Configuration Management System for Distributed Software Development

Mr. Aaditya Vishwakarma, Prof. Mahendra Rai(HOD)
Dept. of CSE , M.E.(SS) SRIT Jabalpur(M.P.) ,
Dept. of IT. SRIT Jabalpur(M.P.) India.

Abstract -Distributed software development (DSD) has recently became an active research area it is popular in software industry. In this paper we propose complete environment for distributed software development. Multiple developers can utilize this environment to work on same research but from geographically distributed locations. Software development in geographically distributed environment has been becoming popular in the software industry since last decade. In contrast with distributed software development, in centralize development all team members resides in the same building or same office. This kind of centralize development has been disappearing since last decade because of lack of recourse and lack of expert people at single place, outsourcing, reduced labor cost in developing countries etc. Distributed software development suffers from communication, coordination and collaboration are key problems. In order to cope up with these problems following features will be implemented and integrated in single environment.

Keyword*:* Distributed Development, Software Engineering, Software Development Process.

1.0 Introduction

Software development in geographically distributed environment is becoming popular in today's software era. Centrally located Development, in which all team members located in the same building or same office, has been disappearing since last decade. In distributed software development, various software development activities like requirement gathering, coding, testing etc. are performed concurrently from multiple locations. The reason behind distributed software development are lack of resources at one place, lack of expert people, reduced labors cost in developing countries, business outsourcing etc. In centrally located development, all the team members work in close interaction with each other. But due to physical distance, this kind of close interaction among team members is not possible in geographically distributed setting. Therefore the main problems in distributed software developments are communication, collaboration and coordination among team members. Lack of proper Communication and coordination among team members make the various activity in Distributed software development difficult for example requirement management, testing, project management, Project planning, tracking and oversight etc. [1][4].

1.1 Distributed Software Development

Distributed Software Development is becoming popular nowadays because of outsourcing, reduced labor cost in developing countries, unavailability of resources and skilled people at one location etc. Later in this chapter, some of the ways to solve the problems in distributed software development is discussed

After that this paper gives overview of some of the popular tools that has been developed for distributed software development.

Distributed Software Development. Distributed Software Development is becoming popular nowadays because of outsourcing, reduced labor cost in developing countries, unavailability of resources and skilled people at one location etc. Configuration management is of prime importance in Distributed Software Development. Distributed Software Development. Distributed Software Development is becoming popular nowadays because of outsourcing, reduced labor cost in developing countries, unavailability of resources and skilled people at one location etc. Later in this chapter, some of the ways to solve the problems in distributed software development is discussed. After that this chapter gives overview of some of the popular tools that has been developed for distributed software development. In the last section, some of the research papers published in this area is described in short. Configuration management is of prime importance in Distributed SoftwareDevelopment.

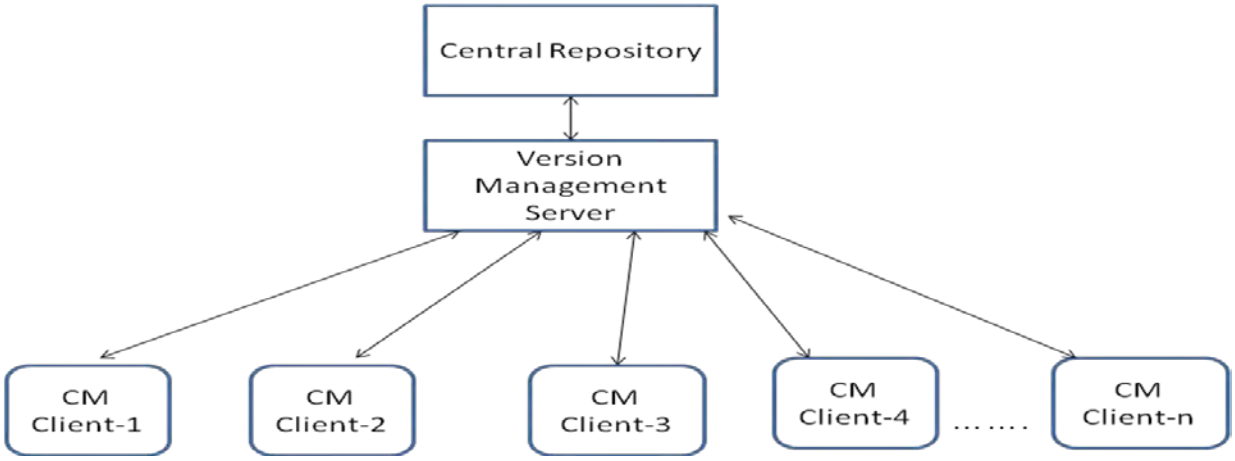

Fig. Architecture of Configuration Management System

1.2 Distributed Configuration Management System

This tool is aimed at resolving conflicts among developers. It will provide support for basic configuration management functions e.g. check in and check out of files, version management etc. The main purpose of this tool is to allow multiple developers to work on same projects, even on same files without conflicting each other's work. Whenever any change is made to any file, original file is preserved by creating new version of that file. This tool will be based on centralized architecture. All the information will be stored at server side. Whenever access to any information will be needed by client, connection is made to the server and information will be relayed to client.

Centralized Architecture is used for the development of this tool. All the information is stored at central place. This information is relayed to client on demand. There is a central **Repository** where all the information is stored. There is a version management server that listens to clients and serves various client requests e.g. check in, check out, import, merge, update etc. version management server directly talks to the repository. There is a client module that provides easy to use GUI for performing various configuration management information e.g. check in, check out, import, merge, update etc.

2. Exiting System

Software development in geographically distributed environment is becoming popular in today's software era. Centrally located Development, in which all team members located in the same building or same office, has been disappearing since last decade. The time and distance that spread members of a distributed team, tend to enlarge the effects of everyday challenges to successful projects. This enlargement of these challenges is so significant that they become almost entirely new problems. The primary challenges are:

- Increased distance and time between team members
- Reduced response times
- Reduced bandwidth
- Wider cultural differences

Increased distance and time means more infrastructures, collaboration and coordination is required to set up connections for communication. Reduced response time means any conversations among team members will take longer time to complete. Reduced bandwidth, in the sense the rate and amount of information that can be effectively exchanged among remotely located team members, similarly slows down communication. Wider cultural differences among people enlarge the problems with articulating complex concepts in such a way that this can be easily understood by other people. People might no longer share local textual information, basic traditional metaphors, or even speaking language in order to communicating ideas. These problems of communications consolidate to slow down the process of shaping an effective and intelligent network or team that can communicate and collaborate effectively.

3.Proposed System

In distributed software development project, multiple team work on the same project but in geographically distributed locations. The key problems that are faced in this kind of project are coordination, communication and collaboration among team members. There must be some methods to share ideas among team members. There must be some mechanisms to share information among team members. There must be some kind of shared repository for proper coordination and collaboration among team members. This thesis will focus on two principal areas that are of prime importance in distributed software development project

This thesis is aimed at devising effective strategies and their implementation for two kinds of problems that arise in distributed software development.

- communication problems among developers due to physical distance
- Work conflict among developers due to working of multiple team members on the same part of the project but from different locations.

In order to cope up with these problems following features will be implemented and integrated in single environment architecture. All the information will be stored at server side. Whenever access to any information will be needed by client, connection is made to the server and information will be relayed to client.

3.1 Document Repository:

In any software development project, there are different kinds of things that need to be shared among all the developers e.g. WebPages, help documents, e-books etc. Document repository is aimed at providing shared space for all these kind of information. Each developer can read and write any informal things to this document repository. This tool is used for sharing of only informal things. There is no version management kind of things in this tool.

3.2Instant Messenger :

This tool is aimed at providing informal communication mechanism among developers. Using this feature any team member can send the instant message to any other team member.

3.3 E-Conference:

In Distributed Software Development environment, due to physical distance face to face meeting is not possible. E-Conference tool is aimed at providing facility by which multiple team members can collaborate with each other by means of text.

There are many issues related with distributed software development. Scope of this thesis includes study of communication and coordination problem among developers in geographically distributed software development. This thesis mainly focuses on practice approach and its implementation to solve the communication and coordination problems that arise in geographically distributed software development. This thesis will be a sincere attempt to develop an environment that can be efficiently used by developers engaged in distributed software development.

In this thesis we have designed and developed complete environment for distributed software development. Multiple developers can utilize this environment to work on same project but from geographically distributed locations. Configuration management tool, instant messenger, document repository and E-Conferencing tool are incorporated in this single environment.

4. TOOL USED

4.1 Importing project into repository

Once client tool is properly running, the first step towards the use of this tool is to import the empty or initial project into repository located at server. Once project is imported into repository, any number of developers can simultaneously work on that same project but from geographically distributed locations without worrying about conflicting each other's work.

4.2 checking-out project from repository

Once project will be imported into repository, it will become available for all developers. In order to work on project, the first step is to check-out recent version (or any version you want to work) of the project into local working directory.

4.3 checking-In project into repository

Once any revision of project is checked-out into local working directory, developer can edit the files into project, can modify the files, can delete the files and can add new files into the project. Once significant change is made to project, developer should check-in these changes into repository so that his or her changes will become available to other developers working on the same project but from remote locations.

4.4 Synchronizing local working directory with most recent version in repository

Update function is provided by this tool in order to synchronize local working directory with most recent version in repository. As a result of execution of this function, changes made by other developer will become visible to developer.

5. Results and Discussions

Distributed Software Development suffers from basically two kinds of problems as given below.

- **Communication problems among remote developers:**

In distributed software development, due to physical distance, face to face communication is not possible. Effective communication is critical to the successful completion of each of the principal phases of software development, as well as to the successful transition between phases.

- **Coordination problems among remote developers:**

In distributed software development, it is likely that multiple team members work on same part of the project from multiple locations. There must be some way to share information among developers.

To overcome the problems of communication among remote developers this tool provides two kind of communication mechanism namely, **Instant messaging** facility and **e-conferencing** facility. Using instant messaging developer can send instant message to any other developer located remotely. In this way, developers can remain in instant contact with other developers located remotely. Using e-conferencing facility multiple developers can organize meeting instantly using text messages.

Configuration management facility provided by this tool is of prime importance for Distributed Software Development. This facility allows multiple developers from geographically distributed locations to work on the same project, even on the same module or same file of the project without disturbing each other's work. In order to do this, versioning facility is provided by this tool. Document repository provides shared space where any developer can read and write any informal things e.g. web pages, images, files etc.

Distributed Software Development can be made as effective as centralize development. Some differences between this tool and some of the tools developed in this area are given in the next section. Integration of communication tools with coordination tools makes this integrated environment effective for Distributed Software Development.

6. Conclusion and Future work

6.1 Conclusion

The purpose of this thesis is to deal with the problems arising in Distributed Software Development. If communication and coordination problems are solved, Distributed Software Development can be made as effective as centralized development. In this thesis, I have successfully implemented and integrated tools for solving communication problems and coordination problems. Tool and approach presented in this thesis can be used by software developers engaged in Distributed Software Development to make the Distributed Software Development as effective as centralized development.

6.2 Future Work

In distributed software development each developer should be aware of responsibility and task assigned of every other team members .Task Manager is aimed at providing facility for creating work break down structure, assigning task to team members, browsing of work break down structure and task assigned to each member, tracking progress of each task etc.

7. References

[1] Anuradha Mathrani , Sanjay Mathrani "Test strategies in distributed software development environments" School of Engineering and Advanced Technology, Massey University, Auckland, New Zealand.

[2] Lars Taxe´n " An integration centric approach for the coordination of distributed software development projects" Lars Taxe´n Department of Science and Technology, Campus Norrko¨ping, Linko¨ping University, Norrko¨ping, Sweden.

[3] Jorge Luis Nicolas Audy "Process models in the practice of distributed software development: A systematic review of the literature" Pontificia Universidade Católica do Rio Grande do Sul – PUCRS, Computer Science School, 6681 Ipiranga Avenue, Building 32, 90619-900 Porto Alegre, Brazil

[4] Zane Galviņa, Darja Smite,"Software Development Processes in Globally Distributed Environment"Scientific Papers, University of Latvia, 2011. Vol.770.

[5] Miguel Jimenez, Mario Piattini, and Aurora Vizca " Challenges and Improvements in Distributed Software Development: A Systematic Review"Hindawi Publishing Corporation Advances in Software EngineeringVolume 2009

[6] Roberto Zanoni "Project Management Model For a physically Distributed Software Development Envronment" Proceedings of the 36th Hawaii International Conference on System Sciences – 2003.

[7] Anita Sarma, Andre van der Hoek, "Palantír: Coordinating Distributed Workspaces" Department of Information and Computer Science University of California, Irvine 92697-3425 Irvine, CA USA.

[8] David K.M Mak and Philippe B. kruchten, "Task Coordination in an Agile Distributed Software Development Environment" University of British Columbia

[9] Jon Froehlich and Paul Dourish, 'Unifying Artifacts And Activities in a Visual Tool for Distributed Software Development Teams' Department of Information and Computer Science University of California, Irvine 92697-3425 Irvine, CA USA.

[10] Catherine Lowry CAMPBELL and Bartel VAN DE WALLE, "Asynchronous Requirements Engineering:Enhancing Distributed Software Development" College of Computing Sciences, New Jersey Institute of Technology, Newark, NI, 07102, USA

[11] Sigrid Goldmann, Jürgen Münch,Harald Holz,"A Meta-Model for Distributed Software Development"Universitat Kaiserslautern.

[12] L. Yu, A. Mishra, "Risk analysis of global software development and proposed solutions", Automatika 51 (1) (2010) 89–98.

[13] M. Cataldo, M. Bass, J. Herbsleb, L. Bass, "On coordination mechanisms in global software development", in: IEEE (Ed.), Second IEEE International Conference on Global Software Engineering, 2007, IEEE, Munich, 2007, pp.71–80.
[14] M. Shi," Software functional testing from the perspective of business practice",Computer and Information Science 3 (4) (2010) 49–52.
[15] R. Prikladnicki, D. Damian, and J. L. N. Audy, "Patterns of evolution in the practice of distributed software development: quantitative results from a systematic review,"in Proceedings of the 12th Conference on Evaluation and Assessment in Software Engineering (EASE '08), Bari, Italy,June 2008.
[16] Karolak, D. "Global Software Development: Managing Virtual Teams and Environments". Wiley-IEEE Computer Society Pr; 1st edition, December 27, 1998.
[17] Prikladnicki R.,Audy J., Evaristo R. "Global Software Development in Practice Lessons Learned". Software Process: Improvement and Practice.2003, pp. 267-281.

Mr. Aaditya Vishwakarma, received his B.E.degree in Information Technology from Hitkarini College of Engg. And Tech. , Jabalpur in 2009, and pursuing M.E. in Computer Science Engg(S. S.). from SRIT Engineering College Jabalpur.