009. APSEC '09. Asia-Pacific* , vol., no., pp.457,464, 1-3 Dec. 2009

[10] Hashmi, S.I.; Clerc, V.; Razavian, M.; Manteli, C.; Tamburri, D.A.; Lago, P.; di Nitto, E.; Richardson, I., "Using the Cloud to Facilitate Global Software Development Challenges," *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on* , vol., no., pp.70,77, 15-18 Aug. 2011

[11] Cocco, L.; Mannaro, K.; Concas, G., "A Model for Global Software Development with Cloud Platforms," *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on* , vol., no., pp.446,452, 5-8 Sept. 2012

[12] Patidar, S.; Rane, D.; Jain, P., "Challenges of software development on cloud platform," *Information and Communication Technologies (WICT), 2011 World Congress on* , vol., no., pp.1009,1013, 11-14 Dec. 2011

[13] Bohem, B., "Making a difference in the software century", *IEEE computer society*, 2008

[14] Gelperin, D., "Exploring Agile",*Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral*,pp1-3, 2008

[15] Ahmed, A.; Ahmad, S.; Ehsan, N.; Mirza, E.; Sarwar, S.Z., "Agile software development: Impact on productivity and quality," *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on* , vol., no., pp.287,291, 2-5 June 2010

[16] Rizwan Jameel Qureshi, M., "Agile software development methodology for medium and large projects," *Software, IET* , vol.6, no.4, pp.358,363, August 2012

[17] Bilsel, T., "Scrum vs XP – What is the difference?" Internet: www.tatayfunbilsel.com , 4th April, 2011

[18] Paasivaara, M.; Durasiewicz, S.; Lassenius, C., "Distributed Agile Development: Using Scrum in a Large Project," *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on* , vol., no., pp.87,95, 17-20 Aug. 2008

[19] Qureshi, M. Rizwan Jameel. "Empirical Evaluation of the Proposed eXSCRUM Model-Results of a Case Study." *International Journal of Computer Science 8*, 14th April, 2010

[20] Ktata, Q.,Lévesque, G., "Agile development: issues and avenues requiring a substantial enhancement of the business perspective in large projects"*C3S2E '09 Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering*,pp.59-66, 2009

[21] Jain, A., Rani, R., "Analytical Study of Agile Methodology with Cloud Computing." *IJCA Proceedings on National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing*,13-14, June 2012.

[22] "Agile Methodology meets cloud computing for extra ordinary results" Internet: www.salesforce.com

[23] Beck, K., "Manifesto for Agile Software Development " Internet: *http://agilemanifesto.org/* 2001.