The Nucleus 52, No. 3 (2015) 123-129

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)

ISSN 2306-6539 (Online)

Working Environment and Software Configuration Management Assimilation using Traceability Enhancement Technique

H. Iqbal*, A. Javed and M. N. Majeed

Department of Software Engineering University of Engineering and Technology Taxila, Pakistan

engr.hinaiqbal@gmail.com, [2]ali.javed@uettaxila.edu.pk, [3]nadeem.majeed@uettaxila.edu.pk

ARTICLE INFO

Article history :

Received : 14 January, 2015

Revised : 14 July, 2015

Accepted : 19 August, 2015

Keywords :

Artifacts,
Baselined documents,
Change management,
Software Configuration Management
Traceability
Working environment

ABSTRACT

Software Configuration Management (SCM) Systems are very useful in coordinating group effort in large and complex software systems. As a result of change in user requirement, market needs, tools, technology or new business goals emanate out, changes are continuously induced while developing the software product. For change management, Traceability technique and SCM are two prominent practices in the software development process. SCM helps in managing configuration items while traceability helps in tracing the knowledge about the configuration items. In this paper we propose a model of the SCM system with the working environment when changes are introduced in multiple artifacts and by which high quality products are developed.

1. Introduction

Software quality and productivity using SCM System depends upon organized and trustworthy change management process. With the development of various artifacts e.g. requirements specification, design, coding and testing documents etc. the need to change the system arises due to the change in tools, technologies, changing environment, error editing, and function improvement. User requests the changes and those changes are propagated to the configuration management team for approval and to notify the team about these changes. Artifacts changes rapidly due to the changes introduced in the system either from the developer side or the user of the system. Through the use of SCM system these changes can be easily propagated to all the team members and by using the version management all the versions of the system are stored that are before or after introduction of the change. Version numbers are placed with each version of the artifact which shows the order of the introduction of change.

Traceability enhancement technique is important in tracing these changes and providing correct version of the artifact to the team for developing the quality product. If any change, when implemented, causes the system to go in unstable state or affects its required functionality then through the use of configuration management and version control system it is easy to rollback to previous version. Changes can be easily traced with the help of the change traceability report which shows, who made the change and what is path of change introduction. Integrating working environment to the configuration management system helps in tracing the work/changes introduced by the team members working on the current projects.

Configuration management system is implemented for maintaining imminent change of product artifacts under control. This can be achieved by eliminating confusion and errors from the artifacts that arises when dealing with multiple versions of project artifacts. Goal of using CM system is to make sure the veracity of a product and allow more manageable development. Effective CM system entails power over unmanageable activities which require revising and exploiting multiple versions of project artifacts. Organizations should have a comprehensible defined CM process to manage unique intricacies of each product. This mingled with well-defined, renowned, accepted CM procedures and processes, and stringent devotion to them for a project team to efficiently work with CM tools. This allows for the combined ability to assemble, organize, correct, and update project artifacts and, if necessary, can move back to earlier version of products.

It is difficult to determine how artifacts are related without proper knowledge and artifact’s traceability otherwise it is complex to trace the changes and incorporating these changes in the system and difficult to ensure correctness and accuracy [1].

* Corresponding author

In this paper, we focused on incorporating working environment with CM system when multiple changes are introduced at the same time on multiple artifacts.

The rest of the paper is arranged as follows. Section II contains the previous work done by many researcher related to the SCM, the working environment and tracing changes in the working environment. Section III describes the proposed framework for SCM system and working environment. In section IV, our discussion is concluded and we explain the future research.

2. Background and Related Work

SCM is a process of controlling and tracking amendments in the software product artifacts. SCM involves revision controlling and establishing the baselines of the artifacts. If the system go in unstable state or something goes wrong, through the use of the SCM system one can easily determine what has changed and who made the change.

Traceability can be defined in a variety of ways depending upon the organization and research field. ISO defines traceability as the method that guarantees customer the ‘quality’ product. In Software industry traceability is the ability to deal with the relationships between the requirements and the artifact derived from those requirements [2]. Traceability is not only tracing the changes in requirement but it also includes tracing changes in different artifacts with respect to the time.

Change management main focus is on using SCM and development enterprises recognized its importance. As CM is an activity that remains active throughout the development process and allows efficient supervision of requirement specification changes. SCM [2] as an activity maintains reliability of software, guarantees uniformity of system quality and allows development productivity to be enhanced. SCM system is developed to support these SCM activities.

Baselined documents are developed following the proper official process of SCM. Consequent artifact changes in individual working environment escort to creation of supplementary versions with respect to the earlier versions are reflected in SCM system. The person other than the one who introduced the changes, these changes are tricky to understand from individual working environment. This leads to cynical side effects, such as decline in stability due to overlapping of tasks, additional work necessary to bring out integration in the changes made, obscurity in guaranteeing accuracy of the consequential changes. To avert from such problems, changes coordination is necessary to be implement in SCM system and individual working environment. Further, a method of displaying the combined list of tracing information for each change is necessary.

SCM baselined document are connected with the individual working environment using integrated environment. If these changes are not properly propagated, it may result in overlapping of tasks, subsequent time wasting and additional work expenses. By properly managing these changes and propagating to the team will ultimately result in quality and development productivity of the product to be improved. Quality is difficult to measure as it is intangible quantity and is often measured by customer / user satisfaction [3].

Berczuk et al. [15] classify traceability of artifact in two ways: local traceability (tracing changes within individual’s working environment) and global traceability (tracing changes that all users can share i.e. from central repository).

Kim and Youn [2] described a method in their paper for enhancing the traceability of the artifacts for providing high productivity and a quality product while maintaining and following the SCM activities to achieve their goals also. SCM system manages changes in a consistent way whereas with the integration of the individual working environment traceability of the changes also managed on the individual systems also and help to improve the quality as well as productivity in software development.

Ping and Yang [5] described in their research that software maintenance is one of the main activity after the product development. Changes may occur as a result of new customer requirements and error correction. These changes are traced in order to reduce the inconsistencies and to lower the level of confusion and misunderstanding that arises due to these inconsistent changes.

Shen and Sun [6] explained in their research that concurrent support to system development is the key to deliver high quality product in less time and for that they proposed an algorithm that is used for merging the textual information in the artifacts and through the use of SCM system communicating these changes to all the groups involved in the development of a software product.

Sarma et al. [7]; explained the importance of workspace (working environment) in software configuration management due to the early detection of the error and showing in parallel the changes to the development team and showed that it promotes self-coordination among teams.

The researchers [8] suggest that it is good to take component as the basic unit rather than taking and tracking the changes in individual files.

In large projects consisting of many phases, each phase contains many artifacts and to manage/tracing the changes in these artifacts through configuration management system and tracing how the change is propagated and how it impacts the changes introduced in the working environment. Srinath *et al*. [9]; discussed the revisions/versions of artifacts captured through configuration management tool. They design a model for semantic-based changes in artifacts in large projects.

Li and Young [10] in their research urge the importance of team work for successful and timely completion of larger and complex projects. For managing the interaction between the teams for tracking project progress, they proposed SCM model which could be used for the distributed framework in layered architecture.

When the complexity of projects for development is higher or of uncertain nature there is need for the software maintenance process because of inconsistent requirements [11].

Distributed projects [12] are of complex nature and to manage those projects we need a configuration management system. Changes in the working environment can be traced and managed using these distributed projects.

Software application's quality depends upon the quality of the artifacts. Versions of the artifacts are managed through the use of SCM system [13].

Luo et al. [14]; described the importance of change management in enterprises. They explained that change management process helps in improving the overall efficiency of the system that is by minimizing change completion time and by increase in change capacity.

If traceability of changes has not done properly then certain problems arises for example increase in number of changes, loss of knowledge due to turnover, misunderstanding and miscommunication, erroneous decisions and system quality decreases [1].

3. Proposed Framework

3.1 Change Management Process

Changes are introduced continuously in artifacts in individual working environment and to manage these changes, change management process is followed under the SCM System. Changes are mimicked in SCM's baselined documents with respect to SCM's check-in process by connecting SCM's baselined documents with the working environment's artifacts. Individuals can obtain artifact versions from SCM system using baselined document by acquiring the newly created version can connect to individual working environment and can also obtain information about changes. Fig. 1 shows the process of creating the artifact in working environment and change introduction in SCM system.

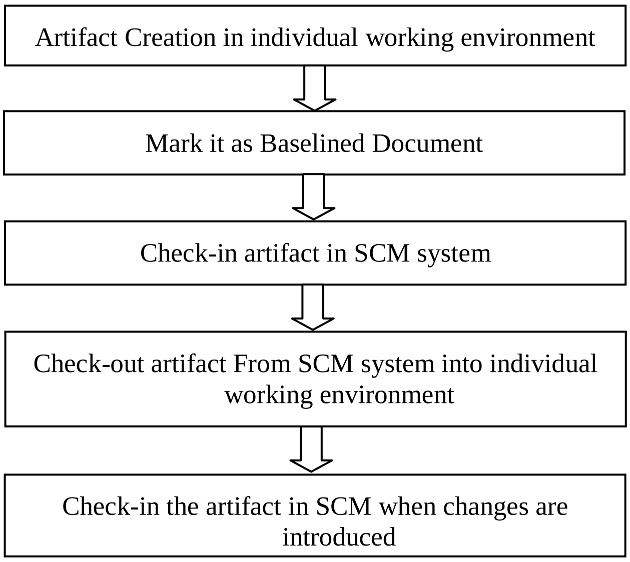
Artifact Creation in individual working environment
Mark it as Baselined Document
Check-in artifact in SCM system
Check-out artifact From SCM system into individual working environment
Check-in the artifact in SCM when changes are introduced

Fig. 1: Artifact creation and change introduction

Artifacts are created in individual working environment and they are marked as baselined document. These baselined documents are then checked-in the SCM system. When any changes are required in the artifacts, These baselined documents are checked-out from the SCM system to individual working environment. Changes are introduced in artifacts and those are then checked-in in SCM so that the artifact is available to all the team. Fig. 2 shows working environment and SCM change management process stages.

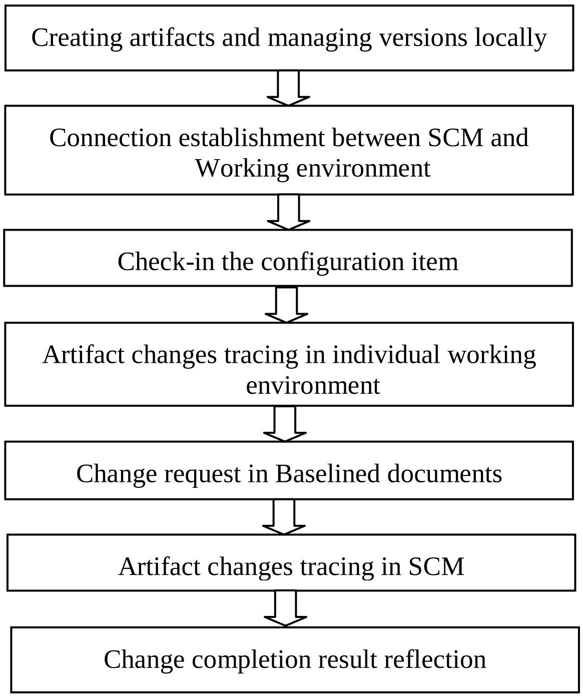
Creating artifacts and managing versions locally
Connection establishment between SCM and Working environment
Check-in the configuration item
Artifact changes tracing in individual working environment
Change request in Baselined documents
Artifact changes tracing in SCM
Change completion result reflection

Fig. 2: Change management process in SCM and the working environment

3.2 Creating Artifacts and Managing Versions Locally

In the Individual working environment multiple artifacts are created and to manage the change in these artifacts, version management method is applied in individual working environment. The change process in working environment ensues following steps.

- Creation of the artifact
- Change Requests in artifacts
- Managing the changes
- Highlighting the changes
- Managing the versions

These steps are also shown in Fig. 3. The first step in the change process involves creation of the artifact. Members of the team, while working on the project/system in their individual working environment, create different artifacts depending upon the need of creating artifact for the project or system.

Artifact contains all the user requirements fulfilling needs of the user and when these needs changes, users demand for a change in the system that results in change of the related artifacts. Environmental scanning and analysis.

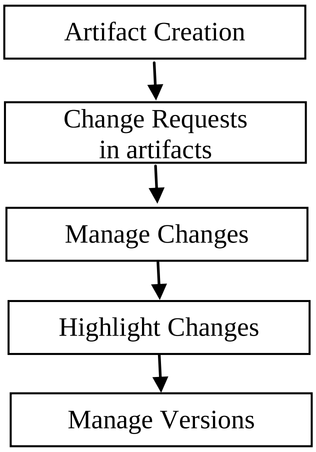

Fig. 3: Change management process in individual working environment

Whenever a new change is introduced in the system/ product to be developed or the new demand of the user arises, changes are introduced in the artifacts in individual working environment related to those changes.

For managing these changes in individual working environment, version management is used for sustaining local traceability in individual working environment in unsanctioned domain. Version management manages these changes by highlighting these changes, maintaining the versions of artifacts and note down who made the change.

Multiple artifacts are created in individual working environment. Users may request changes that affects or required changes in multiple artifacts. In order to manage such type of changes, the individuals working on their environment uses the traceability technique in order to determine the flow of change that who made the change and what is the path of change through which it is introduced in the artifact. These changes are highlighted and new version is created. If change is of higher importance or new system functionality is introduced and subversion if change is of lesser importance or only minor change in functionality arises.

3.3 Connection Establishment between SCM and Working Environment

Individuals are working on artifacts and when work is completed in the individual working environment then these artifacts are placed in the central repository using SCM system to make this available to all the members of the team working on the project. Connection is established between the artifacts of the working environment and configuration items of SCM system by either referencing the file directory of the individual working environment or directly attaching the file. When changes are introduced in multiple artifacts then all these changed artifacts are required to have a connection with the SCM system so that change is properly propagated to the entire team.

When changes are introduced in multiple artifacts, then all these changed artifacts must reflect this in the SCM system and for this purpose, these artifacts are connected to the configuration items of the SCM system. The connection can be established by either referencing the path where artifacts are placed in individual working environment or by simply place these artifacts in the SCM system.

3.4 Check-in the Configuration Item

Changes are introduced by users and they made request to introduce these changes in configuration items and for that they require approval. Request of the user is sent for approval and when that change is authorized it is introduced in the artifacts. That artifact is checked-in SCM and it is marked as baselined document. When changes are introduced in the configuration item, these configuration items after the introduction of the changes are stored as the base lined documents and the previous version is also stored for the purpose in order to move backward if the change causes disaster then to be helpful.

These baselined documents are used for global traceability and anyone in the team can access the document or the configuration item from the repository/central storage using the configuration management system. Individuals can copy the documents/configuration items in their environment and apply the changes as required and in order to place/share the document/configuration item/artifact globally to all the team members, establish the connection to the configuration management system and share documents or artifacts.

3.5 Artifact Changes Tracing in Individual Working Environment

In individual working environment, whenever a new change is introduced, artifacts are changed in the individual working environment and these changes are emulated in SCM system when they follow the proper change management process and also these changes are approved to be placed in the SCM's system repository. In the individual working environment many changes are introduced depending upon the team member wish or the need but all these changes are not introduced in the SCM system. Typical SCM system does not allow temporary changes in the artifacts while in the working environment user can make temporary changes and after the approval introduced in the SCM system. Individual working on their artifacts can make the changes depending upon their needs. They can either create branches to carry out tasks in different way or create version to carry out tasks in the same way as reflected in the SCM system.

Traceability helps in finding the artifact version that is combined with the baselined document. It also helps in tracing earlier versions of the artifacts in the working environment and also verifies those changes.

3.6 Change Request in Baselined Documents

In order to introduce changes in baselined documents, a change request is made that requires approval before introduction in the artifact or become part of the baselined document. Final approval is given by the person with the authority for approval of the changes. They check the baselined document and the artifact connected to that from the working environment containing those changes. All team members can access the baselined document by using SCM system. Users can check-out the baselined document and to trace the changes, can follow the previous version already stored on the central location.

3.7 Tracing Artifact Changes from SCM

Artifacts are stored in SCM system based on the version numbers. Artifact with the highest version number is said to be the baselined document. If the system go in unstable state or introduction of the changes lowers system performance or degrades functionality of the system, by tracing the version numbers of the artifacts one can easily move back to previous one and restore the system to the quite stable state.

3.8 Change Completion Result Reflection

From working environment artifact changes are being traced. The person responsible for approving the changes should reflect their results in the SCM systems baselined documents.

4. Integrated System (SCM and Individual Working Environment)

Integrated system is a system in which the two or more systems are combined to make a new system that can be used where these two or more systems are interlinked and affect each other. In our case, we combined the SCM system with the working environment using the traceability technique to increase the tracing of the individual changes possible and also to communicate these changes to all team members.

Individuals are working on their working environment and they have access to the central repository of the SCM system. SCM system manages the version of the artifacts. Individuals having authority can access these artifacts and the check-out these artifacts to their working environment to introduce changes. Changes can be in single artifact or in multiple artifacts depending upon the nature of the change. Single artifact changes are easy to manage as it requires changes in a single artifact. The problem arises when changes are introduced in multiple artifacts. To manage the changes in multiple artifacts, changes are introduced in multiple artifacts and these changes are then managed through the use of the version management process and tracing of these versions in SCM system's repository.

In order to prevent same task from multiple users, artifacts from individual working environment and configuration items from the SCM are traced and version management is used so that all the work with respect to time is saved and provide the facility to backtrack if any error occur or system go in unstable state. Traceability of artifacts, either locally or globally, is done to ensure that all the users in the system development process contains updated version of artifacts so that these artifacts are available for introduction of any change from the users. Team members check-out the updated versions of artifacts and start developing /implementing the changes in the system.

Whenever changes are introduced in multiple artifacts this means that these artifacts are dependent on each other i.e. any changes introduced in one artifact will result in changes in other artifact and vice versa. To manage such types of artifacts whenever change is introduced, changes are placed either in single artifact or in multiple artifacts depending upon its nature. Introduce the changes in artifacts and create the new versions of the artifacts and generate a report which shows which artifacts are changed and what changes are to be implemented.

For example consider two artifacts e.g. requirement specification document and the design document as shown in Fig. 4. If user request a change in the requirements or is demanded new functionality, this requires changes in the requirement specification

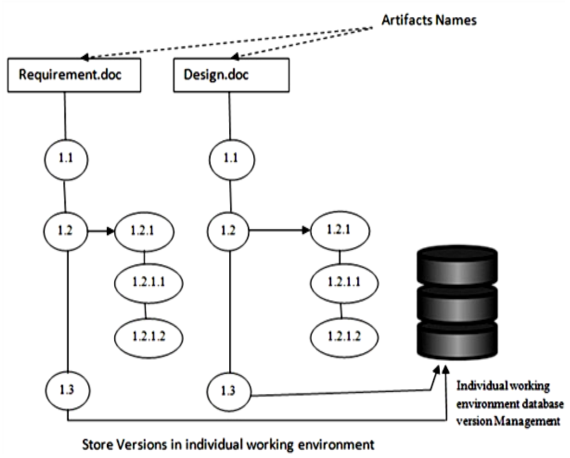
Artifacts Names
Requirement.doc
Design.doc
1.1
1.2
1.2.1
1.2.1.1
1.2.1.2
1.3
1.1
1.2
1.2.1
1.2.1.1
1.2.1.2
1.3
Individual working environment database version Management
Store Versions in individual working environment

Fig. 4: Explains the version management of multiple artifacts in the individual working environment

document. As the requirements changed, the design of the system also changes. In the working environment version management is used to manage changes in multiple artifacts. If changes in multiple artifacts occur for better results a text file is also checked-in the SCM which contains changes introduced in multiple artifacts so that change traceability is easy. All team members are notified when the changes are checked-in the SCM so that all knows about the change. Fig. 5. displays the overall model of the system when integrating SCM system with the individual working environment using the traceability enhancement technique.

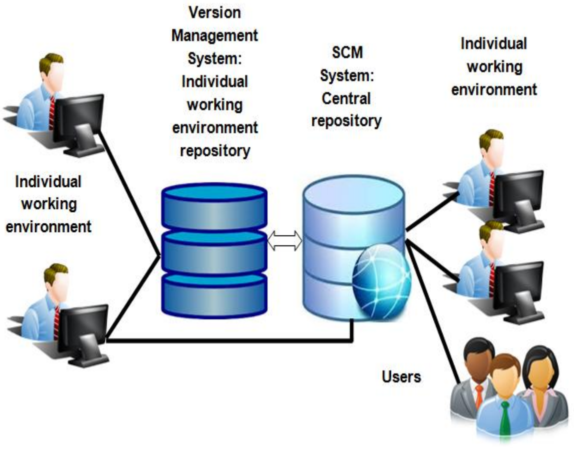
Version Management System: Individual working environment repository
SCM System: Central repository
Individual working environment
Individual working environment
Users

Fig. 5: SCM and Working Environment Integration

Individuals are working in their environment on the project under development. In the working environment the individuals are managing the versions locally and the finalized version that is marked as baselined document is then checked-in the SCM central repository. In the SCM, the versions of the artifacts are stored reflecting the changes introduced and make these versions available to all the team members and users of the system. Through the use of SCM with individual working environment helps in managing and tracing the changes in multiple artifacts and also increases the performance, reduces the chances of confusion and error rate decreases.

4. Conclusion

In our research we proposed a model for enhancing the traceability by combining working environment of individuals with the SCM system. The model is proposed keeping in mind the goals aspired by SCM in order to increase the development productivity. SCM systems follow the strict planning and control method to manage these changes whereas in individual working environment there is no strict planning and control method and changes can be freely made and the integrated environment results in increase in efficiency.

SCM system allows the global traceability whereas version management allows local traceability. Consolidating global traceability with local traceability helps in consistent change management. Our future work includes the implementation of the proposed model in real world.

References

[1] K. Mohan, P. Xu, L. Cao and B. Ramesh “Improving change management in software development: Integrating traceability and software configuration management”, Decision Support Systems vol. 45, no. 4, pp. 922–936, 2008.

[2] D. Kim and C. Youn “Traceability Enhancement Technique through the Integration of Software Configuration Management and Individual Working Environment”, Proceedings of the Fourth IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI), vol. 1, pp. 163-172, 2010.

[3] D. Zhang, Q. Gao and Z. Lin “Critical quality chain analysis and evaluation based on quality loss in service industry”, Control and Disease Conference, IEEE, vol. 1, pp. 1558-1592, 2008.

[4] T.N. Nguyen, E.V. Munson, and C. Thao “Object-oriented Configuration Management Technology can Improve Software Architectural Traceability”, Proceedings of the Third ACIS International Conference on Software Engineering Research, Management and Applications, vol. 1, pp. 86-93, August 11-13, 2005.

[5] L. Ping and L.J. Yang “A Change-Oriented Conceptual Framework Of Software Configuration Management”, 2007 International Conference on Service Systems and Service Management, vol.1, pp. 1-4, June 9-11, 2007.

[6] H. Shen and C. Sun “A Complete Textual Merging Algorithm for Software Configuration Management Systems”, Proceedings of the 28th Annual International Computer Software and Applications Conference, vol. 1, pp. 293-298, 2004.

[7] A. Sarma, A. van der Hoek, and D.F. Redmiles “A Comprehensive Evaluation of Workspace Awareness in Software Configuration Management Systems” IEEE Symposium on Visual Languages and Human-Centric Computing, 2007.

[8] L. Zhang, H. Me and H. Zhu “A Configuration Management System Supporting Component-Based Software Development” In Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC' 01), pages 25-30. IEEE Computer Society Press, 2001.

[9] S. Srinath, R. Ramakrishnan, and D.J. Ram “A Generic Model for Semantics-Based Versioning in Projects” IEEE Transaction on Systems, Man, And Cybernetics—Part A: Systems And Humans, vol. 30, no. 2, pp. 108-123, March 2000.

[10] R. Li and Z. Yong, "A New Configuration Management Model for Software Based on Distributed Components and Layered Architecture" Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003, IEEE Digital Library pp.665-669, August 27-29, 2003.

[11] I.H. Chou, C.F. Fan, "A Regulatory Software Maintenance Environment Using Agent-Based Software Configuration Management" International Conference on Dependability of Computer Systems, 2006. DepCos-RELCOMEX '06, vol. 1, pp. 264-275, May 25-27, 2006.

[12] L. Bendix, J. Magnusson, and C. Pendleton, "Configuration Management Stories from the Distributed Software Development Trenches" IEEE Seventh International Conference on Global Software Engineering, vol.1, pp. 51-55, 2012.

[13] T.N. Nguyen, E.V. Munson, J.T. Boyland, and C. Thao "Flexible Fine-grained Version Control for Software Documents" Proceedings of the 11th Asia-Pacific Software Engineering Conference, vol. 1, pp. 212-219, November 30-December 03, 2004.

[14] X. Luo, K. Kar, S. Sahu, P. Pradhan and A. Shaikh "On Improving Change Management Process for Enterprise IT Services" IEEE International Conference on Services Computing, vol.2, pp. 341-348, July 2008.

[15] S.P. Berczuk and B. Appleton, "Software Configuration Management Patterns, Effective Teamwork, Practical Integration", Addison Wesley Longman Publishing Co., 2002.

[16] Y. Ren, Q. Quan, T. Xing and X. Chen, "Fuzzy Decision Analysis of the Software Configuration Management Tools Selection" Proceeding of Third International Symposium on Information Science and Engineering, vol.1, pp. 295-297, Dec 2010.

[17] http://www2a.cdc.gov/cdcup/library/pmg/concept/cnfg_description.htm.