International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 2
ISSN: 2088-5334

s-SCMM: A Simplified Software Configuration Management Model for Software Services in Public University

Zulaikha Mohamad[a], Jamaiah Yahaya[b1], Aziz Deraman[c], Zulkefli Mansor[b2]

[a] *Information Technology Centre, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia*
E-mail: zulaikha@ukm.edu.my

[b] *Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia*
E-mail: [1]jhy@ukm.edu.my; [2]kefflee@ukm.edu.my

[c] *Faculty of Ocean Engineering, Technology and Informatics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia*
E-mail: a.d@umt.edu.my

***Abstract*— Software Configuration Management (SCM) is a process that supports Software Development Life Cycle (SDLC), which benefits project management, development activities, maintenance activities, quality assurance activities, and the end users of the software product. However, findings from series of interviews have proven that SCM is difficult to operate and to be implemented by the practitioners in public universities specifically in Malaysia. Some of the SCM activities could be aided by using the SCM automated tools, however not all practitioners opted that due to inflexibility of the tools. Theoretical and empirical study have identified and revealed the current practices, success factors, essential elements, and components of SCM that were used frequently and beneficial among software practitioners. The empirical study also found that the main components of s-SCMM are process, SCM tool, and human. The s-SCMM was proposed and developed by considering the main components: process, tools, human, and success factors. The s-SCMM model was validated through expert review. As the result, a Simplified SCM Model (or s-SCMM) was proposed and developed to help the execution and implementation of SCM in public university environment. The s-SCMM model is simplified by focusing on change management, version control, system building, and release management. Therefore, s-SCMM will become a platform to assist total implementation of SCM in organization. By using this model, the preparation and usage of SCM artefacts could become more effective, efficient, and systematic.**

***Keywords*— software configuration management; SCM model; simplified software configuration management; public university; qualitative interview.**

I. Introduction

Software is used as a tool and intermediary for supporting operation in organization. A team or an individual within or outside the organization could develop the software and the operation in the organization could be affected and interrupted, if the supporting software was not functioning well or in an optimum state. Through this aspect, it is essential for the organizations to have and maintain good quality software which is also easy to maintain throughout the life cycle [1], [2]. Previous studies revealed that a good quality software hold several quality attributes [1], [2] such as reliability, efficiency, flexibility, integrity, maintainability, portability and so on. Maintainability in software quality closely related to software configuration management which acts as the change management and control in the development and operation phase.

For the last decade, studies have shown that software has gone through series of changes throughout its life span to ensure continuous relevancy of the software in the operating environment [3], [4]. This level of change continues to grow in software industry. Therefore, it has becoming a necessity for the practitioners and developers to ensure the continuous development and maintenance throughout its life cycle in order to reduce and save in human resource and cost.

Software Configuration Management (SCM) is a discipline to control evolution of complex system, manage artefact changes, ensure the accuracy and consistency of system [5], [6] by allowing only valid, predicted, and tested item to the final version of the system [7]. Software changes can occur at any time throughout it life cycle, thus the SCM activities are needed to 1) identify change, 2) control change, 3) ensure change is done appropriately, and 4) report change. IEEE defines SCM that supports the requirement management, design, implementation, integration, verification, release, operation and maintenance, and project management [8]. It consists of a set of activities to manage changes during the life span of the software.

SCM was developed in 1950 by the USA manufacturing industry for controlling documentation of missile manufacturing [9], [10]. The evolution continued in 1970s where SCM was developed as an independent discipline with the emergent of SCM tools such as Source Code Control System (SCCS) and Revision Control System (RCS). Now many SCM tools are available as an open source software as well as a license software such as Git, Subversion, Jerkins, Bamboo and etc. [7]. SCM was initially used to control large and complex system software but later was growing to control diverse and distributed software project. The SCM implementation is diverse between organizations as it depends on organizations' requirements [11]. The existing SCM models have different approaches and focused on specific groups.

The proposal to apply SCM through multiple models integration [12] requires the software practitioners to fully understand the entire models involved. The different approach of SCM models includes the Life Cycle model that can be adapted to the development methodology [13], the GALO method to manage knowledge more efficiently in SCM [14], and to integrate configuration management with specific model-based cloud management [15].

The adaptation of SCM towards software development has changed and it aligns with the transformation of software development approach and methods. The traditional SDLC for software development is now being switched to Agile and Prototyping approach. Leon (2015) stated that agile method does not include and integrate with SCM comprehensively. There is still future works for this integration to be done [9].

Several SCM activities can be assisted by using SCM automation tools, however it was not fully used and utilized by the practitioners due to complexity of the process [16], [17]. Furthermore, SCM process was considered as an extra task and burden to the practitioners [9], [18] and therefore, contributing to the factor of SCM's implementation failure. In general, there is no SCM model that easy enough to be adopted and practiced by the software practitioners. Hence, the objective of this paper is to propose and develop a Simplified SCM Model (s-SCMM) to help the execution of SCM in public universities environment. The s-SCMM model is simplified by focusing on change management, version control, system building, and release management.

In view of the above, this paper is organized as follows. Section II discusses the theoretical study, which is based on critical literature review. It also elaborates the method used in empirical study to identify elements related to implementation SCM in public university. Section III presents the results and discussion and lastly, section IV concludes the findings and contribution of this research by a conclusion.

II. Materials and Methods

The development of s-SCMM was carried out by analyzing and evaluating the aspects of the theoretical and empirical study.

A. Software Configuration Management Standard

From theory, we have studied the generic process from the SCM model, the first model is ISO 10007:2003 Quality Management Systems - Guidelines For Configuration Management [19], the second model is IEEE Std-828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering [20], and the third is a software process improvement model, Capability Maturity Model Integration (CMMI) [21]. These three SCM models were used as our baseline of this research.

ISO 10007:2003 Quality Management Systems - Guidelines for Configuration Management [19] offers guideline in SCM implementation in organization. It applies from product proposal, requirement, design, development, installation, operation, maintenance and product disposal.

IEEE Std-828-2012 is IEEE standard for configuration management in systems and software engineering [20]. The IEEE SCM is applicable for any type of system software and it covers configuration activities in SDLC phases including identification of configuration items, configuration control management, account status, configuration audit and release management [20].

CMMI was proposed and developed by the Software Engineering Institute (SEI) for developing the principles and practice in software engineering. It acts as the main reference in software engineering, computer security and process improvement [21]. The development of CMMI is applicable in assessment and evaluation for software process improvement. CMMI offers five levels of maturity in software development by organization: level 1 (initial), level 2 (managed), level 3 (defined), level 4 (quantitatively managed) and level 5 (optimizing). Configuration management is included in level 2 of CMMI where SCM activities are considered as part of project management process. It supports other processes such as items configuration, configuration control, account status and audit.

B. Research Approach

This research was conducted in four main phases: theoretical study, empirical study, model development and model validation. Theoretical study involved literature review in the identified problems and related background works. This was carried out by reviewing journal papers, proceedings, white papers, theses and research reports.

The second phase of this research was the empirical study. We adopted the qualitative interview method using stratified sampling to identify selected informants in this study. Before the actual interviews were conducted, a pilot study and content validation of the interview questions were carried out to ensure the validity, reliability and appropriateness of the interview questions.

The third phase of this study was model development. The proposed model was developed and constructed based on the inputs from theoretical and empirical study. The baseline models and frameworks of the proposed simplified SCM model were ISO 10007:2003 Quality Management Systems -

Guidelines for Configuration Management, IEEE Std-828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering and Capability Maturity Model Integration (CMMI) as discussed in previous section.

The fourth and last phase of this study was model validation, which was carried out through expert review approach. Three experts were identified and invited to contribute in this research as expert panels. The s-SCMM was refined and finalized after the expert's comments and feedbacks.

C. The Empirical Study

Public University (later in this paper will be referred as UA) in Malaysia are the universities that are supported by the Malaysia Ministry of Education and currently there are 20 public universities with different categorizations. Mostly all public universities have their own department or center that responsible for managing information technology infrastructure in the UA. Thus, the departments and centers comprise of software development team and they develop software in-house. It was discovered that not all UA practices SCM inclusively [22], [23].

Qualitative interview study was conducted in Malaysia, which involved Centre of Information Technology and Communication of five public universities and one teaching hospital. The empirical study was conducted involving 12 interviewees among UA software practitioners. The main objective of this qualitative interview was to identify essential components and practices among software practitioners in UAs. Implementation success factors for SCM in public universities were also being asked and identified during these interview sessions. Criteria for informant selections were practitioners who worked at the ICT department of UA, involved in the in-house software development with minimum 5 years' experience. These criteria were considered as important to ensure the informant's answers were precise and accurate based on their knowledge and experience.

The summary and demography of the informants are shown in Table I.

TABLE I
INFORMANT DEMOGRAPHY

Item		Frequency	Percentage
Sex	Male	5	42%
	Female	7	58%
Age (years)	30-39	7	59%
	40-49	4	33%
	50-60	1	8%
Position in Organization	Project Member	1	8%
	Project Leader	1	8%
	Head of Unit	2	17%
	Head of Section	8	67%
Year experience	6-10 years	2	17%
	11-15 years	5	42%
	16-20 years	4	33%
	>20 years	1	8%
Expertise	Project management	12	100%
	Software development	12	100%
	Strategic management	2	10%
	Database management	3	30%
	Others	2	20%

The analysis concluded that there were three main elements relevant with theory exploration supported with the success factors for SCM implementation. The three elements are process, SCM tool, and human. The following sub sections discuss the identified elements and the success factors for SCM implementation.

1) *Process:* In theory, SCM process is categorized into six main generic process: SCM planning, configuration identification, configuration control, SCM account status, SCM audit, and system building and release [19]–[21]. The empirical study has found that there are four sub-processes from the generic processes that are highlighted as important to the practitioners. The sub-processes are change management, version control, system building, and release management. Table II shows the comparison of theoretical findings and empirical study of SCM process. Also included in the table are the percentage of practices of each process, and whether the process will be selected to be part of the proposed model. This study has revealed that there are generic processes that are not selected due to lack of practices among informants in UAs (only 17% practiced). It also reveals four processes that been selected for the proposed model with 100% practices rate in UAs. As shown in Table II, the processes are:

- Change Management
- Version Control
- System Building
- Release Management

TABLE II
PROCESS ELEMENT COMPARISON BETWEEN THEORY AND INFORMANT PRACTICES IN UA

SCM Generic Process	**Theory**			**Empirical**	**Proposed Model Selection**
	ISO	IEEE	CMMI	Practice in UA	
SCM Planning	✓	✓	×	17%	No
Configuration Identification	✓	✓	✓	17%	No
Configuration Control					
- Change Management	✓	✓	✓	100%	Yes
- Version Control	✓	✓	✓	100%	Yes
SCM Account Status	✓	✓	✓	17%	No
SCM Audit	✓	✓	✓	17%	No
System Building and Release					
- System Building	×	✓	×	100%	Yes
- Release Management	×	✓	×	100%	Yes

The discussion about each selected SCM process and later will be known as component, is as follows:

- Change Management: The change management sub components were derived by comparing the theoretical and empirical findings. The listed activities are shown in Table III and are related to change management, which are change request (CR) application, CR selection, CR evaluation, CR development, CR testing, CR preparation for release, and CR closure. All the CR sub components are selected for the proposed model except for the CR selection and CR closure as demonstrated in Table III.

TABLE III
CHANGE MANAGEMENT SUB COMPONENT COMPARISON

Change Management Sub Component	Theory	Empirical	Model Selection
CR application	✓	✓	Yes
CR evaluation	✓	✓	Yes
CR selection	✓	×	No
CR development	✓	✓	Yes
CR testing	✓	✓	Yes
CR preparation for release	×	✓	Yes
CR closure	✓	×	No

- Version Control: The version control in this research is focusing on the source code version control of which can be controlled with the aid of SCM tools, or known as version control software. However, this study discovers that there are still a few UAs manage version control manually; e.g. by adding comments to the changes of codes, and backup each set of source code within certain time interval and store the backup in a server. The processes involved in the version control are compared between theory and empirical findings as demonstrated in Table IV. In the table, also shown the sub components that are selected for the model development. The first process for version control is to determine the item configuration. After that, the source code is checked based on codes repository before the programming took placed. After the programming is done, the source code will be checked against the codes that stored in the repository. At this point, the version control software will generate the new version for the codes.

TABLE IV
VERSION CONTROL SUB COMPONENT COMPARISON

Version Control Sub Component	Theory	Empirical	Model Selection
Configuration item determination	✓	✓	Yes
Comments on source codes	×	✓	No
Checked out source codes	✓	✓	Yes
Programming	✓	✓	Yes
Checked in source code	✓	✓	Yes
New version generation	✓	✓	Yes

- System Building: The comparison between the theory and empirical findings of the system building are shown in Table V. The empirical finding shows that all informants of UAs practice the system building's activities aligned with the sub components defined in theory. The system building starts by merging the system component that consists of source code files and related library files. The source codes are needed to be compiled into certain file format before the release process is carried out. Normally, off-the-shelves (OTS) software and Integrated Development Environment (IDE) support this process. The system component is checked and revised later to ensure the accurate version is selected for release. Then, the system packaging will take place to produce file that can be delivered to the users without exposing the actual source codes.

TABLE V
SYSTEM BUILDING SUB COMPONENT ELEMENT COMPARISON

System Building Sub Component	Theory	Empirical	Model Selection
Component merging	✓	✓	Yes
System component revision	✓	✓	Yes
System packaging	✓	✓	Yes

- Release Management: The release management process is also supported by OTS and IDE. Table VI lists the comparison of release management sub processes, which are release-planning, determination of release (major or minor release) and distribution. The release management process starts with the release planning which involves the description of the functional parts of the release and related rationale. At this point, the reference preparation for product content, release schedule, release impact, and release notification are implemented. Next, the system version is determined by setting a new major or minor version. Finally, software distribution is accomplished by distributing to users through updates on the application server and sending notification to users.

TABLE VI
RELEASE MANAGEMENT SUB COMPONENT ELEMENT COMPARISON

Release Management Sub Component	Theory	Empirical	Model Selection
Release planning	✓	✓	Yes
Determine the major and minor software release version	✓	✓	Yes
Software distribution	✓	✓	Yes

2) SCM Tools: The empirical study and analysis discussed in this paper has discovered that the SCM tools used in public universities in Malaysia consist of three (3) type, which are:

- In-house developed application (IH)
- Off-the-shelves software (OTS)
- Integrated development environment (IDE).

The comparison between the theory and empirical findings on the SCM tools are shown in Table VII. Theoretical study makes known that for the change management process; the SCM tools are available through in-house development and OTS software. On the other hand, the empirical study reveals that only in-house developed software that could meet the UA's change management requirement. For the version control, both studies agree that only OTS software available for that process. For the system building, from the theory perspective, the in-house development and OTS software are available for the process, but the empirical only stated that only the IDE is used for system building. Most IDEs have system building and release management features. Finally, this study also discovers that the release management uses OTS and IDE as stated in theory as well as from empirical findings. Since all type of SCM tools contribute to managing each process component, all three types of SCM tools are selected for included in the proposed model.

TABLE VII
SCM TOOLS COMPARISON

Process Component	**Theory**			**Empirical**			**Proposed Model Selection**
	IH	**OTS**	**IDE**	**IH**	**OTS**	**IDE**	
Change Management	✓	✓		✓			Yes
Version Control		✓			✓		Yes
System Building		✓	✓			✓	Yes
Release Management		✓	✓		✓	✓	Yes

IH: In-house development; OTS: Off the shelf;
IDE: Integrated Development. Environment

3) Human: For human element, the comparison between theory and empirical is shown in Table VIII.

TABLE VIII
HUMAN ELEMENT COMPARISON

SCM Role	**Theory**	**Empirical**	**Practice in UA**	**Proposed Model Selection**
Change Manager	×	✓	100%	Yes
Change Control Board (CCB)	✓	✓	67%	Yes
Developer (version control)	×	✓	50%	Yes
Tester	×	✓	100%	Yes
Configuration Manager (CM)	✓	✓	100%	Yes

The human roles that involve in SCM implementation are change manager, developer, and tester but are not being highlighted in previous study and literature review [24]. This empirical study has revealed that current practices in UAs shown that the roles have significant involvement during the SCM process and implementation. Majority informants agree that the appointment of change control board has an advantage to reduce numbers of CRs. Appointment of configuration manager is essential and important to manage and monitor the SCM implementation in the organization. In addition to CCB and CM, there are other implementers needed to success in SCM implementation which are change manager, developer (version control) and tester are significant in SCM implementation based on informants of this study. For that reason, all five (5) roles of change manager, change control board, developer, tester, and configuration manager are selected to be included in the proposed model. This finding too is supported by another research done by Fahmy et al. [25].

4) Success Factors for SCM Implementation: In the interview sessions, informants were asked about factors that influent the success of SCM implementation. The study has revealed the success factors based on three identified components: SCM tool, process and human. Table IX lists the findings of success factors by component derived from the informant's feedbacks and based on practices in UAs involved in this study. Further elaboration can be found in [22], [26].

TABLE IX
SUCCESS FACTORS FOR SCM IMPLEMENTATION BY COMPONENT

Component	**Success Factors**	**Description**
SCM Tool	User training given	• Training on SCM tool
	Easy to use	• The SCM tool must be easy to be used
		• Usage of OTS SCM tools
	Easy integration with IDE	• SCM tools support that can make SCM application easier
Process	Clear user requirement	• Clear user requirement can reduce the CR
	Training for SCM process adaptation	• Training to adapt to SCM process and development
	Prepare documentation	• Documentation preparation
	Consultant support	• Appointed consultant to help in SCM application
Human	Understand SCM terms	• Full understanding related to SCM terms
	Understand job scope	• Dedicated configuration manager appointed
		• Dedicated head of SCM implementer
	SCM awareness	• SCM awareness
		• SCM practices
	Management commitment	• Clear management direction
		• Management commitment in supporting SCM
		• Appointed CCB
		• Management awareness
		• Strengthen the structure of the development and maintenance team
		• Infrastructure's commitment to support technology change
	Good communication and attitude	• Good internal communication
		• Trustworthy
		• High staff motivation
	Comply with management instructions	• Compliance with management instructions

In summary, the empirical study was conducted to identify current SCM practices in public universities in

Malaysia and the success factors for SCM implementation from practitioner's perspectives. Based on the analysis and findings have discovered that the success factors are influenced by several factors: human, process and tools. These factors have influential impact on the implementation since they are interrelated between components. Furthermore, the empirical study has identified four main activities of SCM that being practiced in UAs. The main processes are change management, version control, system building and release management.

III. Results and Discussion

The Simplified Software Configuration Management Model (or s-SCMM) is proposed and developed to enable the systematic and effective SCM implementation specifically in public university environment. SCM process was known and reported as a complex process [16], [17] and because of that many claims that practitioners did not used SCM effectively and inclusively [22], [26]. The s-SCMM is developed based on the findings of theoretical and empirical study as discussed in the previous sections. Fig. 1 illustrates the proposed s-SCMM.

A. Simplified Software Configuration Management Model (or s-SCMM) and Elements

Through the comparison between the theory and empirical findings, the important components and sub components for the development of s-SCMM are highlighted. As mentioned in previous section, the main elements of s-SCMM are process, SCM tool, and human. Once the elements are identified, the components of each element are added into the model. For process elements, each component is detailed out with sub components. Sub components, components, and elements are interrelated in this model to create a more accessible and practical SCM model for the UAs. In addition, the s-SCMM also includes the success factors that assist in supporting each element of the model.

As shown in Figure 1, it demonstrates the inter relationships between elements, components and sub components. The following sections explain the relationship between elements and the success factors embedded in the model.

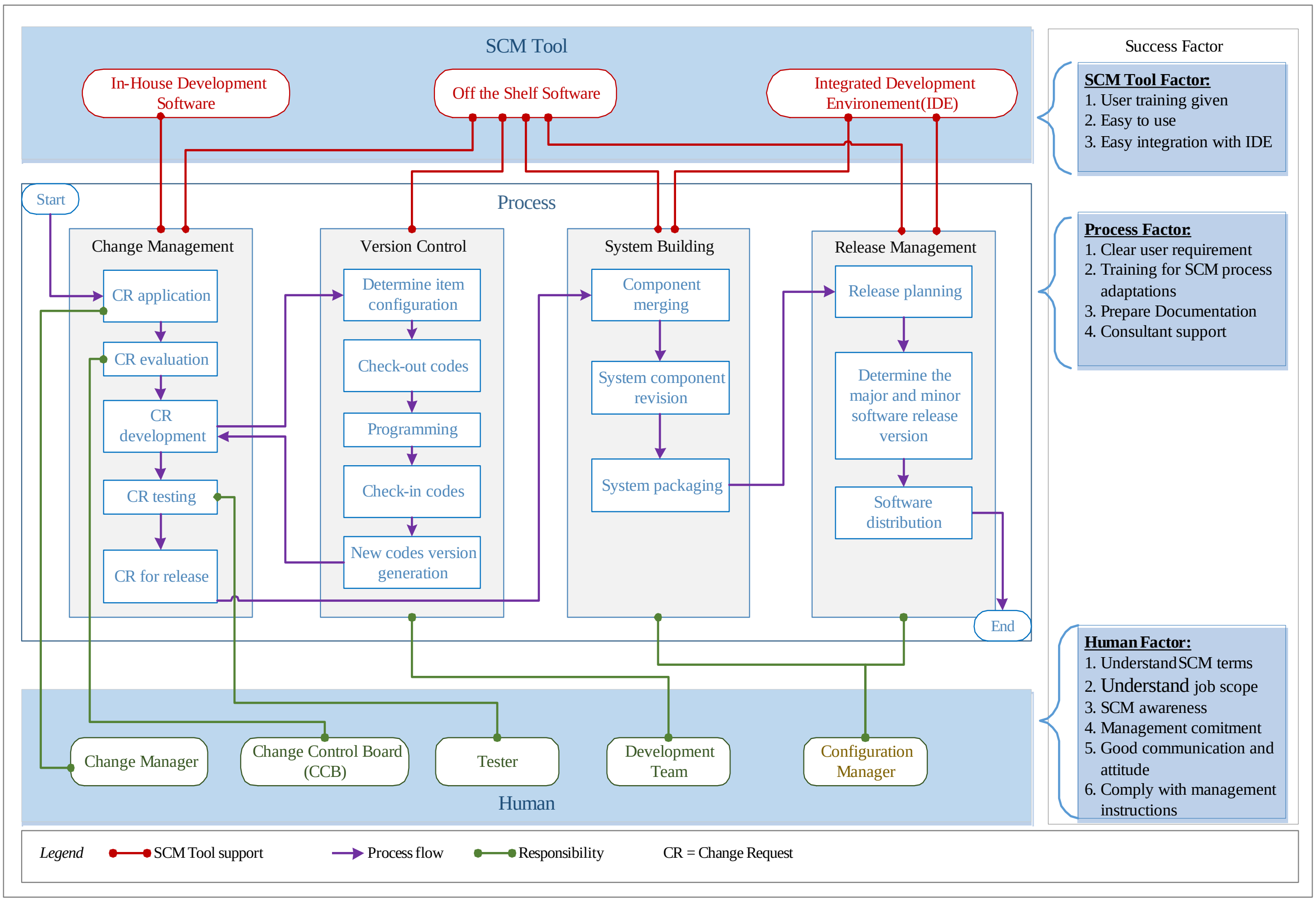

Fig. 1 Simplified Software Configuration Management Model (s-SCMM)

1) Process and Human Element: The Process element consists of four (4) components: change management, version control, system building, and release management. Table X lists the relationship and flow between input, process, output, and human that responsible for each sub component.

TABLE X
RELATIONSHIP BETWEEN INPUT, PROCESS, OUTPUT, AND HUMAN

Input	**Process**	**Output**	**Human**
Component: Change Management Process			
Start	CR application	CR evaluation	Change Manager
CR application	CR evaluation	CR development	CCB
CR evaluation	CR development	CR testing	Development Team
CR development	CR testing	CR for release	Tester
CR testing	CR for release	Determine item configuration	Configuration Manager
Component: Version Control Process			
CR for release	Determine item configuration	Checked out source code	Development Team
Determine item configuration	Checked out source code	Programming	Development Team
Checked out source code	Programming	Checked in source code	Development Team
Programming	Checked in source code	New codes version generation	Development Team
Checked in source code	New codes version generation	Component merging	Development Team
Component: System Building Process			
New codes version generation	Component merging	System component revision	Configuration Manager
Component merging	System component revision	System packaging	Configuration Manager
System component revision	System packaging	Release planning	Configuration Manager
Component: Release Management Process			
System packaging	Release planning	Determine the major and minor software release version	Configuration Manager
Release planning	Determine the major and minor software release version	Software distribution	Configuration Manager
Determine major/minor software release version	Software distribution	End	Configuration Manager

2) Process and SCM Tool Element: The components of SCM tool element are in-house development software, OTS software, and IDE. The relationship of process and SCM tool element are shown in the Fig. 2.

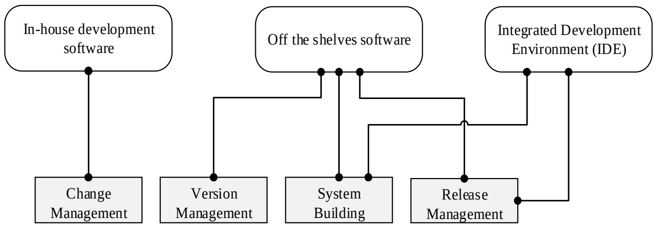

Fig. 2 Relationship between Process and SCM Tool Element

The relationship shows that only the in-house development software supports the change management process in the UAs. The off-the-shelf software supports version management, system building, and release management. On the other hand, the integrated development environment or IDE supports system building and release management.

3) Success *Factors Element:* The determination of elements, components, sub components, and relationship are finalized and inclusively added into the s-SCMM. After these items have been conceptualized, the success factors for SCM implementation, which were identified through the empirical study, are mapped into the model. As shown in Table IX, the success factors are categorized into three main elements: process, tool and human. The success factors are valuable for further understanding of criteria and requirement for successful SCM implementation of model by the organization.

B. Model Validation

The proposed s-SCMM was validated through expert validation. The validation process considers four aspects:

- Experts review on process, input, output and human (or executor)
- Review on the relationships between elements and components of s-SCMM
- Review on the success factors defined in the model
- Recommendation for improvement

Three experts from industry and academic were invited to join this research as the experts to validate the model. Based on the validation process, the experts gave average scores for the validation of 93%, 97% and 86% respectively and the average from the score is 92%. This shows that the s-SCMM is an appropriate model and can be applicable to public university as the simplified software configuration management model. Furthermore, the proposed model was refined and improved later based on feedback from the experts.

IV. CONCLUSION

This paper has presented the development of a simplified software configuration management or s-SCMM. The development is based on inputs from theoretical and empirical findings. The empirical study was conducted involving 12 experienced practitioners in public universities in Malaysia and has revealed three main elements of s-SCMM, which are process, tools and human. Each element is broken down into several components and sub components as discussed in this paper. The arrangement of elements and components in the model are to make better understanding on how SCM can be implemented in UA. The process elements embedded in the model are the fundamental and essential component for SCM process and are closely related to the development process that can aid the software practitioner to understand the model further. The relationship between elements, components and tools provide an overview on the SCM implementation requirements. Each element is mapped with implementation success factors, which were derived from empirical study, discussed in this paper. The proposed model was then validated by three experts and was being refined and improved. The s-SCMM model is developed that aims to offer a SCM model that is more practical and simplified for software practitioners in public university.

ACKNOWLEDGMENT

This research is funded by Malaysian Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UKM/02/1) and PP-FTSM-2019 Universiti Kebangsaan Malaysia.

REFERENCES

[1] Y. Jamaiah, D. Aziz, and H. Abdul Razak, “Software product certification model: Classification of quality attributes,” *First Reg. Conf. Comput. Sci. Technol. (RCCST 07), Kota Kinabalu*, pp. 436–440, 2007.

[2] J. H. Yahaya, A. Deraman, S. R. A. Ibrahim, and Y. Y. Jusoh, “Software Certification Modeling: From Technical to User Centric Approach,” *Aust. J. Basic Appl. Sci.*, vol. 7, no. 8, pp. 9–18, 2013.

[3] J. Yahaya, A. Deraman, A. R. Hamdan, Jamaiah H. Yahaya, Aziz Deraman, and Abdul Razak Hamdan, “JHY Continuously Ensuring Quality through Software Product Certification,” *2010 Int. Conf. Inf. Soc.*, pp. 183–188, 2010.

[4] R. Robbes and M. Lanza, “Of change and software,” Thèse de doctorat : Università della Svizzera Italiana, 2008.

[5] J. Estublier *et al.*, “Impact of Software Engineering on the Practice of Software Configuration Management,” *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 4, pp. 383–430, 2005.

[6] R. Conradi and B. Westfechtel, “Version models for software configuration management,” *ACM Comput. Surv.*, vol. 30, no. 20, pp. 232–282, 1998.

[7] A. Bartusevics and L. Novickis, “Models for Implementation of Software Configuration Management,” *Procedia Comput. Sci.*, vol. 43, pp. 3–10, 2015.

[8] “IEEE Draft Standard for Software and Systems Engineering--Software Testing--Part 2: Test Process,” *IEEE/ISO/IEC P29119-2-DIS, December 2011*, pp. 1–58, 2012.

[9] A. Leon, “Overview of SCM,” in *Software Configuration Management Handbook*, 3rd Ed., Norwood, MA: Artech House Publishers, 2015, pp. 1–14.

[10] J. Whyte, A. Stasis, and C. Lindkvist, “Managing Change in the Delivery of Complex Projects: Configuration Management, Asset Information and ‘Big Data,’” *Int. J. Proj. Manag.*, vol. 34, no. 2, pp. 339–351, 2016.

[11] N. P. Er and C. Erbaş, “Aligning Software Configuration Management with Governance Structures,” in *International Conference on Software Engineering*, 2010.

[12] A. Bartusevics, “The Development And Implementation Of Model – Driven Software Configuration,” Riga Technical University, 2015.

[13] S. Kucherov, J. Lipko, and O. Schevchenko, “The Integrated Life Cycle Model of Configurable Information System,” in *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT 2014)*, 2014, pp. 182–186.

[14] J. De Almeida Monte-Mor and A. M. . Da Cunha, “GALO: A semantic method for software configuration management,” *ITNG 2014 - Proc. 11th Int. Conf. Inf. Technol. New Gener.*, pp. 33–39, 2014.

[15] J. Wettinger *et al.*, “Integrating configuration management with model-driven cloud management based on TOSCA,” *CLOSER 2013 - Proc. 3rd Int. Conf. Cloud Comput. Serv. Sci.*, pp. 437–446, 2013.

[16] K. Rejström, “Implementing Continuous Integration in a Small Company: A Case Study,” Aalto University, 2016.

[17] R. Krikhaar and I. Crnkovic, “Software Configuration Management,” *Sci. Comput. Program.*, vol. 65, no. 3, pp. 215–221, 2007.

[18] M. E. Moreira, *Adapting Configuration Management for Agile Teams*, 1st Ed. United Kingdom: John Wiley & Sons, Inc., 2010.

[19] ISO, “ISO 10007:2003 - Quality management systems - Guidelines for configuration management, Second Edition,” 2003.

[20] IEEE, “IEEE Standard for Configuration Management in Systems and Software Engineering,” IEEE, NY, 2012.

[21] CMMI Product Team, “CMMI® for Development, Version 1.3,” Software Engineering Institute, MA, 2010.

[22] Z. Mohamad, J. Yahaya, A. Deraman, and Z. Mansor, “Issues and Challenges on Implementing Software Configuration Management in Public University,” *Adv. Sci. Lett.*, vol. 24, no. 7, pp. 5230–5233, 2018.

[23] Z. Zainol and E. . Hassan, “Amalan SCM di UKM.” 2016.

[24] Syahrul Fahmy, A. Deraman, and Jamaiah Yahya, “The Role of Human in Software Configuration Management,” in *ICSCA 2018 Proceedings of the 2018 7th International Conference on Software and Computer Applications*, 2018, pp. 56–60.

[25] Syahrul Fahmy, Aziz Deraman, Amir Ngah, and F. A. Salman, “A Model for People-Centric Software Configuration Management,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3, pp. 7–13, 2017.

[26] Aziz Deraman, Jamaiah Yahaya, Zulaikha Mohamad, and Syahrul Fahmy, “Technical Report 59297-3 - Software Configuration Management In Public Higher Education Institutions: An Observation,” Kuala Terengganu, 2016.