Review on Agile Modeling for Distributed Development

Dr. K. Sunil Manohar Reddy & Mrs. G. Pratibha

Associate Professor, Department of CSE, Matrusri Engineering College, Hyderabad, Telangana.

Degree Lecturer, Department of Computer Science, TSWRDC, Warangal East, Telangana.

sunil186@gmail.com, pratibhareddy19@gmail.com

Abstract: *As compared to the traditional process models, agile model has proved its worth with fast paced development and quick iterations. Enormous growth in global IT industry has created offshore clients and development resources for many software companies. Many global clients and organizations are out sourcing their projects for better work on cheap rates with high demands. Agile model has certain limitations when development environment is globally distributed and as in most cases client is offshore. Existing agile principles do not support distributed development architecture. This paper focuses on the problems that arise when client is offshore and proposes an agile model for distributed development environment.*

Keywords: **Requirement gathering, Distributed Development, Agile development**

1. Introduction

The enormous growth in the global IT industry has opened doors for software companies with offshore clients involved in out sourcing their projects. Traditional web applications have been evolved into web Information Systems (IS). Globalisation have tremendously impacted IT industry and changes the IS development style dramatically [2]. Web applications vary from traditional software development in various terms like:

1. More technologies (HTML, XML, ASP, JSP, PHP etc.)
2. Large number of users
3. Fast development
4. Short and frequent iterations
5. Continuous maintenance
6. Scalability

Web application development contributes 70% of the total development to the IT industry. But there is still a lack of some well defined and established process model typically suitable for developing web applications. Web engineering follows many conventional Software Engineering's fundamental concepts and principles [8]. In addition, it incorporates specialized processes models, software engineering methods adapted to the characteristics of this kind of application, and a set of enabling technologies [6]. Agile model hence is a process model adapted by most of the software companies for web applications development. The claims made by agile process proponents and critics lead to questions about what practices, techniques, and infrastructures are suitable for software development in today's rapidly changing development environment [11]. The true picture inside these companies is totally different. The principles of agile manifesto which were published by agile alliance during 2001 meeting are actually not the basis of agile development in these companies.

Agile model except some basic principles do not provide any concrete model and set of processes to be followed like Water fall model, Spiral model and Rapid Prototyping. Mostly agile practices are adaptive. Each organization modifies agile methods according to its requirements and development environment. This approach has produced many different development techniques each claiming to be an agile as in recent years a number of processes claiming to be "agile" have been proposed in the literature. This practice has left the standing principles of agile far behind [11].

Globally Distributed Development Environment is a new paradigm of agile development. Limitations of agile model as discussed above in this context have started new research directions in agile development. The distributed development in this respect is widely in practice and 60% - 70% development is focused on large web applications newly termed as web information systems. Shorter time to market, shorter product life cycles and continuous maintenance are much more pronounced in the case of web applications as compared to traditional ones [9].

According to agile manifesto agile method is best suitable for small scale, short duration projects where client and developers have face to face communication. Face to face communication is as important in distributed environments as non-distributed environment [11]. Documentation less development is another aspect of agile model but where client is offshore and project is a large web based IS this principle does not support the overall process. A lot of problems then arise in distributed environment. Requirement gathering and tracking is one of the main phases of process models responsible for 90% success of the projects. Requirement specification and analysis do not take the implementation platform into account, and, thus, are very similar to the corresponding activities in the conventional software engineering [8]. Light weight model and processes with high output are required to support this fast offshore development.

This research proposes a solution to improve the overall development process and presents a light weight model termed as "The Distributed Agile Development (DAD) model emphasizing on most required phases and respective documentation. This model would facilitate developers to synchronize their tasks and priorities with client requirements and expectations.

The proposed model can be implemented in web based projects for a distributed development environment where client is at a remote location. Some development resources can also be located at client site. With some modification in requirement document and phases this model can be adopted as well for software development (Desktop applications, utility software). For a software company, it is essential to understand how to link business management and software development and employ a solid, business oriented approach in its development decision making [12].

2. Methods

The Agile Manifesto

To standardize the agile processes agile principles are developed by agile alliance in 2001 called agile manifesto (www.agilealliance.org/principles.html). Agile manifesto describes a set of twelve principles for a process to be qualified as agile. This research focuses on four principles of agile manifesto in context of distributed development environment. These principles are listed below:

1. Welcome changing requirements, even late in development
2. Deliver working software quickly
3. The best architectures and designs emerge from self-organizing teams
4. The most efficient and effective method of conveying information to and within a development team is face-to-face.

Limitations of Agile Principles in Distributed Development Environment:

In this section we discuss the limitations of above said agile principles identified in context of distributed development environment as unique role the client has in agile software processes is not reflected or supported by the existing metrics [7].

1. Poor documentation
2. Communication gap between Client and Developers

The Distributed Agile Development (DAD) Model

All the processes that claim to be an "agile" inherits the deficiency of agile manifesto for not supporting the distributed development environment. The model proposed in this research presents an agile framework for distributed development environment. The phases in the DAD model (Figure1) are very light weight to complete a project with minimum necessary documentation.

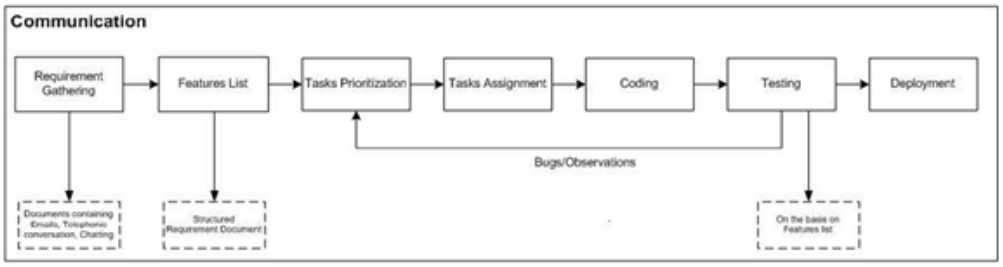

Figure 1: Phases in DAD Model

All the phases in the model are linear and a phase is dependent on the output of its previous phase.

Main emphasis in this model is given to the communication between offshore client and developers. All the phases completes under the umbrella of communication which is shown as a big rectangle containing all the phases. From requirement gathering to deployment of the project which is actually a delivery of a complete build all the phases require a close interaction with the client at each and every step.

Requirements from client are gathered and a simple proper structured document is prepared. On the basis of the requirements a features list is prepared. Tasks are prioritized depending upon the client's requirements, functionality and are assigned to particular developers for Coding. All the possible features are implemented and complexities are continuously communicated to the client. After testing observations and bugs

The Incremental Approach

The DAD is used in incremental fashion in which a build is released in each iteration. At each increment, we determine what assets have been added, and what new attack paths might have been opened [4]. Each next build contains new features received from the client or changes in the previous build. In a week a build is released. So all the functionality, enhancements and bug fixing is released in each increment with respect to functionality enhancement and time (Figure 2).

In each Build a complete set of functionality is delivered to the client. All the phases for each build are repeated and this process tend to progress in incremental fashion till the project ends. Internal build numbers are fixed to track record of the functionality which can also be exposed to the client. Communication with the client is ongoing task throughout all the phases.

The most important implication to managers working in the agile manner is that it places more emphasis on people factors in the project: amicability, talent, skill, and communication [3]. Regular communication and feedback daily or after a day or two from the client is a continuous part of the DAD model.

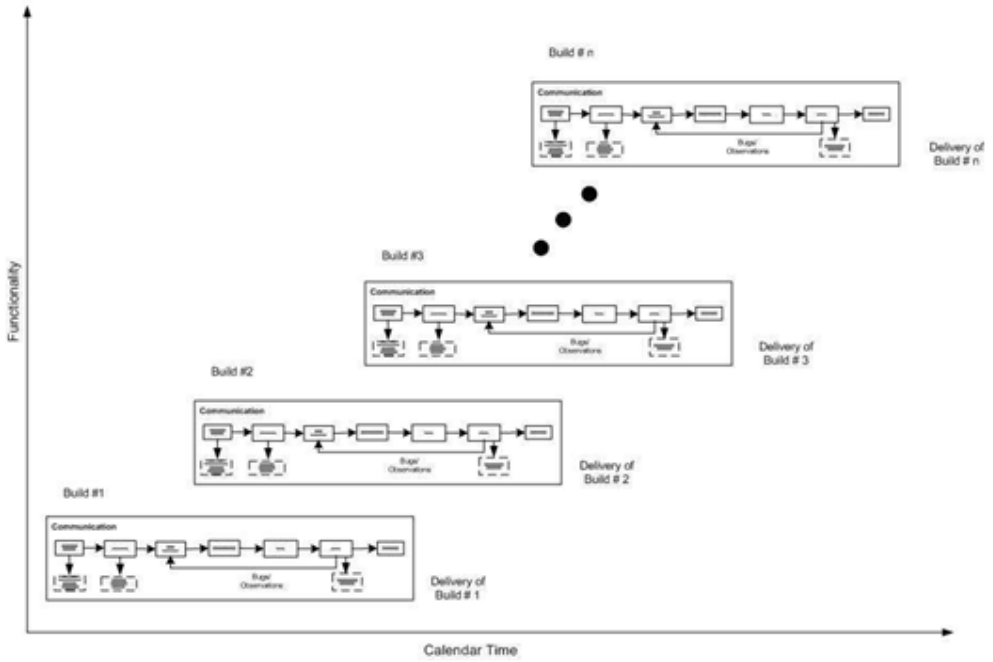

Figure 2: The stages of Waterfall Model

3. Results and Discussions

In distributed development environment communication process becomes more critical. Unlike other agile processes the DAD model covers the problem of communication gap and documentation when development resources are distributed. Using multiple models and each model should present a different aspect of the system and only those models that provide value to their audience should be built [8]. Agile processes such as Extreme Programming are of increasing interest in software development [4]. Communication phase is an integral part of this model. Frequent and very regular communication with client enables the developers to implement all the functionality with 100% customer requirements and satisfaction. Almost daily communication on emails, conference calls, and chatting makes developers everything clear and provides the scalability and transparency to distributed resources. The problems faced by client being at some other global location are highly overcome with maintaining minimum documentation.

Unlike other agile techniques like XP, Scrum, Feature Driven development DAD model supports longer duration projects. While we believe that both agile and heavy weight processes, e.g. model driven development, have merits in different contexts, agile processes are particularly well suited for building web application, particularly because they aim to deal with changing requirements and rapid delivery [1]. Time constraint, communication gap and minimum documentation when development is incremental or evolutionary the documentation generation may also be incremental or evolutionary are part of this model in a realistic manner [10]. Using this model for distributed development environment for all kind of software gives complete customer satisfaction overcome the deficiency of agile manifesto in distributed development environment. If both client and suppliers use the common process, they can communicate better and reduce unnecessary problems arising in coordination and control of projects [2]. Also, clients should consider the level of expertise, available resources, technology, and infrastructure in the supplier site [5].

4. Conclusion

This model emphasizes on the communication as a full time activity between client and development team. Web application development requires "demanding support infrastructures". Light weight processes to make this model more efficient in terms of requirement gathering and tracking processes will be the future enhancements to more strengthen this model. Further improvements to make this model efficient are being investigated. Enhancements and modifications in its phases are expected to produce a more efficient model.

REFERENCES

[1] *A. McDonald and R. Welland. Agile web engineering (AWE) process. Technical report, Department of Computer Science, University of Glasgow, UK. 2001.*

[2] *Cho. J. Globalization and Global Software development. ACM. USA. 2007.*

[3] *Cockburn, A. and Jim, H. Agile Software Development: The People Factor. USA. 2001.*

[4] *Ge. X., R.F. Paige, F.A.C. Polack, H. Chivers and P.J. Brooke. Agile development of Secure Web Applications. ICWE, ACM, California, USA. 2006*

[5] *Herbsleb. J., and Moitra. D. Global Software Development. IEEE software, March/April. 2001*

[6] *Pressman. R.S. Software Engineering: A Practitioner's Approach, 6th edition, Mc Graw Hill, 2005.*

[7] *Racheva. Z., and M. Daneva. Using Measurements to Support Real-Option Thinking in Agile Software Development. ACM. 2008.*

[8] *Souza. V.E.S. and R.A. Falbo. An Agile Approach for Web Systems Engineering. ACM. Pocos de Caldas, Minas Gerais, Brazil. 2005.*

[9] *Tarawneh. H, A. Elsheikh and S. Lahawiah. Web based application development in small firms. Proceedings of 6th Conference on Software Engineering, Greece. 2007.*

[10] *Theunissen. W.H.M., D.G. Kourie and B.W. Watson. Standards and Agile Software Development. ACM. SAICIST.2003.*

[11] *Turk. D, R. France. Limitations of Agile Software Processes, ACM,USA. 2008.*

[12] *Vahaniitty. J and K. Rautiainen. Towards a Conceptual Framework and Tool Support for Linking Long-term Product and Business Planning with Agile Software Development.ACM. 2008.*