ISSN : 2350-0743

www.ijramr.com

International Journal of Recent Advances in Multidisciplinary Research

Vol. 04, Issue 11, pp.3042-3046, November, 2017

RESEARCH ARTICLE

SOFTWARE CONFIGURATION MANAGEMENT TECHNIQUES AND PRINCIPLES

***Vinay R. Menon, Sreehari Tharayil Sathyanathan, Ken Thomas Abraham, Jaiesh Sunil Pahlajani and Prof. Santhi H.**

School of Computer Engineering, VIT University, Vellore, Tamil Nadu

ARTICLE INFO

Article History:
Received 09th August, 2017
Received in revised form
27th September, 2017
Accepted 04th October, 2017
Published online 30th November, 2017

Keywords:

Configuration, Development,
Flexible, Management,
Semantics, Software.

ABSTRACT

Software Configuration Management (or Software Change Management) is defined as the task of tracking and controlling changes in software. It is a part of a larger field of configuration management. It helps in identifying individual elements and configurations, tracking changes, and version selection, control, and baselining. Software products are very complex to develop and are even more difficult to maintain. There are various problems faced in SCM such as balancing the costs of productions with benefits from increased sales of software products. As a solution to these problems, SCMs have a key role in handling the complexities of software variability. More and more systems are developed using components. There is a move from monolithic to open and flexible systems. In such systems, components are upgraded and introduced at run-time, which affects the configuration of the complete system. Keeping up-to-date information about which components are installed is a problem. Updating a component also affects the compatibility of the system. It is therefore important to keep track of changes introduced in the system. In the product life cycle, CM is traditionally focused on the development phase, in particular on managing source code. Now when changes are introduced in systems at run-time and systems are component-based, a new discipline, component configuration management is required. The future holds many avenues for progress in addressing technology, such as new functionality; process, such as better process support involving semantics of data; management, such as improved recognition of the need for resources and politics and standards.

INTRODUCTION

This research paper provides a detailed insight into software configuration management as a tool for evaluating and managing changes. Configuration management plans are also studied with importance where every plan is project specific and follows a certain standard. This research paper discusses the need for software configuration management in web projects so that the risk of failure is reduced by bringing up efficient management putting up short goals which ultimately pile up to the execution of the end product. Most existing SCM and version control tools that are used are focused on version control of individual files with limitations in content change and configuration management of a web project. Software architecture, management of configuration, and configurable distributed systems are three areas of research that have until now evolved separately. A single unified system that advocates the joint evolution of all these disciplines is a more appealing alternative to advance the state of the art for the same. The benefits of using a unified system model, identifying the commonalities, and finding the differences among the existing system models leads to the development of a single system that is usable in all three fields.

This research paper provides a detailed insight into software configuration management as a tool for evaluating and managing changes. Configuration management plans are also studied with importance where every plan is project specific and follows a certain standard. The basic idea behind Software configuration management is well covered including the activities and SCM models. It is specified how the evaluation of SCM process takes place and evaluating the tools for the process. The study found that agile methodologies and any associated practices are considered important and used frequently by all-sized organizations. The paper shared the experiences from CM activities in two industrial organizations over a period of more than a decade, which helped us to conclude that agile principles supported by the CM activities achieved interesting effects, both as a direct result of the CM activities and the effects of the agile practices supported the most significant one being the possibility of reduction of the overhead involved in CM activities, increase the possibility of rapid feedback through more frequent iteration deliveries and to be able to control more rapid change leading to increased ROI of the software project. The outcome showed a unanimous agreement that agile methods helped them become more flexible and adaptive to change. Although large organizations are still the highest consumers of traditional processes such as the SCM process, an increasing number of medium and small organizations are adopting a customized version of SCM

***Corresponding author:*Vinay R. Menon,**
School of Computer Engineering, VIT University, Vellore, Tamil Nadu

process. It was demonstrated that the comparison of configuration management automation implementation time using the old methods, when existing solutions have not been re used.A second hypothesis was proven by comparing configuration management automation timing after introducing a new methodology for different projects; a gain of 9 percent was noted. In turn, the longer solutions exist in the database and develop, then they become more stable and automation implementation time decreases by 96 percent. The concept of agile SCM is briefly discussed as well as SCM evaluation and improvements. This paper discusses about maintaining integrity as software configuration systems evolve. After analyzing the paper three key requirements for component configuration management were identified which were component description, maintain history, and ability to compare two components. Verifying through proofs, the notion of upward compatibility can be believed to include notions which preserves various proofs for a certain class of changes. NuMIL is also used in the research to design the software architecture.

A more thorough study can be carried on how structural and functional changes affect the integrity of system as a future work. The importance of software configuration management in software development is discussed. SCM is divided into activities and the process is carried out through these activities such as configuration, identification, configuration control, configuration status accounting and audits. It discusses how agile methodology plays a role in SCM. The history of software configuration management is discussed from 1950's till 2000. This paper specifies the various user concepts which are present in the existing configuration management systems. The many concepts implemented by various CM systems are represented in a spectrum. From the paper, it raises concerns about the development of CASE tools and how it should incorporate CM within the tool itself. The SCM model known as Capability Maturity Model is also discussed for the development of products and services. Tools for SCM such as IBM Rational ClearCase, Perforce, Microsoft Visual SourceSafe, etc. are mentioned as well. The concept of dynamic change management is also discussed.

History

The history of software configuration management is discussed from 1950's till 2000. The SCM model known as Capability Maturity Model is also discussed for the development of products and services. Tools for SCM such as IBM Rational ClearCase, Perforce, Microsoft Visual SourceSafe, etc are mentioned as well. The concept of dynamic change management is also discussed. The development of a software configuration manager is one of the solutions for problems concerning development and maintenance of more complicated embedded systems. The problem is to find the right procedures for applications and further development of the same in practice. The main target is to have a standardized version control and software manufacturing practices, despite change control solutions assuming the key role in future improvements. This research paper provides a detailed insight into the version models for SCM. Version model defines the items to be versioned, the properties shared by all versions of an item and the differences between them. The use of database technology is supposed to help in the building of such configuration management software. The different activities involved in SCM such as auditing, accounting, controlling and process are briefed. They considered software configuration management with respect to configuration selection, configuration verification, and configuration update, because it was believed that these aspects are the most relevant to get an efficient configuration management of automotive software. Configuration management with respect to configuration selection, configuration verification and configuration update was discussed. Explicit configuration selection reduced the number of configurations used in the cars due to the limited number of releases.

Basic Activities in an SCM

The importance of software configuration management in software development is discussed. SCM is divided into activities and the process is carried out through these activities such as configuration, identification, configuration control, configuration status accounting and audits. It discusses how agile methodology plays a role in SCM. The different activities involved in SCM such as auditing, accounting, controlling and process are briefed. The concept of team in SCM is introduced which is aimed at saving valuable developer time. Deployment, which is a new field of SCM which was assumed under maintenance is well defined. The scope of further research on the topic is also given good importance.

Software configuration management can be segregated into

- Version control,
- Document control,
- Change management
- Build management, and
- Release control.

Verifying through proofs, the notion of upward compatibility can be believed to include notions which preserves various proofs for a certain class of changes. NuMIL is also used in the research to design the software architecture. A more thorough study can be carried on how structural and functional changes affect the integrity of system as a future work. Traditional SCM practices affect the agile development process for XP software teams. It is realized that combining both practices in the development process will not affect the SCM practice but make the agile process less agile. The SSCM approach discussed in this paper currently finds high interest on the user and manufacturer side in the scope of high security software products, and leads to busy developmental activities of the tool manufacturers. Following SCM practices for XP software, not only adds safety to the process but also make every process heavily dependent on the team. The idea of this paper is to document the process and the result of deploying a configuration management tool. It also describes about the various factors, technical considerations as well as social issues which are responsible during the process. Describing the rationales for each practice, the benefits for the same, they are traces to their underlying principles which have a greater acceptance than practices; it also used the concepts of the service oriented approach - loose coupling, abstraction, reuse, heritability, autonomy to name a few. The model then applied these two ideas to make a better configuration to optimize the architecture of the software. It also provided a collection of services and improvement of the configuration of the software architecture. The provided information should permit the software professional to analyze these principles and practices considering specific constraints to develop an overall optimal economic solution. The primary issue therefore becomes what practices one can select that provide the greatest reward risk ratio given the existing constraints. It is believed that configuration management is an important technology to deploy in any environment. A service that permits reconfiguration delegation should be considered as a future work. This paper focusses on creating a suitable SCM model for the CBSD based on the objects analyzed. Here, a component based SCM model called JBCM is used which is divided into three parts.

Models

This research paper provides a detailed insight into the version models for SCM. Version model defines the items to be versioned, the properties shared by all versions of an item and the differences between them. The use of database technology is supposed to help in the building of such configuration management software. This paper has discussed the recognition of a new standard model for software configuration management and how it makes possible pragmatic software configuration management in the E-World. It has also described the old theoretical model of software configuration management and given a new definition to the above stated points. The principle of modularization is one of the main techniques engineers use to tame the complexity of programming. Focusing on one of these complex areas, namely software configuration management and outlining how modularization is natural and powerful. Analysis is conducted partly based on experiences from case studies where small to medium sized development projects are using a prototype tool that supports modular software configuration management. This research paper provides an application of software configuration management for flight critical digital control systems of advanced aircrafts. Software configuration management is used to ensure efficient flight system development which is of complex nature. Different forms of software configuration models are discussed for this application. The software configuration management is discussed here in terms of functionalities. SCM is discussed as an engineering discipline with its academic impact being given importance. It gives us a glimpse of the evolution of SCM throughout the years discussing the upbringing as well as the downfalls of the field.

It also gives a heavy emphasis on research as being a constant and essential methodology in SCM. A version model can be developed which integrates extensional and intentional versioning, state-based and change-based versioning, revisions and variants, construction of source and derived versions, as well as workspaces and long transactions into a coherent framework. The framework should be expected to provide a customizable model to suit the needs of a specific application, not just "the" model. It has been expected that the use of database technology will aid the building of such configuration management software. This research paper discusses the need for software configuration management in web projects so that the risk of failure is reduced by bringing up efficient management putting up short goals which ultimately pile up to the execution of the product. Most existing SCM and version control tools that are used are focused on version control of individual files with limitations in content change and configuration management of a web project. Maintaining integrity as software configuration systems evolve. It was demonstrated that the comparison of configuration management automation implementation time using the old methods, when existing solutions have not been re used. A second hypothesis was proven by comparing configuration management automation timing after introducing a new methodology for different projects; a gain of 9 percent was noted. In turn, the longer solutions exist in the database and develop, then they become more stable and automation implementation time decreases by 96 percent. Certain proposals for improvements of the process were specified. This paper also brought out some topics in Component-based Software Engineering which are still open and not completely explained. Further, a development process was discussed, and some problems, related to lack of information about the component properties, are given importance. Certain advice was given on how to minimize the risks of using components.

Configuration and modularization of an Scm

Modulization is one of the most important techniques used to simplify the complexity of software. Managing configurations and source code with different structures is error prone and therefore introducing modules in SCM becomes helpful. There is a strong influence of object-oriented programming in this type of model. The ability to handle a large module as a single unit becomes easier. This paper discusses on how dynamic service composition from service components, dynamic adaptation of service components with multiple classes of service, and dynamic evolution of network management software acts as a challenge to SCM. The paper also focusses on increasing flexibility and availability; at the same time examine scalability and performance issues. It is believed that one of the main challenges for SCM in the future will be its integration with other management. Modulization is one of the most important techniques used to simplify the complexity of software. Managing configurations and source code with different structures is error prone and therefore introducing modules in SCM becomes helpful. There is a strong influence of object-oriented programming in this type of model. The ability to handle a large module as a single unit becomes easier. Consistency maintenance is performed, concurrency strategy is brought up and horizontal and vertical merging is done before the final model is presented. As future research, component merging support should be introduced at the syntax level, multiple support versions should also be made available. A lack of coordination creates dependency delays among project developers, leading to increased time required in completing MRs in a GSD.

In addition, the lack of coordination can also result in each site working in a different software configuration management, which causes process management issues in GSD. These issues arise in a collocated environment; however, they are exacerbated by coordination and group awareness complexities. The introduction of tools and techniques not properly tied to a well-defined software process can limit their effectiveness. The study found that agile methodologies and any associated practices are considered important and used frequently by all-sized organizations. The outcome showed an unanimous agreement that agile methods helped them become more flexible and adaptive to change. Therefore, it mentioned the most important hashing techniques as open hashing, separate chaining and minimal perfect hashing. The hashing techniques utilizing large data sets were evaluated. The hash functions were compared based on time taken to build the data structure, performing only successful lookups and performing only unsuccessful lookups. The results indicated that minimal perfect hashing clearly outperforms the other presented hashing techniques. Although large organizations are still (as of the date of the paper) the highest consumers of traditional processes such as the SCM process, an increasing number of medium and small organizations are adopting a customized version of SCM process for their specific software development needs.

Impact of Software Engineering Research on SCM

The software configuration management is discussed here in terms of functionalities. SCM is discussed as an engineering discipline with its academic impact being given importance. It gives us a glimpse of the evolution of SCM throughout the years discussing the upbringing as well as the downfalls of the field. It also gives a heavy emphasis on research as being a constant and essential methodology in SCM. The challenges for future work includes finding the appropriate information about the components needed in different phases of a software lifecycle. These components provides information that is easily identifiable and selectable as proper candidates during the requirements and design phases; efficiently identifying configurations of component based systems and using this information to make it possible to safely and efficiently change the configurations. The concept of team in SCM is introduced which is aimed at saving valuable developer time. Deployment, which is a new field of SCM which was assumed under maintenance is well defined. The scope of further research on the topic is also given good importance.

It avoids these problems by

- Enabling developers to detect emerging conflicts.
- Providing the developers, the opportunity to self-coordinate

The given approach contributes in the given manner

- It breaks the effects of workspace isolation that occur when a configuration management system is used by transmitting information about changes across workspaces.
- It brings conflict detection to an earlier point in development

Configuration tool helps the developers to interact between one another. She looks at formal procedures that is present within the CM tool and check how it is used by developers. From the research, it is understood that there is no total solution to resolve difficulties of team coordination and more work can be done to find a mechanism to fix this problem. Making use of classes, functions and inheritance similar to OOPS we learnt. Smalltalk is used as a model medium due to its high flexibility and power. The model developed however is not suitable for large data storage and retrieval. In the future, more features can be added to the model which is derived from other models such as Gypsy, EPOS and Orion. Abstract SCM tools are the real process-centered tools due to their ability to model, support and enact the processes by which all software developers are supposed to manipulate the product. It also investigated the application of process-mining techniques in the software development domain. The suitability of software development support tools, i.e., SCM tools, to provide process data was addressed. Building software systems with reusable components brings many advantages. The development becomes more efficient, the reliability of the products is enhanced, and the maintenance requirement is significantly reduced. Designing, developing and maintaining components for reuse is, however, a very complex process which places high requirements not only for the component functionality and flexibility, but also for the development organization.

Conclusion

This research paper discusses the need for software configuration management in web projects so that the risk of failure is reduced by bringing up efficient management putting up short goals which ultimately pile up to the execution of the product. Most existing SCM and version control tools that are used are focused on version control of individual files with limitations in content change and configuration management of a web project.

REFERENCES

"A Case Study in Configuration Management Tool Deployment", Narayan Desai, Rick Bradshaw, Scott Matott, Sandra Bittner, Susan Coghlan, RémyEvard, Cory Lueninghoener, Ti Leggett, John-Paul Navarro, Gene Rackow, Craig Stacey, and Tisha Stacey – Mathematics and Computer Science Division, Argonne National Laboratory, 2005

"A Component-Based Software Configuration Management Model and Its Supporting System", MEI Hong, ZHANG Lu and YANG Fuqing, Department of Computer Science and Technology, Peking University, Beijing 100871, P.R. China, 2000

"Agile Software Configuration Management", Andreas Back, February 2006

"An Object-Oriented Model of Software Configuration Management", Hal Render and Roy Campbell, 1991

"Concepts in Configuration Management Systems" , Susan Dart, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA. 15123-3890 USA, 1991

"Maintaining Configurations of Evolving Software Systems", by K. NARAYANASWAMY AND WALT SCACCHI, MEMBER, IEEE, 1987

"Software Configuration Management Related to Management of Distributed Systems and Services and Advanced Service Creation", Vladimir Tosic, David Mennie, Bernard Pagurek, Network Management and Artificial Intelligence Lab Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, 2003

"Using a Configuration Management Tool to Coordinate Software Development", Rebecca E. Grinter Computers, Organizations, Policy, and Society Department of

Information and Computer Science University of California, Irvine, 1995

Agha Salman Haider, &quot; Basic Activities of Software Configuration Management&quot;, Senior Lecturer, Jazan University, Saudi Arabia, 2014, ISSN:2349-3224

Anita Sarma,” Palantír: Enhancing Configuration Management Systems with Workspace Awareness to Detect and Resolve Emerging Conflicts”, UNIVERSITY OF CALIFORNIA, IRVINE

Antti Kokkonen, &quot; Evaluating the Software Configuration Management Process&quot;, Department of Computer and Information Sciences, University of Tampere,2003

ArtursBartusevics, "The Development and Implementation of Model Driven Software Configuration Management Solutions" RIGA Technical University, Latvia, 2007

Bernhard Grill,” A Survey on Efficient Hashing Techniques in Software Configuration Management” Vienna University of Technology Vienna, Austria

Bersoff, Edward H. "Elements of software configuration management." IEEE Transactions on Software Engineering 1 (1984): 79-87.

Buckley, Fletcher J. "Implementing configuration management. Hardware, software, and firmware." Los Alamitos, CA: IEEE Computer Society Press and Piscataway, NJ: IEEE Press, c1996, 2nd ed. (1996).

Henrik Baerbak Christensen, "Modularization of Software Configuration Management" Centre for Experimental Computer Science, University of Aarhus, Denmark, 1999

Hoek, Andre van der; Heimbigner, Dennis M.; and Wolf, Alexander L., "Software Architecture, Configuration Management, and Configurable Distributed Systems: A Menage a Trois ; CU-CS-849-98" (1998). Computer Science Technical Reports. 798

Crnkovic I. 1991. "Large Scale Software System Management" Doctoral Thesis Department of Electrical Engineering, University of Zagreb.

Imran Ali Qureshi, Asif Iqbal Paracha, Saqib Afzal, and ShahzadRafiq,” Key Requirements for Component Configuration Management (CCM)”, Mohammad Ali Jinnah University, (MAJU), Islamabad, Pakistan

Ivica Crnkovic, Magnus Larsson, Frank Lüders,” The Different Aspects of Component Based Software Engineering”, Mälardalen University, Department of Computer Engineering, 721 23 Västerås, Sweden

Jacky Estublier(Grenoble University), David Leblang (Massachusetts Institute of Technology), Andre Van Der Hoek(University of California), ReidarConradi(NTNU), Geoffrey Clemm(Rational Software), Walter Tichy(Universitat Karlsruhe) and Darcy Wiborg-Weber(Telelogic), &quot; Impact of Software Engineering Research on the Practice of Software Configuration Management&quot;, ACMJ091-01,2005

James E. Tomayko & quot; Software Configuration Management & quot, Software Engineering Institute, Carnegie Mellon University,1990, Module SEI-CM- 4-1.4

Jana Šamalíková, Rob Kusters, Jos Trienekens, Ton Weijters, IS,” Data From Configuration Management Tools As Sources For Software Process Mining”, IE&IS University of Technology Eindhoven Eindhoven, The Netherlands

Jens Norin, Daniel Karlström,” Lean Configuration Management”, Softhouse Consulting, Stormgatan 14, SE-211 20 Malmö

Karol Fruhauf and Andreas Zeller,&quot; Software Configuration Management: State of the Art, State of the Practice & quot;,1999

Klaus Keus, Thomas Gast,” Configuration Management in Security related Software Engineering Processes”, Bundesamtfür Sicherheit in der Informationstechnik Postfach 20 03 63, D - 53133 Bonn

Lubkin, David C., et al. "Heterogenous software configuration management apparatus." U.S. Patent No. 5,339,435. 16 Aug. 1994.

Michael K. Jones,” Software Configuration Management for the Web”Director of Quality Assurance, TransactPlus,Inc. Assistant Professor, Western International University, Phoenix, Arizona, USA

Paul L Bannerman, "Software Configuration Management in Global Software Development: A Systemic Map" School of Computer Scuence and Engineering, University of NSW, 2003

RazieAlidoosti, ShahrouzMoaven and JafarHabibi,” SERVICE ORIENTED CONFIGURATION MANAGEMENT OF SOFTWARE ARCHITECTURE”, Department of Computer Engineering, Sharif University of Technology

ReidarConradi(Norwegian University of Science and Technology, Trondheim) Bernhard Westfechte l(Aachen University of Technology), &quot;Version Models for Software Configuration Management&quot;, ACM Computing Surveys, Vol. 30, No. 2, June 1998

Shukor S. M., Paul L Bannerman, Mark Staples, "Software Configuration Management in Global Software Development" School of Computer Science and Engineering, University of NSW, Australia, 1995

Taramaa, Jorma. Practical development of software configuration management for embedded systems. Espoo 1998. Technical Research Centre of Finland, VTT Publications, 366, 147 p. +108 p

Tien N. Nguyen, Ethan V. Munson and Cheng Thao,&quot; Fine-grained, Structured Configuration Management for Web Projects&quot;, Department of EECS, University of Wisconsin, Milwaukee, ACM 1- 58113-844-X/04/0005,May 2004

Ulf Asklund, and AnnitaPerssonDahlqvist. Implementing and integrating product data management and software configuration management. Artech House, 2003.

Usman K. Durrani, Joan Richardson, John Lenarcic, "Adaptable Software Configuration Management: An Investigation on Australian Agile Software Development Organizations", RMIT University, Melbourne, Australia, 2013

Yi-Jing Lin and Steven P. Reiss, &quot;Configuration Management in Terms of Modules&quot;, Department ofComputer Science, Brown University,2005, DOI: 10.1007/3-540- 60578-9_8

****