

Research Article

Effective Implementation of Agile Practices – In Collaboration with Cloud Computing

Veerapaneni Esther Jyothi^{A*} and K. Nageswara Rao^B^ADepartment of Computer Applications, V.R.Siddhartha Engineering College, Vijayawada – 7, Andhra Pradesh, India^BPotti Sriramulu Chalavadi Mallikharjuna Rao College of Engineering and Technology, Vijayawada – 1, Andhra Pradesh, IndiaAccepted 02 May 2014, Available online 01 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

Even though the release cycles are speedier with agile software development there can be delays with respect to the insufficient availability of underlying platforms. But when agile using the cloud can overcome the delays and greatly enhance the speed of continuous integration and release cycles because of the large number of resources availability in the cloud. Effective implementation of agile practices is much more achievable with the collaboration of cloud. Developing software on time and within budget is not good enough if the product developed is full of defects and customers today are demanding higher quality software than ever before. Now-a-days the software market is mature enough and users want to be assured of quality. There are several benefits and opportunities to maximize the agility of the software development process and practices with the help of cloud computing.

Keywords: Agile, Cloud computing, services, staging and production.

1. Introduction

Over the years many different software methodologies have been introduced and used by the software engineering community. Developers and users of these methods have invested significant amount of time and energy in improving and refining them. The method they choose for software development depends on the type of organization, the type of project and the type of people involved. Agile software development methodologies have been greeted with enthusiasm by many software developers because work is done at different levels in parallel (Craig Larman, 2003). We can use our creativity on the design level and can also have the fun of implementing the design in a smart way.

Handling unstable and volatile requirements throughout the development life cycles and delivering products in short time frames and under budget constraints when compared with traditional development methods are the two most significant characteristics of the agile approaches. Even though the release cycles are speedier with agile software development there can be delays with respect to the insufficient availability of underlying platforms. But when agile using the cloud can overcome the delays and greatly enhance the speed of continuous integration and release cycles because of the large number of resources availability in the cloud. Effective implementation of agile practices is much more achievable with the collaboration of cloud. Now with the help of

cloud, agile software development is accelerated in terms of efficiency and effectiveness.

The present research work focuses on the impact of cloud computing during various phases of agile software development that affect the efficiency of software delivery in time and on budget

This research paper is organized as follows; section II presents the agile ideals. Section III Surveys the importance of cloud computing. Section IV explains how agile software development is accelerated using cloud.

2. Agile Ideal

Agile methods deal with unstable and volatile requirements by using a number of techniques, some of them are simple planning, short iterations, earlier release, plan driven, frequent customer feedback, etc.

These are the factors that made to deliver product release in short period compared to the waterfall model. Agile methods link developers and stakeholders/users to hit the mark quickly (Veerapaneni Esther Jyothi 2012). The identified strengths are listed below:

- Simplicity (process, design, and code)
- Accepting changing requirements even late in development
- Stakeholders and users satisfaction
- Refactoring, pair programming and frequent integration
- Interactive, incremental and test – driven development.

In Agile software development predefined set of steps may not lead to a predictable and desirable outcome because of changing requirements but a work with

*Corresponding author: **Veerapaneni Esther Jyothi**

determination and commitment is needed (E. Arisholm, 2007). Agile managers understand that demanding certainty in the face of uncertainty is dysfunctional. They set goals and constraints, providing boundaries within which creativity and innovation can flourish (Leffingwell, Dean, 2007).

User involvement is given high priority in the working style of agile, drawing user's right in to the heart the development process. The development of functionality is based on the user's desire. Agile processes is the discussing point here in this paper regarding how we can reduce development risk and avoid last minute surprises (Veerapaneni Esther Jyothi 2010).. Defect rate can be lowered with phased integration and continuous user feedback.

Despite being moving towards improved quality software, this also enables the principle of small iterative incremental releases which helps us to attain quicker ROI through quick release of business functionality (Alleman, 2005). While transferring to agile methodology from the traditional pattern it produced the benefits above expectations which replicated in the decrease in fault rate as well as producing high quality software (Baker, 2005).

One of the advantages of adopting an agile software development approach is a considerable raise in software quality. Amazingly there are two kinds of benefits for the customers who try agile. The first one is Quality and the second benefit is lower defect rates. Agile methods are a rising movement in the software field (Ming Huo, 2004). This realistic, people oriented method to software development demands software practitioners. Several adopters experienced enrichments in competence, superiority, work inspiration and consumer satisfaction (Nerur, 2005).

The Agile methodology can be summarized as below:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.
- Deliver product based on need from high priority down.
- Eliminating waste wherever possible.
- Refactoring, pair programming and frequent integration are the key factors.
- Interactive, incremental and test driven development (Veerapaneni Esther Jyothi 2012).

Table -1: Comparison of agile methods

Concept	XP	SCRUM	DSDM	CRYSTAL
Team size	3-16	5-9	2-6	4-8
No. of teams	1	1-4	1-6	1-10
Volatility	high	high	low	high
Team distribution	no	no	yes	yes

Different agile software development methodologies are Scrum , Extreme Programming (XP) , Dynamic Systems

Development Method (DSDM) , Crystal family of methodologies , Feature Driven Development (FDD). Agile methods place more emphasis on people, interaction, working software, customer collaboration, and change rather than on tools, processes contracts and plans (Veerapaneni Esther Jyothi, 2010). A number of new methodologies claiming these agile principles have been introduced. Each method has its own active research and user communities.

The best practices of agile software development methodologies that make the delivery of software in time and on budget are Continuous integration, Quality Assurance, Code Refactoring and more over customer satisfaction.

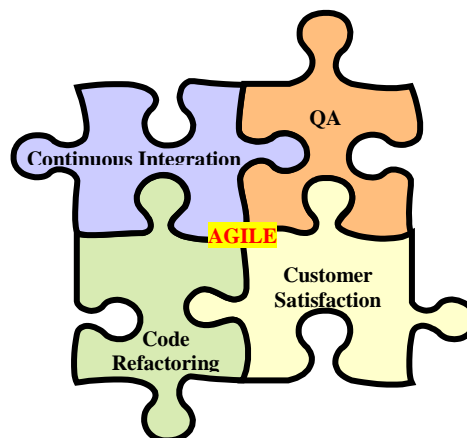


Fig.1: Best practices of agile software development methodologies

3. What cloud computing offers

The term Cloud Computing refers to the on-demand delivery of IT resources via the Internet with pay-as-you-go pricing.

The benefits of cloud computing can be listed as below:

- **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
- **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
- **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
- **Streamline processes.** Get more work done in less time with less people.
- **Reduce capital costs.** There's no need to spend big money on hardware, software or licensing fees.
- **Improve accessibility.** You have access anytime, anywhere, making your life so much easier!
- **Monitor projects more effectively.** Stay within budget and ahead of completion cycle times (Bowen, 2011).
- **Less personnel training is needed.** It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.

- **Minimize licensing new software.** Stretch and grow without the need to buy expensive software licenses or programs.
- **Improve flexibility.** You can change direction without serious people or financial issues at stake.

The services that cloud computing provides are

- **Software as a service (SaaS)**
Cloud-based applications—or software as a service (SaaS)—run on distant computers in the cloud that are owned and operated by others and that connect to users' computers via the Internet and, usually, a web browser.
- **Platform as a service (PaaS)**
Platform as a service provides a cloud-based environment with everything required to support the complete life cycle of building and delivering web-based (cloud) applications—without the cost and complexity of buying and managing the underlying hardware, software, provisioning and hosting.
- **Infrastructure as a service (IaaS)**
Infrastructure as a service provides companies with computing resources including servers, networking, storage, and data centre space on a pay-per-use basis.

Cloud computing deployment models are

- **Public cloud**
Public clouds are owned and operated by companies that use them to offer rapid access to affordable computing resources to other organizations or individuals. With public cloud services, users don't need to purchase hardware, software or supporting infrastructure, which is owned and managed by providers.
- **Private cloud**
A private cloud is owned and operated by a single company that controls the way virtualized resources and automated services are customized and used by various lines of business and constituent groups. Private clouds exist to take advantage of many of cloud's efficiencies, while providing more control of resources and steering clear of multi-tenancy.
- **Hybrid cloud**
A hybrid cloud uses a private cloud foundation combined with the strategic use of public cloud services. The reality is a private cloud can't exist in isolation from the rest of a company's IT resources and the public cloud. Most companies with private clouds will evolve to manage workloads across data centres, private clouds and public clouds—thereby creating hybrid clouds.

4. Accelerating agile software development by using cloud

Developing software on time and within budget is not good enough if the product developed is full of defects and customers today are demanding higher quality software than ever before (Coram, M., 2005). Now-a-days the software market is mature enough and users want to be

assured of quality. Even though the release cycles are speedier with agile software development there can be delays with respect to the insufficient availability of underlying platforms. But when agile using the cloud can overcome the delays and greatly enhance the speed of continuous integration and release cycles because of the large number of resources availability in the cloud. Effective implementation of agile practices is much more achievable with the collaboration of cloud.

Below are the benefits and opportunities to maximize the agility of the software development process and practices with the help of cloud computing as shown in figure 2:

- **Reduces the lead time** – Agile development teams are limited to one physical server. But when cloud instances are used, practically an unlimited number of servers are available for development teams. This leads to the reduction of lead time since they need not to wait for physical server to become free to continue their work.
- **Speed up release cycles** – Even though agile methodology is used for software development, there can be delays with respect to the availability of underlying platforms. Having a large number of resources available in the cloud, the agile development team can overcome the delays and greatly enhance the speed of continuous integration and release cycles.
- **Accelerates efficiency and effectiveness** – Even though the agile software development team performs several activities in parallel and also in serial, the parallel activities are delayed due to lack of sufficient services. Cloud computing can push it towards parallel activity by providing the required services which accelerates efficient and effective agile software development.
- **Encourages innovation** – Cloud encourages the development teams to innovate and experiment new strategies because of the wide range of available services.

We can provision platform as a service environments for all stages such as development, testing, staging and production. This will make sure; they have all the base services, such as databases, application servers, artifact deployers, service handlers and API managers setup in a consistent manner and ready to go. When the developer is done developing the application or service, the artifact can be moved to the next environment for testing, and all that is required is to point to the testing environment and test the application by the QA team (Mc Breen, 2003). This is development governance with cloud computing.

Project creation, repository creation, build plans, triggering builds upon commits and deploying artifacts onto the servers upon successful builds can all be automated with a cloud based integrated setup. In other words, continuous integration just got simpler. By incorporating the cloud into the agile development process, customer feedback becomes easier to obtain, because the application is being tested in a cloud-based environment.

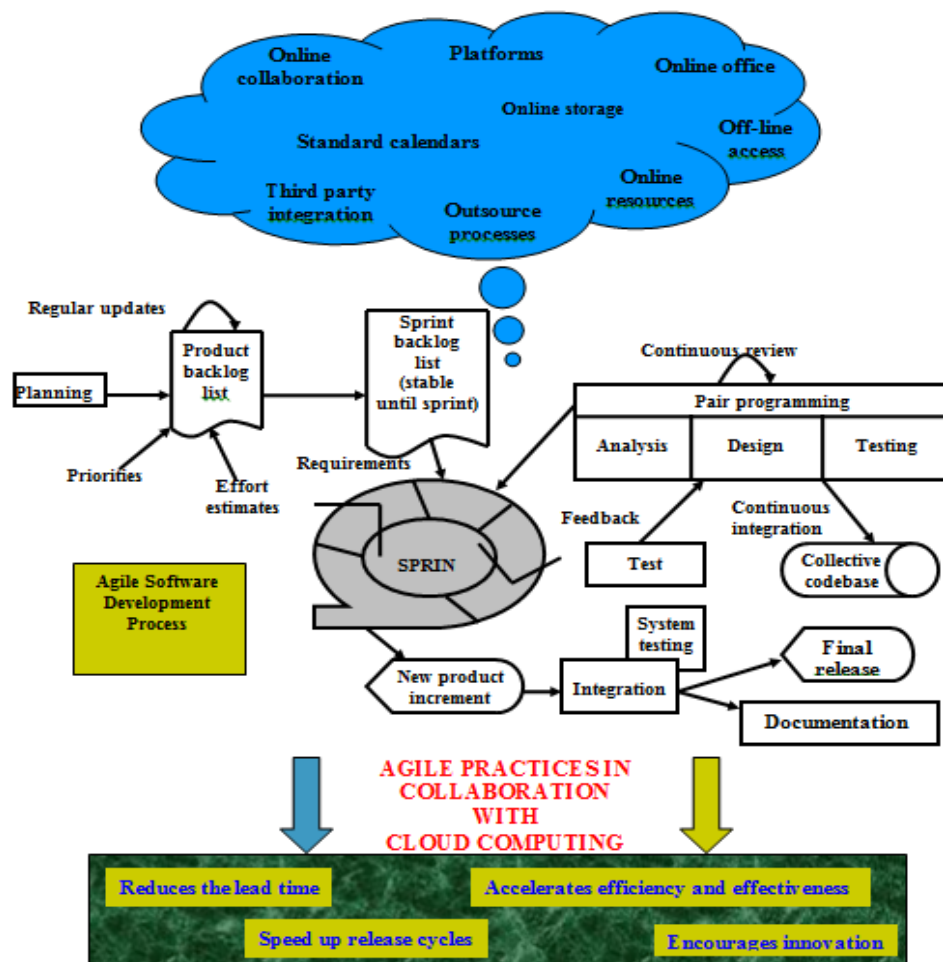


Fig.2: Accelerating Agile software development in collaboration with Cloud computing

Conclusions

What makes an organization agile is when the software that is being delivered is producing enough value to meet the business demand. In order to increase the value of our releases, we need to stop spending so much precious time racking and stacking infrastructure and managing application servers and databases, and spend more time adding valuable features for our business partners and customers. In other words, we need to embrace the cloud.

References

- Veerapaneni Esther Jyothi, K. Nageswara Rao (2012) Effective Implementation of Agile Practices, *International Journal on Computer Science and Engineering*, Vol. 4 No. 01 Page No. 87
- Veerapaneni Esther Jyothi, K. Nageswara Rao (2010) Effective implementation of agile practices – A collaborative and innovative framework, *CiiT International Journal of Software Engineering and Technology*, September 2010
- E. Arisholm, Gallis, H.E., Dybå, T., Sjøberg, D., (2007) Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise, *IEEE Transactions in Software Engineering*, pp. 65-86.
- Alleman, Glen. Agile Program Management: Moving from Principles to Practice. Agile Product & Project Management, Vol. 6 No. 9, Cutter Consortium, September, 2005
- Coram, M., and Bohner, S. (2005), The Impact of Agile Methods on Software Project Management, In *Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS)*, April 2005, pp. 363-370.
- Bowen, Janine Anthony. (2011). Cloud Computing: Issues in Data Privacy/Security and Commercial Considerations. *Computer and Internet Lawyer Trade Journal*, 28 (8), 8.
- Nerur, S., Mahapatra, R., and Mangalaraj, G. (2005), Challenges of migrating to Agile methodologies, *Communications of the ACM*, Vol. 48, No. 5, May 2005, pp. 72-78.
- Mc Breen (2003), P: Quality Assurance and Testing in Agile Projects. *Mc Breen consulting*, 2003, <http://www.mcbreen.ab.ca/talks/CAMUG.pdf>
- Baker, Steven W. (2005) Formalizing Agility: An Agile Organization's Journey toward CMMI Accreditation. *Agile 2005 Proceedings*, IEEE Press, 2005.
- Ming Huo, June Verner, Liming Zhu and Mohammad Ali Babar (2004), Software Quality and Agile Methods, *COMPSAC '04*, IEEE 2004.
- Craig Larman, Victor R (2003), Basili. Iterative and Incremental Development: A Brief History, 0018-9162/03/
- Leffingwell, Dean (2007), Scaling Software Agility: Best Practices for Large Enterprises, *Addison-Wesley Professional*, 2007.