page: 26-32 (7 pages)

Board for National Seminar on

ADVANCEMENT IN IT: CURRENT TRENDS, FUTURE SCENARIO AND IMPACT ON SOCIETY

KAMADHENU GUNA VIDYA
SADAIVA PHALADAYINI
BUNTS SANGHA'S S. M. SHETTY EDUCATIONAL INSTITUTIONS

Bunts Sangha's

S.M. SHETTY COLLEGE OF SCIENCE, COMMERCE &
MANAGEMENT STUDIES, POWAI

ISBN: 978-93-5142-876-3

Analyzing Agile Software Development Environment in Past-paced World using Cloud Computing

Ms. G.B. Hema Latha

S.K. Somaiya College of Arts, Science and Commerce,
Vidya Vihar, Mumbai.

Abstract

Aggressive competition and the heady rush of new technology are forcing software development organizations to reevaluate and readjust to market conditions every few days. Agile methods support customers business without being delay. Agile software development methods claim to be people-oriented rather than process-oriented and adaptive rather than predictive. Cloud computing provides this necessary acceleration needed to enhance the agile development. Leading companies are changing the way their teams equip and support software development teams by integrating fast, dynamic, flexible, and easily shareable cloud-based environments that are available on-demand, by integrating cloud-based services into the overall agile architecture strategy, software development teams are better enabled to create, change, and scale complex computing environments as often as needed. And at the same time, organizations are able to retain the full visibility and control required for security and operational governance over these environments.

The first part of this paper provides an insight into how agile development methodology enhance software development and the second part deals with the cloud computing virtualization that provides the apparent framework for agile software development.

Keywords: *Cloud Computing, Agile, Virtualization.*

Introduction

Industries are now using agile software development teams and cloud computing to accelerate development operations. The Agile development model enables software teams to produce higher quality software that is more in sync with customer needs and delivers release cycles faster and more cost effectively than ever before. Unfortunately, most software teams that adopt agile development struggle to achieve its full potential due to legacy challenges, especially in the enterprise. While the Agile model accelerates the software development process, many teams find that their environments are not optimized to support the full potential of their agile development release cycles.

These legacy environments are often too slow, inflexible, and inadequate for agile development processes. Consider this typical provisioning time for an enterprise-grade development environment can take, at a minimum, from several weeks to several months. Most Agile release cycles span four to six weeks at most. Not only does a four week provisioning delay result in sub-optimal outcomes, but once created, these static environments are also difficult to change and cannot support the rapid iteration required for agile development. Essentially, agile development methodologies require agile infrastructure for optimal efficiency. Cloud computing and virtualization make agile development teams to combine multiple phases of development with other cloud services and support rapid release of product.

Objectives

The main objective of this research is to analyze different agile methodologies and brainstorming sections associated with agile methods, becoming more and more established with large global organizations and learning about rapid feedback loops from customers that support rapid software development also, the study of how cloud computing provides general purpose on demand infrastructure of computing resources using on premise, public or external data centers.

Agile Methodology

Agile development was invented in the nineties and has revolutionized how software is developed by emphasizing short development cycles based on timely customer feedback. Agile software development is a method based on iterative and incremental development. With Agile framework, the development phases is constantly subjected to the reality check of actual users. This methodology involves interaction between self-organizing and cross-functional teams. There are a number of agile software development methods but the most popular agile methods are Extreme Programming (XP) and Scrum. In general, agile methods attempt to minimize risk and maximize productivity by developing software in short iterations and deemphasizing work on interim work artifacts and it is people-oriented.

Here are some of the key characteristics of the agile methodology.

- Deliver frequently.
- More iterations.
- Less defects.
- Test frequently.
- Collaborative approach.
- Maximum ROI.

It is very effective in environments where frequent changes in the requirement happen quite often due to various business reasons. Since it has more iterations, more than one build deployment for a project. It involves more client interaction and testing effort which assures a solution for the Requirement. Life cycle of agile shown in Figure 1. It is believed by some people that there is less documentation in Agile. But agile also includes documentation and it can be used either small or large projects.

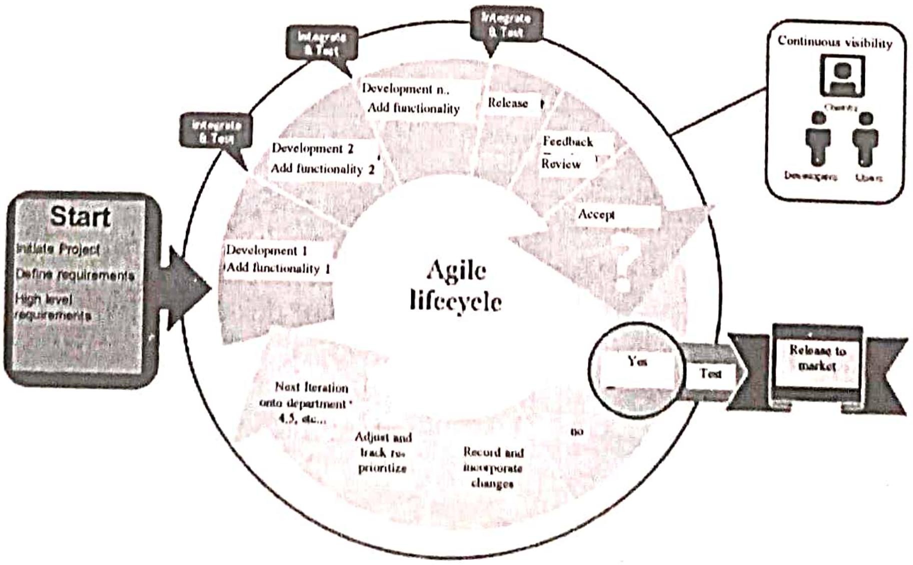

Fig. 1: Agile Life Cycle

Effective Practices for Agile Approach

Implementing Agile software development came from a sample of five organizations that successfully adopted an Agile approach as per the US General Accounting Office (GAO) Report GAO-12-681, "Effective Practices and Federal Challenges in Applying Agile Methods", here presented below, grouped into five key project management activities: Strategic Planning, Organizational Commitment and Collaboration, Preparation, Execution, and Evaluation.

A. Strategic Planning

1. Strive to be more Agile. Allow for a gradual migration to Agile appropriate to your readiness. Observe and communicate with other organizations implementing Agile.
2. Follow organizational change disciplines, such as establishing a sense of urgency and developing a change vision.
3. Be prepared for difficulties, regression, and negative attitudes.
4. Start with Agile guidance and an agile adoption strategy.

B. Organizational Commitment and Collaboration

1. Ensure all components involved in Agile projects are committed to the organization's agile approach. Identify an Agile champion within senior management.
2. Ensure all teams include coaches or staff with agile experience.
3. Empower small, cross-functional teams.

C. Preparation

1. Train the entire organization in your agile approach and mindset, and train agile practitioners in your Agile methods.
2. Ensure that subject matter experts and business team members have the required knowledge.
3. Enhance migration to agile concepts using Agile terms and examples. Create a physical environment conducive to collaboration.
4. Identify measurable outcomes, not outputs, of what you want to achieve using Agile Negotiate to adjust oversight requirements to a more Agile approach.
5. Make contracts flexible to accommodate your agile approach.

D. Execution

1. Use the same duration for each iteration.
2. Combine Agile frameworks such as Scrum and XP if appropriate.
3. Enhance early customer involvement and design using test-driven development.
4. Include requirements related to security and progress monitoring in your queue of unfinished work (backlog).
5. Capture iteration defects in a tool such as a backlog.
6. Expedite delivery using automated tools.
7. Test early and often throughout the life cycle.

E. Evaluation

1. Obtain stakeholder/customer feedback frequently and closely.
2. Continuously improve Agile adoption at both the project level and organization level.
3. Seek to identify and address impediments at the organization and project levels.
4. Determine project value based on customer perception and return on investment.
5. Gain trust by demonstrating value at the end of each iteration.
6. Track progress using tools and metrics.
7. Track progress daily and visibly.

Types of Agile Methodologies

Agile methods are a family of development techniques designed to deliver products on time, on budget, with high quality and customer satisfaction. This family includes several and very different methods. The most popular include:

- Scrum
- eXtreme Programming (XP)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)

- The Crystal Family
- FDD

(i) Scrum Development: Scrum is a lightweight management framework with broad applicability for managing and controlling iterative and incremental projects of all types. This method concentrates particularly on how to manage task within a team based development environment.

The Scrum development process concentrates on managing sprints. Product owners determine what needs to be built in the next 30 days or less. Before each sprint begins, the team plans the sprint, identifying the backlog items and assigning teams to these items. Teams develop, wrap, review, and adjust each of the backlog items (See Figure 2).

During development, the team determines the changes necessary to implement a backlog item. The team then writes the code, tests it, and documents the changes. During wrap, the team creates the executable necessary to demonstrate the changes. In review, the team demonstrates the new features, adds new backlog items, and assesses risk. Finally, the team consolidates data from the review to update the changes as necessary.

Following each sprint, the entire team—including management, users, and other interested parties—demonstrates progress from the sprint and reviews the backlog progress. The team then reviews the remaining backlog and adds, removes, or reprioritizes items as necessary to account for new information and understanding gathered during the sprint.

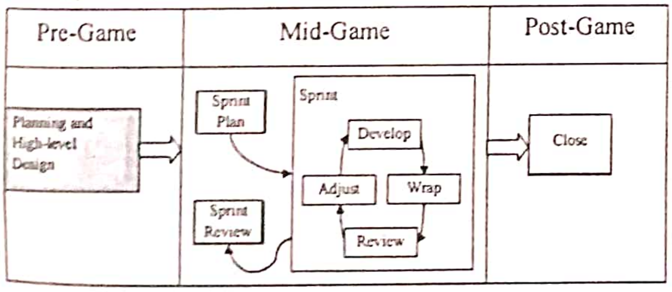

Fig. 2: The Scrum Process

(ii) XP (eXtreme Programming): XP is a more radical agile methodology, focusing on the software development process and addressing the analysis, development and test phases with novel approaches aimed at making a substantial difference to the quality of the end product.

In agile software development projects, project management takes a slightly different form, relying more on the project manager's skills in communication, facilitation, coordination, and emphasizing less on planning and control. XP stands for extreme programming. Working of simplified XP process is shown in Figure 3. It concentrates on the development rather than managerial aspects of software projects.

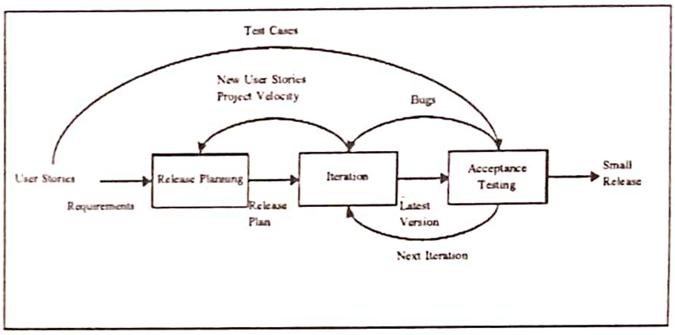

Fig. 3: A Simplified XP Process

Principles of Agile Methodology

1. Active user involvement is imperative
2. The team must be empowered to make decisions
3. Requirements evolve but the timescale is fixed
4. Capture reqùirements at a high level; lightweight and visual
5. Develop small, incremental releases and iterate
6. Focus on frequent delivery of products
7. Complete each feature before moving on to the next
8. Apply the 80/20 rule
9. Testing is integrated throughout the project life cycle – test early and often
10. A collaborative and cooperative approach between all stakeholders is essential

Cloud Computing

Cloud Computing is a model that enables convenient, on-demand network accesś to a pool of shared and configurable computing resources that are rapidly provisioned with minimal management effort or service provider interaction. Cloud computing is purely based on internet. A cloud actually is a grid of computers serving as a "service-oriented" architecture to deliver software and data.

Why Need of Cloud Computing in Agile?

The key advantages of cloud computing includes:

1. Maximizes the effect of sharing resources
2. Avoids upfront infrastructure cost
3. Allows enterprises to run applications faster
4. Helps achieve higher economies of scale
5. Provides a centralized storage mechanism
6. No hardware or software required for cloud computing services
7. Easy combination with other enterprise solutions
8. Highly customizable environment
9. Quick deployment, coupled with less probability of failovers
10. Optimum utilization of in-house IT resources
11. Capitalizing on these advantages, cloud computing has gradually become a rage among Companies around the globe. According to Gartner, cloud computing will become the preferred vehicle for application delivery by 2015. Thus, the majority of CIOs are gearing up IT infrastructure to suit the cloud environment.

Cloud Environment

A typical cloud environment consists of categories of services and cloud deployment models.

A. Categories of Services

- **Infrastructure as a Service (IaaS):** Provides virtual machines and other abstracted hardware and operating systems.
- **Platform as a Service (PaaS):** Allows customers to develop new applications using APIs, implemented and operated remotely.

  The platforms offered include development tools, configuration management and deployment platform.
- **Software as Service (SaaS):** Is software offered by a third party provider, available on demand, usually through a Web browser, operating in a remote manner.

B. Cloud Deployment Models

- **Public cloud:** Infrastructure is available to the general public or large industry group and is owned by an organization selling cloud services.
- **Private cloud:** Infrastructure is operated entirely for a single organization. It may be managed by organization or a third party, and may exist in-premises or off-premises.
- **Community cloud:** Infrastructure is shared by several organizations and supports a specific community.
- **Hybrid cloud:** Infrastructure is composed of two or more clouds (private, community or public) that are bound together by standardized or proprietary technology that enables portability of data and application.

Agile and Cloud

Agile development was lacking a development platform that supports the rapid development cycles that make the methodology work. Cloud computing makes a substantial difference here by eliminating the cumbersome distribution requirements that can bring agile development to a crawl (See Figure 4). With no patches to distribute and no reinstallations needed, Cloud computing makes it available to users immediately by installing the new distributions on hosted servers. This provides a possibility that the application you run today was modified just the night before. It is now evident that cloud computing is what agile development has been waiting for. When it comes to traditional software environments, new software distribution is a tedious task that needs reinstallation, and help from the support team. Under such circumstances, months or even years are needed to get a new distribution to the users. Incorporating their feedback for the next release then requires comparable time.

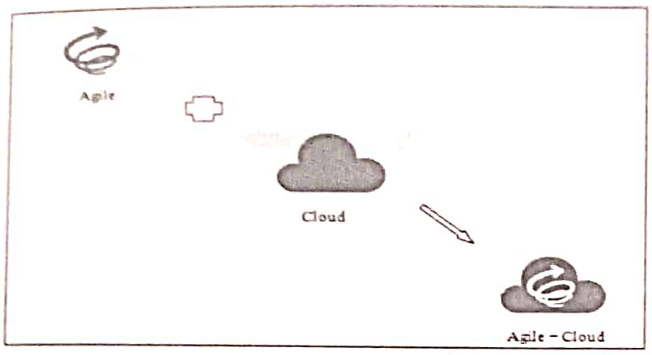

Fig. 4: Cloud and Agile Computability

Agile cloud is an information base that creates information system requirements and maintains the information model. It is the business application need of the hour.

How Cloud Computing Enhances Agile Software Development?

The different ways cloud computing enhances agile development are:

- **Cloud encourages innovations and experimentation:** By combining agile software development with cloud computing, the team can build faster products with handy code and also test. There is absolutely no need to wait for the next release as in the case when only limited servers are available.
- **Cloud enhances iterative development through continuous integration:** The testing phase of the software development cycle is an iterative process that the team needs to subsequently fix the errors that occur on testing. As there are a large number of virtual machines to the agile team within the cloud, it accelerates the speed on delivery. Thus, the cloud's virtualization enhances integration in time.
- **Cloud provides servers for development:** By making use of the Cloud computing virtualization, the software development teams in an agile environment have unlimited number of servers available. Without the cloud's facilities, the teams will be limited to just one server per development. Cloud computing reduces the dependency for physical servers and hence proceed to the development.
- **Cloud computing facilitates code branching:** Agile practices development efforts last longer than a release. Code refactoring is to be enhanced and used in production. In some cases, even code branching is necessary, where a lot of puzzling is necessary. With Cloud computing ,the upfront cost of renting servers for these sort of purposes can be avoided.
- **Cloud computing provides delivery platform for Agile Development:** Cloud offers many development services as Software as a Service (SaaS) and the Agile development can make use of these services in combination with virtualization.

Future Research Implementations

Cloud computing is how software applications are delivered today. It is the result of advances in technology, ranging from increased processing, increased sophistication of scalable storage region networks, ubiquitous high bandwidth network access, and the increased security and reliability of the internet. Agile development processes optimize the opportunity provided by cloud computing by doing software releases iteratively and getting user feedback more frequently. It is essential for software development organizations to consider lean agile development methodologies while coming up with their cloud computing strategy.

Conclusion

Agile methods are light-weight software methods. Agile development methods are very pragmatic in understanding the fact that requirement in a business environment changes constantly. Cloud computing and agile development overcome traditional development risk. This paper is a study related to the cloud computing's indispensability when agile teams aim to produce standard products via continuous integration and delivery methods. Agile development points to a parallel activity with the cloud computing features than a serial one cutting out the delays in provisioning the components. Thus, it is evident that the enterprises turn to this combination as it provides better chances for innovative development with standard Business objectives.

References

1. www.slideshare.net/.../agile-methodology-in-cloud-computing
2. software4cloudcomputing.com/methodology.html
3. http://research.ijcaonline.org/rtmc/number14/rtmc1076.pdf
4. http://global.sap.com/netherlands/services/consulting/pdf/Whitepaper_Agile_ENG.pdf
5. http://www.incose.org/chesapek/Docs/2010/Presentations/2010_03_Rico_ROI_Agile.pdf
6. http://www.slideshare.net/perforce/beyond-scrum-introducing-simple-scalable-agile-development?related=2
7. http://agilemanifesto.org/principles.html
8. http://www.allaboutagile.com/what-is-agile-10-key- principles/
9. http://blogs.atlassian.com/2013/07/agile-requirements-documentation-a-guide/
10. https://www.google.co.in/?gfe_rd=cr&ci=Z9rHU56wO86EoAOOzYDwCg&gws_rd=ssl#q=why+agile+methodology+is+required
11. www.wikipedia.org/