

A Novel Iterative Software Development Model Using Cloud

K.Gobinath

Computer Applications

Bannari Amman Institute of Technology

Sathy, India

gopinathkannan1990@gmail.com

P.Sathish Kumar

Computer Applications

Bannari Amman Institute of Technology

Sathy, India

Sathishk604@gmail.com

Abstract— Software process is used to produce products according to plan, whereas simultaneously improving the organization capability to produce product. This research deals with an efficient process of developing software. It is concerned with the process model which builds the software using the cloud. The iterative model which evolved as a boon to eliminate the cons of waterfall model, also suffers from the drawbacks of repetitive process. So, to overcome the disadvantages of traditional iterative method we have proposed a novel iterative software development model using cloud computing. The main objective of this research is to revive the iterative model for improving the efficiency in software development process using cloud environment.

Keywords-software development; iterative; cloud computing; beta version.

I. INTRODUCTION

The impact of computer is increasing day-by-day in the fields of life such as industry, medicine, commerce, education, media and even agriculture. It has become an important element in the industry and technology of advanced in developing countries. Since the past four decades, software had been developed from a tool used for analyzing information or solving a problem. However, the premature programming techniques have created a number of difficulties making software an impediment to software development mainly those relying on computer. In order to overcome those difficulties the scientists of various disciplines and standards proposed a concept of software engineering. The term “software engineering” was invented in the late 1960s after the recognition that all the instructions educated about how to program well were not helping to build better software systems. A programmer codes a complete program, whereas a software engineer develops a software component that will be pooled with components written by other software engineers to build a system.

The IEEE [IEEE93] termed software engineering: “(1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application

of engineering to software. (2)The study of approaches as in (1).” Parnas [1978] defined software engineering as the “multi-person construction of multiversion software.” In classic software engineering disciplines, the engineer has tools and mathematical guidance to specify the properties of the product separately from those of the design. The typical software engineer relies much more on practice and verdicts rather than mathematical techniques. Whereas practice and verdicts are necessary, formal analysis tools also are essential in the practice of engineering. Software engineering is performed by inventive, knowledgeable people who should adapt a prime software process so that it is suitable for the products they build.

II. SOFTWARE DEVELOPMENT PROCESS

Software engineers framed a series of software process which enables them to build and complete the software. Software development process is schematically represented in Fig 1.

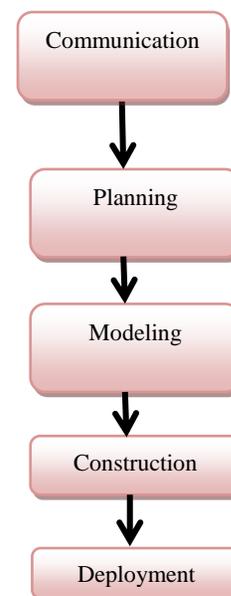


Figure 1. Software Development process.

III. OVERVIEW OF ITERATIVE MODEL

An iterative process flow repeats one or more of the activities before proceeding to the next. The problems with the waterfall model laid the foundation for a new method of developing software which provides faster results, and offer greater flexibility. In iterative software development model, the software project is segmented into small parts. This provides convenience to the development team to predict results prior on in the process and attain important feedback from the system users. Each iteration actually resembles a mini-waterfall model process with the comment from one phase providing needed information for the designing of next phase. The iteration involves reframing and implementation of certain task. The goal of the iterative model is simple, straightforward, modular which supports redesigning. Initially the project estimation will be inaccurate. But after two or three iterations the estimations will be precise. The iterative model based on the notion of optimization cycle, was also developed. It consists of a test and an action component that couple each other by feedback. The meticulous validation of requirements and verification of every version of the software against those requirements inside each cycle of model is the triumph of iterative design. It draws a similarity with mathematical methods that use successive estimation to extract results at final solution. By successive approximation the product can be developed using the iterative life cycle model. The functioning of iterative model is depicted in Fig 2.

A. Pitfalls of Iterative Model

The involvement of user community is demanding on the time of staff and leads to procrastination of project delivery. Repetitive process for customer satisfaction at each phase leads to confusion. Iterations are not scheduling milestones. The process involves an iteration of same phase, so it may result in the prodigal resource usage and so not considered to be economical.

IV. NOVEL ITERATIVE DESIGN USING CLOUD

A. What is cloud?

Instead of running programs and data on an individual desktop computer, everything is hosted in the “cloud” – a nebulous assemblage of computers and services accessed via the internet. Cloud computing lets you to use all of your applications and documents from anywhere in the world emancipating you from the precincts of the desktop and making it easier for group members in different locations to work together.

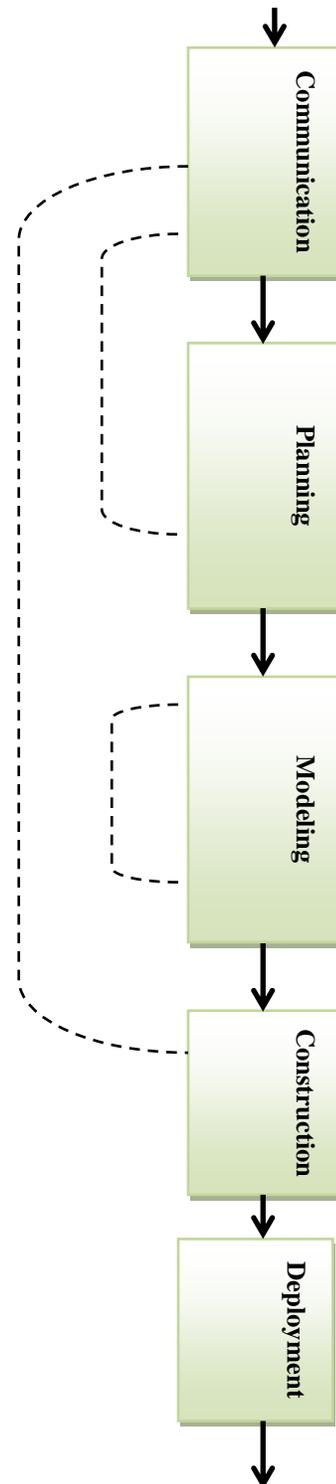


Figure 2. Iterative Process Flow.

B. Relating Cloud with Iterative Model Design

The iterative model repeats the process at each phase, when the user intervenes in the corresponding phase development. This leads to the confusion and time consumption which result as a delay in project completion. The “Novel Iterative Model” using cloud environment, is proposed to overcome the drawbacks of traditional iterative model.

In NIMuC (Novel Iterative Model using Cloud), when the user proposes his expectations the requirements should be gathered. Once the requirements are collected it should be formulated towards planning and actual process proceeds. Before proceeding to the planning activity, the requirements should be shared with the authenticated members of the cloud, who are situated at different locations. When those authenticated members get the requirements they can formulate a plan, design and model. When the user is not satisfied with the actual process it will be convenient for developers to retrieve a plan or model framed by different members on the cloud. Whereas, if there is an obstacle in any of the actual software process development phase, developers can resolve it by sharing the problems in cloud environment. The architecture of NIMuC is represented diagrammatically in Fig 3.

C. Beta Version

With the task accomplishment of the construction phase, the beta version can be released. The beta version enhances the user to have a trial usage of the application on the cloud environment. If the user gets satisfied the activity towards deployment phase can be fulfilled. When the user is not comfortable with application re-modification can be done with the repetitive process which becomes easier by using cloud computing.

Here is an example of how our proposed model NIMuC works, in Fig 4. Consider a company Mac which has branches at different locations of the country. One of the branches of Mac Company, which operates at Newyork, receives a project from a client. The client wanted the company to develop an application for online polling system. Now, Newyork branch developers collect the requirements for the project. Before proceeding towards the planning phase the requirements should be shared on cloud environment with the other branches of the same company situated at Chicago, Boston, and Atlanta. Now, apart from the developers of the other branches, the programmers, testers can collect the requirements shared by Newyork branch. Those authenticated members of the company can formulate a plan and design. So they can share it on cloud with the Newyork branch.

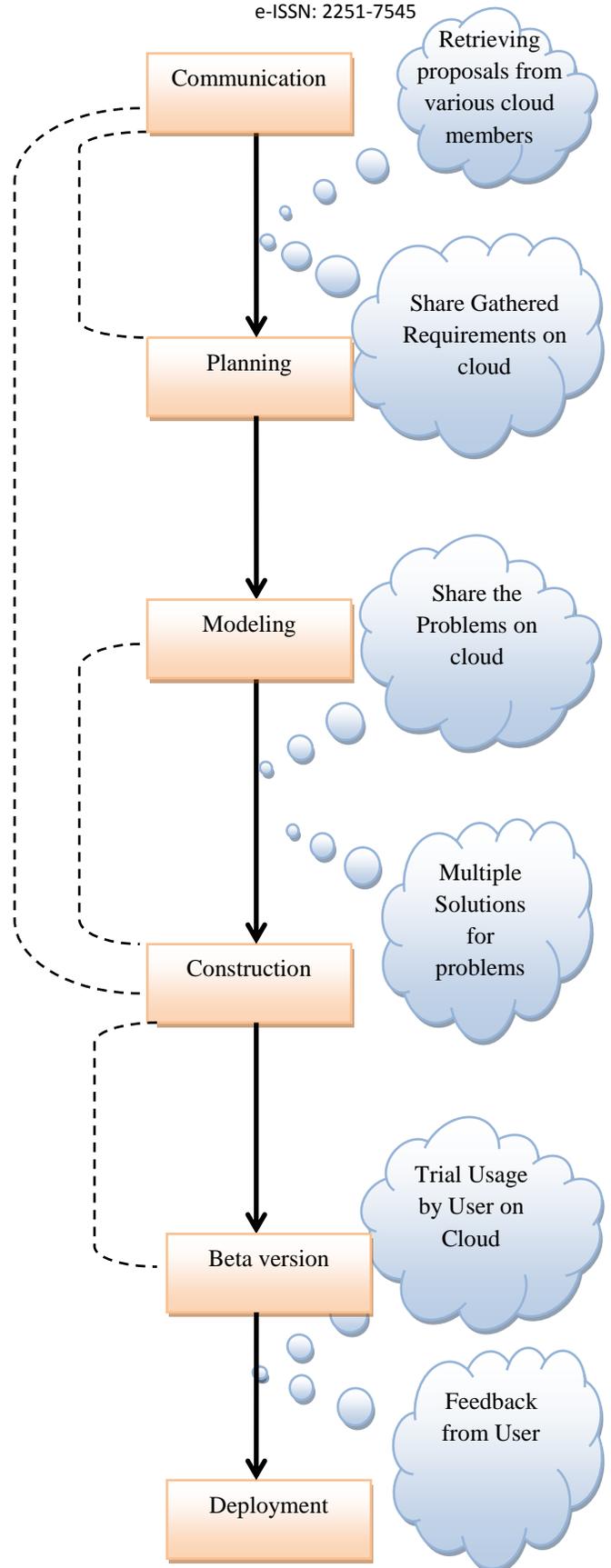


Figure 3. Architectural View of NIMuC

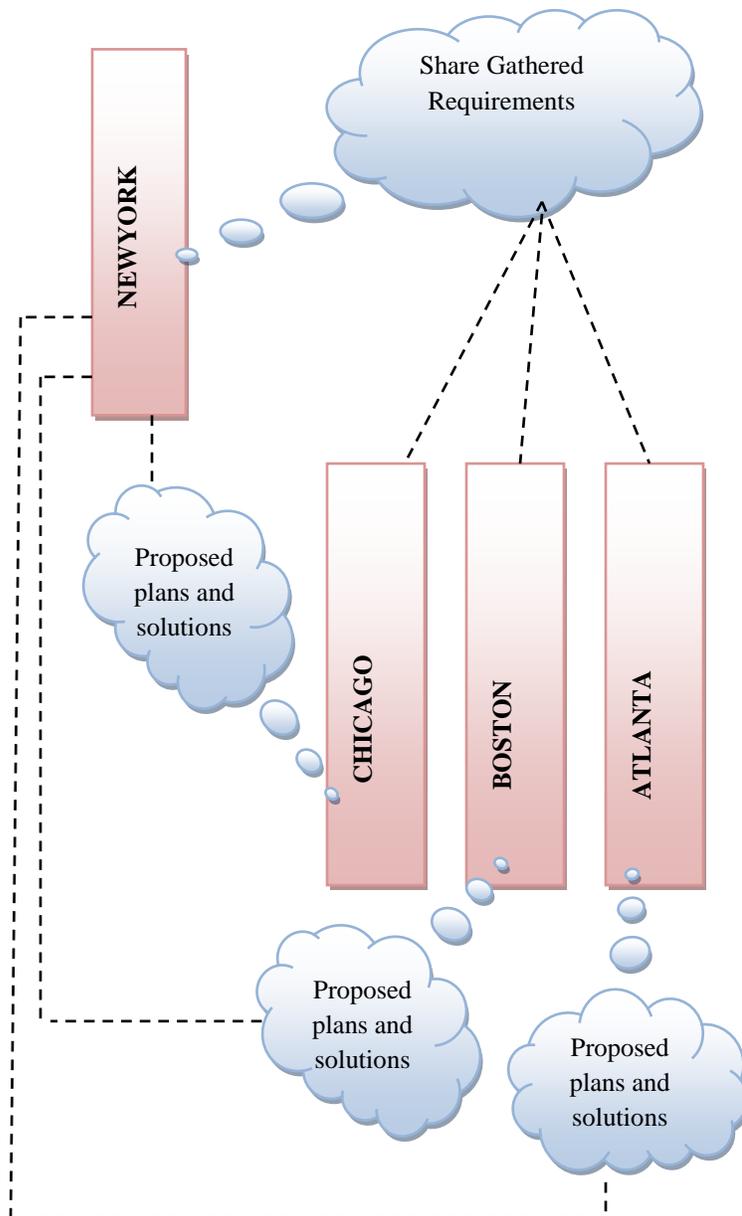


Figure 4. Example of how NIMuC Works

D. Pros of Novel Iterative Model using Cloud

The advantage of using Novel Iterative Model can be summarized as follows:

1) *Usability*: When the process is built with the influence of cloud, it will be convenient for the developers to make maximum utilization of project resources. Thus NIMuC connotes the increased

usability of system. It helps the developers of a project to acquire knowledge in different aspects.

2) *Portability*: The software is portable, if it can run in different environments. The software development itself is built using the distributed environment known as cloud computing. Therefore, during the development of the software there exist possibilities to check the portability for the system.

3) *Interoperability*: The interoperability refers to the capability of a system to coexist and collaborate with other systems. In NIMuC, modules can be designed at different systems and can be integrated as an application.

4) *Productivity*: The term productivity refers to the quality of software process referring to its efficiency and performance. The efficiency can be termed as fast delivery of the product. Although NIMuC is an iterative process it saves time. For example if a problem occurs it can be resolved easily by retrieving a solution from the cloud, shared by many members.

5) *Reusability*: In product evolution, we may tend to modify a product to build a new version of that same product. NIMuC provides the propitious reusability of the product at different levels of development.

V. CONCLUSION

The major novelty of this work is use of cloud computing in the existing iterative software development model. The alacrity of the development process has been maximized by reinforcing the coordination between developers. The NIMuC can be applied to many projects to enhance the completion of projects successfully within the estimated time period. The beta version of a developed application which is also shared on cloud environment will make the users to give convincing feedback.

ACKNOWLEDGMENT

My sincere thanks to Principal, Dr.A.Shanmugam for having given me an opportunity to do a research work, I owe my gratitude to Prof.R.Murugan for continuous support to work on this research, finally I take a privilege of thanking Asst.Prof.P.Sathish Kumar who have been providing me a moral guidance for completing this work.

REFERENCES

- [1] Matthias Rauterberg, "An Iterative-Cyclic Software Process Model," Proc. International on Software Engineering and Knowledge Engineering, Capri, Italy, June 15-20, 1992.

- [2] Waman S Jawadekar, "Software Engineering Principles and Practice," Tata McGraw-Hill Publishing Company, New Delhi, 2004, ch. 2.
- [3] Watts S. Humphrey, "Managing The Software Process," Pearson Education and Dorling Kindersley, New Delhi, 2006, ch. 1-2.
- [4] Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, "Fundamentals of Software Engineering," 2 nd ed. Prentice Hall, New Delhi, 2007, ch. 1-4.
- [5] Michael Miller, "Cloud Computing Web Based Applications That Change The Way You Work and Collaborate Online," 2009, ch. 1-3.
- [6] Nabil Mohammed Ali Munassar, A.Govardhan, "A Comparison between Five Models of Software Engineering," in International Journal of Computer Science Issues, vol. 7, Sep. 2010.
- [7] Roger S. Pressman, "Software Engineering A Practitioner's Approach," 7 th ed. McGraw-Hill, 2010, ch. 2.
- [8] Sandeep Kumar, Satayaveer Sangwan, "Adapting the Software Engineering Process to Web Engineering Process," in International Journal of Computing and Business Research," vol. 2, 2011.