I.J.Modern Education and Computer Science, **2013, 11, 28-35**
Published Online November 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2013.11.04

Cloud Computing ensembles Agile Development Methodologies for Successful Project Development

Ambreen Nazir
Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan
Email: ambreen_nazir_ms@yahoo.com

Ayesha Raana
Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan
Email: ayesha.raana@yahoo.com

Muhammad Fahad Khan
Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan
Email: fahad.khan@uettaxila.edu.pk

Abstract—In today's IT world combination of AD (Agile Development) and CC (Cloud Computing) is a good recipe for the user needs fulfillment in efficient manners. This combination brings superiority for both worlds, Agile and Cloud. CC opportunities are optimized by AD processes for iterative software releases and getting more frequent user feedback while reducing cost. This paper analyzes the AM (Agile Methodology) processes and its benefits, issues with CC. ACD (Agile Cloud Development) approach helps a lot in overwhelming the challenges of both practices, encourages higher degree of innovation, and allows finding discovery and validation in requirements.

Index Terms—Agile, ACD (Agile Cloud Development), AD (Agile Development), AM (Agile Methodology), Cloud, CC (Cloud Computing), Feedback, Requirements.

I. INTRODUCTION

Now a day's CC [1] is flourishing in the IT world and it permits organizations to use shared data center space for running their applications. By removing unwieldy processes linked with software development, testing, installation and failures, the Cloud Technology fetch in competence and effectiveness. No hardware and software necessary, easy integration, quick deployment, extremely customizable environment, optimal consumption of resources are the foremost advantages of CC. Relying on these advantages CC will surely become preferable medium for application development and delivery.

AM [2], an approach of project management, is used for Software Development. The center of attention in AD is following the idea of developing software in small iterations with regular releases, which helps out teams reacting towards the irregularity of software under development. In AD requirements are realized collaboratively throughout the development lifecycle to appraise the proper direction of the project. Extreme inventive people who have implicit the loopholes of software development are using AD methods [3]. The basic aim of AD is to generate a quick response loop among developers and stakeholders.

AD Methodologies and CC both are accompaniment for each other splendidly. ACD [4] is the most modern succession in practical, as when both AD and CC integrate with each other they bring in the investigation of new techniques and tools.

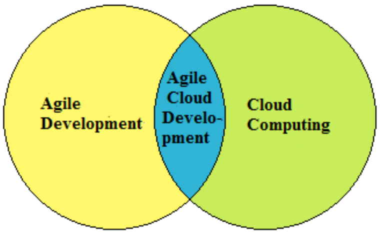

Figure 1. Agile Cloud Development

AD in combination with CC endows an environment which is highly interactive and co-operative. Whenever a new feature for the project is decided by the developers, they can put it in the Cloud as a Cloud Service so they can get stakeholder's feedback instantaneously. Consequently, a prolonged feedback cycle can be eradicated. This significantly shorten the time and extra exertion for the software development, as lessens the probability of misstated or misapprehended requirement thus amplifying end user satisfaction.

ACD approach helps a lot in overwhelming the challenges of both practices, encourages novelty up to a superior degree, and allows finding discovery and validation in requirements. Dwelling in elegant structured Agile Cloud Lifecycle Management leads to distributing AD tools flawlessly with no communication requirements to all the developers and manufacturing supporters throughout the organization. Major benefits ACD includes: permitting organizations to reinforce their IT assortment for better service deliverance, engender speedy user response, boost up business agility, develop understanding among customers and allow organizations to scrutinize quality principles at every slice of the development whilst reducing budget. We accept it a true fact that for all range of industries, ACD is the prospect for worldwide organizations.

This paper comprises of seven sections. In Section II, a critical review of the work done in leveraging AD and CC is given. In Section III, we discussed major challenges of AD and CC. In Section IV, aspects and core ideals are given through which CC enhances and supports AD. In Section V, a Cloud based, Agile Application Development Lifecycle Management is presented in which AD and CC integrates with each other for successful project development. In Section VI, we have highlighted the benefits of ACD for confronting the challenges of Section III. In Section VII, concluding remarks are given.

II. LITERATURE REVIEW

Whenever work on any project started, first we choose the Development methodology that is based upon on suitability of the project and stakeholders.

A. *Why AD?*

During development lifecycle, AD Methodology helps in assessing the route of the project towards progressing. This is attained by doing the work via ordinary temperament, familiar sprints and iterations. Teams must present the increment of work which is in suitable form. So in an Agile Model all the tasks are done in iterations and every facet of requirements, design, and implementation is revisited again and again during the Agile Life Cycle [5].

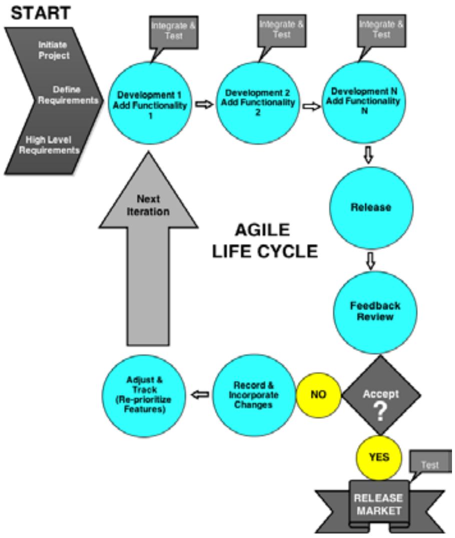

Figure 2. Agile Life Cycle

B. *What we gain from CC?*

CC is warmly accepted for performing industry and academia work. It passes on as prerequisite for computational resources which are available on order through a computer network. It allows businesses to access shared data center space for running their applications. CC is flourishing in the IT world due to the key advantages it provides:

- No need of hardware and software
- Quick deployment
- Less probability of failures
- Highly customizable environment
- Combination to other project solutions with ease

By keeping these advantages on account; CC has become a fury among companies around the globe. In CC Lifecycle [6] we go from Cloud evaluation & strategy to implementation and operation. Fig 3 shows the basic CC Lifecycle.

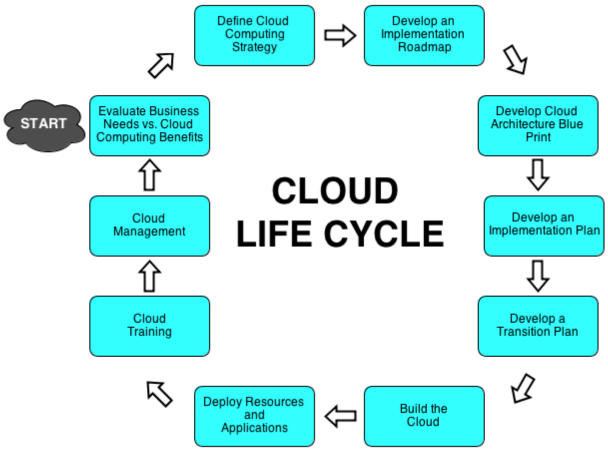
START
Evaluate Business Needs vs. Cloud Computing Benefits
Define Cloud Computing Strategy
Develop an Implementation Roadmap
Develop Cloud Architecture Blue Print
Develop an Implementation Plan
Develop a Transition Plan
Build the Cloud
Deploy Resources and Applications
Cloud Training
Cloud Management
CLOUD LIFE CYCLE

Figure 3. CC Life Cycle

Many advantages and chances can be gained by enhancing the agility of Software Development process and practices with the help of CC. Many researchers have been working on performing AD on Cloud Platform and some of them are concentrating on integrating both for getting extraordinary results. Abishek Jain and Reena Rani [7] performed an analytical study of AM with CC. Agile are software methods which are very realistic and light weight. These methods actually understand the reality that requirements are not rigid; they keep on changing in the business environment. They found that by combining Lean AD methodologies with CC results in proceeding technology. AD processes make the effective and perfect use of favorable circumstances offered by CC by releasing applications in iterative manner and by receiving user response repeatedly. Shortened development time, high stability and utilization of workloads, early feedback from customers, team efficiency, and cost reduction are some of advantages of CC when combined with Agile Methods.

Willie [8], CollabNet Incorporation, in his paper presented the idea of highlighting Agile Software Development in CC. In AM quick feedback is considered as king but new advancements in information technology can elongate the feedback cycle. So in order to lessen the feedback cycle automating production development is necessary. CC provides the business access tools and services and also suggests connections with other Clouds along with intelligence, integration and additional analytics. In Cloud-based AD, data from all the distributed automated tools is captured and shared. Broad detaining of data, intelligibly measuring performance of organization in real time helps a lot in properly managing and monitoring the Agile Projects, along with serving organizations with agility. In this paper, TeamForge supported AD by version controlling software, an enterprise search infrastructure, by instant provisioning of virtual servers and by security and governance.

Luisanna Cocco, Katiuscia Mannaro and Giulio Concas [9] proposed an easy tool that could facilitate the Global Software Development based on AD Method on the Cloud Platform. They used system dynamics for modeling and simulating the Cloud-Based Software Development environments that can be utilized by small and medium enterprises. Their developed process is based on Scrum AM. The basic aim behind proposing this model is to confirm that if Global Software Development environment on Cloud System lessen the budget and time judging against the basis of development environment for Global Software. To depict two dissimilar techniques of Global Software Development in shadow of related preliminary situations they developed a less complex system dynamics model. During simulation they only consider the variables (taken from literature) that have huge impact on development processes. Hence their proposed model helps the project managers in recognizing that how Global Software Development might be made easy by means of Cloud Development Environments.

The Landmark [10], in their white paper demonstrated that CC is the best platform for Agile Software Development. It allows you to satisfy your customer by providing them precious functionality quickly. It permits you to gather instant response from users and to formulate sudden amendments based on their feedback. An intrinsic benefit of CC is the fast development cycle which can never be put into operation in conventional development model due to high allocation cost. Saleforce.com R&D gain several underline advantages when they put AD into their CC stage; increased delivery rate, increased time to market, a good indicator of customer satisfaction, recommend the methodology to several other organization by convincing them with best results and increased production rate across the organization.

Shuai Zhang, Shufen Zhang, Xuebin Chen and Xiuzhen Huo [11] in their research work presented that CC is result of development in grid computing along with distributed and parallel computing. It can undeniably take in many opportunities for large-scale industry/enterprises like cost reduction and privileged earnings. End-user is the true owner of CC and in advancing day’s net service can be effortlessly accessed by notebook PC or mobile phone. The main aim of CC applications is to merge all possible resources, and allow users to access them when they want. Basic CC style on which all people agree is: 1. SAAS (Software as a Service) 2. Utility Computing 3. Network service 4. PAAS (Platform as a Service) 5. MSP (Management Service Provider) 6. Commercial Service Platform 7. Integrating Internet. The characters of CC are: ultra large scale, exceptionally low-cost, service on demand, virtualization, high reliability, Versatility and high extendibility.

Milad Torkashvan and Hassan Haghighi [12] mentioned some Cloud Platforms and Products; each of them only considers a subset of known characteristics of the CC. Due to this issue, Clouds don’t have any unified architecture. This paper tries to find a unified architecture and framework for CC relying on the service oriented view point to deploy all types of enterprises on Clouds and also the aim is to propose a CC framework to support all of aforementioned characteristics. To achieve this goal, the paper attempts to find a good relationship between CC, service orientation and some other technologies among these architecture paradigms, such as event driven architecture, semantic web and semantic web service. The layered architecture of proposed framework has 5 layers: IaaS, PaaS, INaaS (Intelligence as a service; this layer is responsible for automating the Cloud internal operation and providing users with the needed services), SaaS, and environment. Anything happens in the environment will be sent to the Cloud as an event; an event can be a user request for any service (from hardware resource to a business process) or it can be detected from environment data transactions. The framework is appropriate for private and public computing because of the ontology and policy based approach simultaneously. Being based on ontology and event orientation in this framework helps it to be Agile. The event driven approach used in the framework provides the Cloud broker operator; an easier way to communicate with other systems and Clouds. The degree of coupling is low because the Cloud users just deal with events, and the Cloud use ontology to select the appropriate services. Its event driven and semantic based approach make Cloud integration practical in real world.

III. CHALLENGES OF CC AND AM

A. CC Challenges

Users are still doubtful about the dependability of CC as currently its acceptance is coupled with frequent challenges. In 2008, IDC [13] [14] conducted a survey on CC to identify the major issues and challenges associated with it. The foremost challenges recognized by the organization are:

Security: Security is the major challenge that obstructs users to adopt CC such as data loss, phishing and running your data on many machines at a time. Putting your personal data on somewhere else CPU is really frightening as it pretense severe hazard to one's data and software. Furthermore, computing assets and the multi-tenancy model [15] has brought in new security challenges [16] in CC.

Costing Model: Mostly this challenge arises when the user uses the hybrid Cloud Deployment Model because in this the organization data is circulated among a number of Clouds so it increases the charges for data transferring and communication between different Clouds with increase in cost of computing resources as well.

Charging Model: Due to elastic resource pool, cost analysis has become more complicated than normal data centers. Especially, for SaaS Cloud providers it can be very extensive to develop the cost of multi-tenancy inside their recommendations. Re-designing and re-developing the application that was initially using for single-tenancy, new demanding customizations provided by new features, enhancement of security and performance for simultaneous user entrance and treating with the complications are all included in the cost of multi-tenancy. Therefore for the sustainability and prosperity of SaaS Cloud providers, a tactical and feasible charging model is vital.

Service Level Agreement: It is very important for consumers to obtain undertaking from the providers for whichever service they provide. This can be done through a Service Level Agreement (SLA); it expresses proper SLA specifications and details about the consumer expectations after weighing, verifying and assessing resource allocation mechanism on the Cloud. All the negotiations between consumer and provider are done through this agreement for projecting future complications. Different Clouds (IaaS, PaaS, and SaaS) have different SLA specifications which lift up many implementation troubles for providers. SLA evaluation framework [17] should be constantly integrated with user response and customization features to support advance SLA mechanism.

What to Migrate: In 2008, IDC survey [13] ensured that some systems and applications have been transferred to the Cloud. Even then the organizations are still facing privacy/security problems in shifting their data to the Cloud. The results exposed that organizations are most interested in utilizing IaaS instead of SaaS just to keep core activities in-house and outsource subsidiary functions to the Cloud. Survey also revealed that Storage Capacity moved by the organizations to the Cloud is comparatively low than Collaborative Applications.

Cloud Interoperability Issue: Every Cloud offers their client to interact their Cloud in their own special way which leads to create messed up Cloud phenomena. By locking the vendors, the development of the Cloud ecosystem is strictly hampered, because users cannot uses the offering of any other vendor at the same time so optimization of resources at different level of the organization can't be done. Now it's become complicated to incorporate Cloud service with an organization's own presented legacy system due to proprietary Cloud APIs. Interoperability is a fundamental thing for Clouds at several stages but the major goal of interoperability to understand the flowing of flawless data across Clouds and in linking Cloud and local applications.

B. AM Challenges

Although using AM is very beneficial. But when work on a big and complex development and implementation project is performed, many practical challenges could be faced. Here are some challenges [18] which we can consider the limitations of AM.

Quality, Cost, Time and Scope: According to PMI, if any of these constraints changes it brings a change in the left over. AM is flexible and it allows scope to change on regular basis which effects Cost, Time and Quality. On the other hand, time management for the difficult tasks is bit critical. If task is done earlier than defined time, the business might lose the benefits of AM and quiet reverse, if task time is elongated, projects exceeds the limit of budget or time.

'Ready to use' product: Under AD, performance testing can only be carried out after having considerable amount of deliveries because in AD an application process is always evolving and it may contain functional defects. Therefore, during the development stage the deliverable by no means in 'ready to use form' except if it performance tested.

Inability to 'design' for Future Requirements: In AD, reworking is done at various stages of development and testing. Because a system, on the basis of unseen requirements is very difficult to design even having the finest design models and a number of skilled design personnel on a project team.

External and Internal Dependencies: Mostly projects have external dependencies which are not in the control of the Agile Project Team. Any relatively dependent work can only be carried after making the external deliveries and external teams won't always follow the AM. When these dependencies are exposed during the development process many problems can be occurred. When different Agile Teams work on same Agile Project than internal dependencies come between them because work of iteration can be heavily dependent on the outcomes of other Agile Teams.

Need of Experts in Agile Team: As iteration is for a short time period so there is always a danger that the work items will not be completed in the estimated time. Therefore, it is must that each member of Agile Team must be an expert in their particular field.

IV. HOW CLOUD ENHANCES AND SUPPORTS AGILE?

1) 6 Means: Cloud enhances AD

For attaining efficiency and effectiveness and for speeding up development functions, AD Team uses CC. CC helps AD Team in bringing together multiple development, test, and production settings with other Cloud services flawlessly. Six main means [19] CC enhances AD.

A. *Availability of Limitless Number of Testing and Staging Servers by CC*

In Cloud platform, AD teams have almost infinite number of servers for development, testing and production. Multiple servers give ease of performing multiple activities in parallel. Though, when AD is used without Clouds, teams are restricted to only one physical server. And everyone has to wait for the server to become free to carry on their work.

B. *CC turns AD into an Accurately Parallel Activity*

For performing IT operations in Cloud Platform, AD Team can condition the servers according to their rapid need. But in AD delay occurs rather in arranging server instances or in installing essential platforms such as Database Applications. CC makes AD a parallel activity which brings more efficiency and effectiveness and helps in better utilizing AD Team.

C. *It Supports Innovation and Experimentation*

As in Cloud based AD a number of instances can be produced as required makes possible for AD Teams to bring innovations. When performing AD with CC, builds are quicker and less hectic, which promotes experimentation.

D. *It Increases Continuous Integration and Delivery*

Automated testing and builds demand time. In AD, during automated testing if the code fails then team members have to fix the code repeatedly till it passes all the tests. In CC, several virtual machines are available to the AD Team in their own Cloud or on the public Cloud which unexpectedly makes continuous integration and delivery fast.

E. *It Builds More Development Platforms and External Services Available*

AD groups requires assortment in project management, issue management and automated testing environments. A variety of these services are accessible in the Cloud as Software as a Service (SaaS).

F. *It Alleviates Code Branching and Merging*

In AD, it is presumed that features can be broken down into uniform size stories and scheduled for builds. Present releases need to be improved with little enhancements in code refactoring for using in production. Code branching and merging includes coping up with many versions of staging builds and development. You don't have to buy additional physical servers for these purposes with CC.

2) *3 Core Ideal: Cloud supports of Agile [8]*

Feedback, Transparency in Communications, and Time Boxing are the three unique tools, Agile used for supporting its main principles and practices. Also Agile needs an automated development infrastructure to maintain continuous integration. In Cloud based AD these issues can be resolved in lucrative way.

A. *Feedback is Key in Agile, and King in the Cloud*

Instant feedback is crucial in AD, and it's produced through stakeholder contribution, daily assignments, uninterrupted integration and testing. The key to Agile is to shorten this feedback cycle on the whole and this can be done by automating each part of feedback functionality such as code scanning, integration, testing, continuous integration activities etc. This automation will fasten the development and reduce cycles and also enhance scalability in AD process. The Cloud plays role in automating all stages of the projects. Some ways through which Cloud sustain fast response are:

Automated Development in the Cloud: For victorious AD, Cloud makes available distributed and handy source code management to numerous developers within no time. Cloud ensures security and up to time availability of source code.

Automated Build in the Cloud: During build, developers can attain their existing build images which are placed on different platforms with the help of Cloud. Cloud also permits convenient pricing by applying charges only for particular services offered by the Cloud which automatically reduces build costs.

Automated Testing in the Cloud: Testing at Cloud platform brings considerable progress in speed and nimbleness. Multi-platform can be run quickly using virtual images. Unit testing can be done in parallel using multiple Cloud machines as well which results in cost saving.

Automated Production Deployment in the Cloud: Cloud give direct way to production environment in minimum time; sometimes automate deployment with push-button control. Automatically getting stakeholders' feedback confirms the speedy and frequent deployment which keeps the project forwarding swiftly in the correct direction.

B. *Transparency: Getting a Sharp View of Reality in the Cloud*

Brawny and general communications is the center of successful AD projects. What people confer, plan and accomplish needs to transparent both on the business side as well as on the technical side. Real time transparency let team members to make knowledgeable decisions, concern advantageous directions by instantaneously considering possible tribulations and recognize best project asset.

In AD mostly transparency has two faces. One is non circuitous vocal communications in daily meetings within and across different teams. And the second is data capturing in real time which helps organizations in thoroughly supervising and managing the Agile Projects.

C. *Time-Boxing: Prioritizing Tasks at Scale in the Cloud*

In AD, Software Development is done in sprints, and each sprint is mainly the important piece of the product which is typically time-boxed in 2-4 week increments. In time boxing, both business and technical teams incessantly prioritize and re-prioritize features and deficiencies. This is an essential process because it helps in making tough choices and clarifying business priorities.

Transparency and trailing is always needed for prioritizing tasks, and monitoring and managing the actions of the present sprint. Deploying this type of tool takes much time and power in running with IT Enterprise. With Cloud Platform, Agile Management Tools can be provisioned in minutes and can balance with a huge number of users worldwide without managing IT infrastructure.

V. CLOUD BASED, AGILE APPLICATION DEVELOPMENT LIFECYCLE MANAGEMENT

Fig 4 provides a complete but tentative solution of Cloud Agile Application Development that thoroughly makes the management of software project's simple. This lifecycle management facilitates organizations with several development teams which support AD quickly and economically, through an extensive series of functionalities and features. This platform also aids in overwhelming the challenges of both CC and AD by performing Cloud AD all the way through:

Automation of End – to – End Process: [20] Processes workflow is done smoothly by avoiding people intrusion. Automation makes processes controlled, well organized, sustainable, and fully protected and error free.

Team Collaboration: To find out that the artifacts, documents, codes, Wiki content and discussion threads are hosted on Cloud or within our limits; an enterprise search infrastructure is designed.

Social Architecture: For coding and tracking artifacts, it provides co-operation in geographical and organizational limits transversely.

User Tool Integration: Provisioning and consumptions of assembled and test servers can be initiated by the developers. Integration of data from different data servers can be permitted to users. This lifecycle management also provides security and governance which is particularly significant in Cloud-based Platforms.

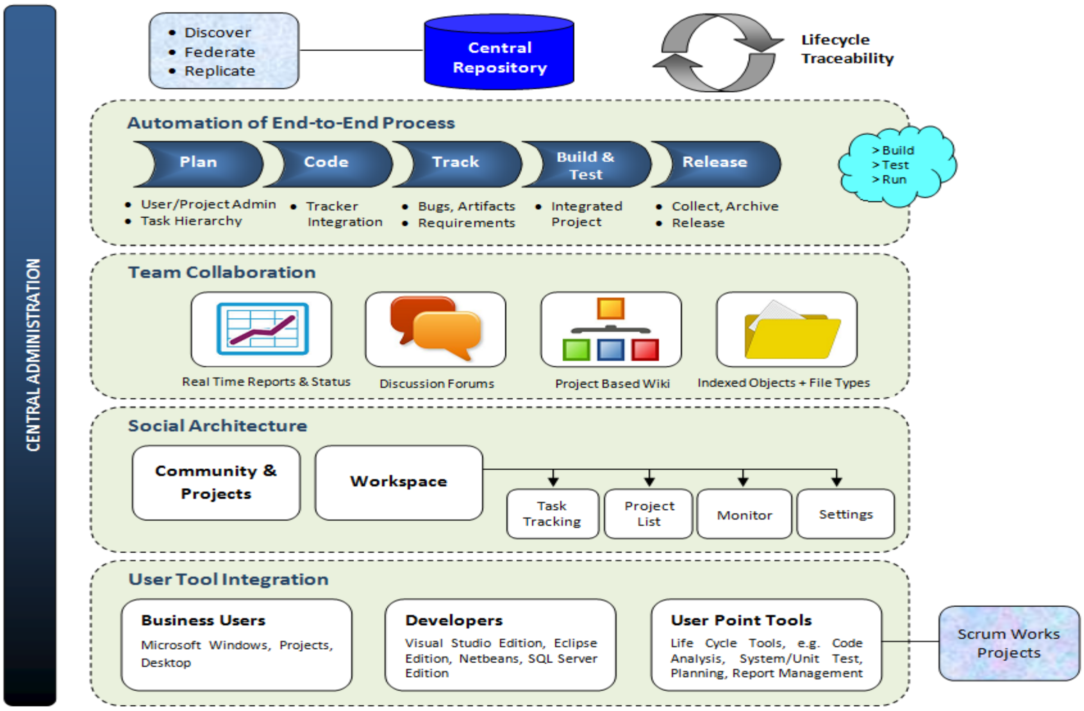

Figure 4. Cloud Based Agile Application Development Lifecycle

VI. BENEFITS OF ACD

- *Cloud is an Agile Accelerator* – Faster release cycles of higher quality can be enabled by combining CC with Agile Engineering. This shrink the time from conception to production operation.
- *Enables Business Agility* – Quick testing of novel ideas in marketplace, assemble customer reaction and changing courses for fulfilling the customer's demand by the organizations leveraging Agile and the Cloud. Only days and weeks are enough to do this. Cloud keeps initial investment in hardware very little and let organizations to spend extra as leveraging the Cloud to get better ROI.

- *Reducing IT Costs* – The costs of bringing ideas/concepts to production can be very low by improving AD and CC services i.e. SaaS, PaaS, and IaaS. Organizations are getting greater capabilities for lower costs, improving the bottom line by using Cloud based tools for development, monitoring, testing and management.
- *Improving Customer Experiences* – Long term business success and proceeds are considered to be driven by continuously improving customer experiences. Shipping and efficiency is desired by all large and small organizations but in upcoming days they will try to put on their customers in evocative experiences.
- *Growing the Economy* – The new financial system is a driving force for global economic growth. Small start-ups are getting strategic advantage of Cloud and AD to compete with large companies and creating new markets too. These start-ups are growing their businesses quickly by minimum direct investment which is an ideal way for AD and CC.
- *Self-service* – Developers can develop apps at the velocity and cadence of their choice as they are given the authority to spin up the app stack. They request only what they want when they want any resource instead of getting when they have no use of it.
- *Actual vs. Forecasts* – Developers consume exactly what they use, nothing more, this is a major benefit. Also a precise metric for both the cost and accomplishment side of things can be made.
- *Speed* – Clouds omit the queue of developers awaiting for the delivery of infrastructure or functioning stacks for their applications. Development and Release times are cut down due to short time required for Environments to be ready.
- *Expense* – Agile requires a sort of 'instant on' development and QA environment preparation capability and allow development teams to run large scale tests without incurring large scale costs. There is no need to incur capital costs for short term needs so pricing by usage is optimal.
- *Flexibility* – Developers can make experiment freely without any stress over capital cost. The expense profile changes from a CapEx (expenditures creating future benefits) to an OpEx (ongoing cost for running a product) orientation (which can have additional finance benefits) as platforms are readily consumed and about to be disposed.
- *Disposible* – Can optionally ensure that all PaaS provisioning for developer testing and app evaluation is precise, reliable, up to date, and right to the most recent platform specs and infrastructure standards. It allows you "infrastructure as code" and eliminates painful complications and expense.
- Work load is highly utilized and stable.
- Development cycle time is shortened up to 75%.
- Number of developers is fixed even for the large scale software development.
- People are physically close and cost of information movement among them is reduced.
- Customer feedback is collected earlier which results higher quality at the end.
- Management and development plans are highly flexible for any change.
- Time between making decision and to see its consequence is reduced.
- Team member's amicability -sense of community and morale- is improved so that they are more liable to convey the valuable information quickly.

VII. CONCLUSION

Combination of AD Methodologies with CC comes out with the best results. AD processes make the best use of the chances delivered by the CC by completing software development in iterations and receiving quick users' response.

The objective of this research is to highlight the main of AD and CC and by integrating both practices we showed that how we can overcome those challenges. Therefore, in this paper first we mentioned the challenges of AD and CC. After that, we stated the manners and core principles by which CC improves and holds up AD. Later we compose Cloud based AD Lifecycle that helps in managing the software projects and provides support to developers. In the end we proclaim the benefits of ACD those help in conquering the challenges of AD and CC.

We believe that ACD is a big hope for IT world, as organizations take pleasure in utilizing the services offered by both practices for making their projects unique and innovative.

REFERENCES

[1] Eric Knorr, Galen Gruman, InfoWorld, “Cloud Computing”, http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031, 2013.

[2] CollabNet, “Agile Methodology”, agilemethodology.org, 2008.

[3] Jim Highsmith, “History: The Agile Manifesto”, http://agilemanifesto.org/history.html,2001.

[4] Ryan Shriver, The Virtualization Practice, “Agile Cloud Development: The Future of Software”, June 4, 2012.

[5] Addicam.V.Sanjay, “Overview of Agile Management & Development Methods”, http://projectperfect.com.au/downloads/Info/info_agile_programming.pdf, 2005.

[6] Intelligence ‘Together We Can’, “Cloud Consulting / Cloud Lifecycle Management”, http://www.intelligence.co.im/Consulting_Detail.aspx?pid=429, 2013.

[7] Abishek Jain and Reena Rani, RTMC, IJCA, “Analytical Study of Agile Methodology with Cloud Computing”,http://research.ijcaonline.org/rtmc/number14/rtmc1076.pdf, 2011.

[8] Willie, CollabNet Inc., White Paper, “Reinforcing Agile Software Development in the Cloud”, http://www.open.collab.net/media/pdfs/CollabNet%20Whitepaper_Reinforcing%20Agile%20Dev%20in%20the%20Cloud.pdf?_=d, 2011.

[9] Luisanna Cocco, Katiuscia Mannaro and Giulio Concas, IEEE, “A Model for Global Software Development with Cloud Platforms”, 2012.

[10] The Landmark, White Paper, “Agile Development Meets Cloud Computing for Extraordinary Results at Salesforce.com”,http://www.developerforce.com/media/ForcedotcomBookLibrary/WP_Agile_112608.pdf, 2008.

[11] Shuai Zhang, Shufen Zhang, Xuebin Chen and Xiuzhen Huo, IEEE, “Cloud Computing Research and Development Trend”, 2010.

[12] Milad Torkashvan and Hassan Haghighi, ACM, “A Service Oriented Framework for Cloud Computing”, http://dl.acm.org/citation.cfm?id=2222469, 2012.

[13] Frank Gens, IDC eXchange, “IT Cloud Services User Survey, pt.2: Top Benefits & Challenges”, http://blogs.idc.com/ie/?p=210, October 2nd, 2008.

[14] IDC Analyze the Future, “IDC Cloud Research”, http://www.idc.com/prodserv/idc_cloud.jsp, 2013.

[15] Success factors, “Distinctive Cloud Technology Platform - Multi-Tenant Architecture”, http://www.successfactors.com/en_us/whysuccessfactors/technology/cloud/architecture.html, 2013.

[16] Ramgovind S, Eloff MM, Smith E, IEEE, “The Management of Security in Cloud Computing”, http://icsa.cs.up.ac.za/issa/2010/Proceedings/Full/27_Paper.pdf, 2010.

[17] C. Weinhardt, A. Anandasivam, B. Blau, and J. Stosser. “Business Models in the Service World”, http://www.im.uniarlsruhe.de/Upload/Publications/2f5d87da-1af5-4d44-b422-9b7e5802b5a5.pdf, 2009.

[18] Vikrant Bansal, G-CUBE, “Challenges of Agile Methodology”,http://www.gcsolutions.net/blog/challenges-with-agile-methodology/, February 26th, 2010.

[19] Nari Kannan, “6 Ways the Cloud Enhances Agile Software Development”, http://www.cio.com/article/714210/6_Ways_the_Cloud_Enhances_Agile_Software_Development?page=1&taxonomyId=3023, 20th Aug, 2012.

[20] Sairoop Technologies, “End to End Process Automation”,http://www.sairoop.com/Sairoop%20Technologies/Home/ProcessAutomation, 2012.

Engr. Ambreen Nazir is a MS Scholar as well as Research Associate in the Department of Software Engineering at University of Engineering and Technology Taxila, Pakistan. She graduated from University of Engineering and Technology Taxila in Software Engineering in July 2012. Her areas of interest are Windows Application Development, Web Development, Databases, Software Quality Assurance, Software System Design, and Software Requirement Engineering. She is trying to bring innovations in above mentioned fields.

Engr. Ayesha Raana is a MS Scholar as well as Research Associate in the Department of Software Engineering at University of Engineering and Technology Taxila, Pakistan. She graduated from University of Engineering and Technology Taxila in Software Engineering in July 2012. Her areas of interest are Software System Design, Software Requirement Engineering, Software Quality Assurance and Operating System.

Engr. Muhammad Fahad Khan is a PhD Scholar in the Department of Computer Engineering at University of Engineering and Technology Taxila, Pakistan. He has received his Master degree in Computer engineering from university of Engineering and Technology Taxila, Pakistan in February, 2010. He graduated from University of Engineering and Technology Taxila in Software Engineering in September 2007. His areas of interest are Video Summarization, Computer vision, Wireless Communication, Software Designing, Design Patterns, Software Requirements Analysis and Mobile Application Development.