Annals of Emerging Technologies in Computing (AETiC)
Vol. 5, No. 5, 2021

Research Article

Human Competency Assessment for Software Configuration Management

Syahrul Fahmy[1,*], Aziz Deraman[2], Jamaiah Yahaya[3] and Abdul Razak Hamdan[3]

[1] University College TATI, Malaysia
fahmy@uctati.edu.my
[2] Universiti Malaysia Terengganu, Malaysia
a.d@umt.edu.my
[3] Universiti Kebangsaan Malaysia, Malaysia
jhy@ftsm.ukm.my; arh@ftsm.ukm.my
*Correspondence: fahmy@uctati.edu.my

Received: 17th October 2020; Accepted: 17th November 2020; Published: 20th March 2021

Abstract: Late product delivery is a common problem in software projects. Late delivery can be attributed to software change especially in the development and maintenance phases, leading to more work than originally planned, thus increasing project completion time. One approach for managing software change is *Software Configuration Management* (SCM). SCM was first introduced in the 1970s to ensure the timely delivery of software products. To date, SCM implementation is supported by recognized standards, best practice, and countless commercial and proprietary tools. However, after more than 50 years, the issue of late software product delivery still prevails, questioning the practice of SCM in software organizations. One aspect of SCM that has received little focus in mainstream research is the *human aspects*. This study aims to identify how the competency of SCM practitioners can be assessed through the identification of SCM competency criteria and the development of a competency assessment framework. The framework was validated through expert reviews and case studies involving practitioners from the public sector, industry, institutes of higher learning and international organizations. Results confirmed the reliability of the SCM competency criteria and the plausibility of the competency assessment framework.

Keywords: ***Software Configuration Management; Software quality; Software engineering; Human competency***

1. Introduction

Late product delivery is a common problem in software projects. Late delivery is caused by, amongst other things, software changes, leading to more development work than originally planned, thus increasing project completion time [1]. *Software Configuration Management* (SCM) has been used in software engineering to ensure timely completion of software projects and delivery of software products since the 1970s. SCM is a systematic change management process and can be defined as a "*supporting process in software engineering that controls changes to software products*" [2]. However, after more than half a century, the problem of late software product delivery and projects still prevail in software engineering [3].

One significant factor that influences the success of software projects is the *human factor* [4]. Although human factors have been studied actively [5-6], the focus are mainly on technological or process-related factors [6]. This is evident in SCM where studies involving human factors have focused on the process, for example collaboration and debugging [7], rather than the success of SCM implementation. Software development is a human activity [8], and human factors play a critical role in its success [9]. Competency is one of the many aspects of human factors and the lack of human competency has been identified as on of the issues in software development [10].

Syahrul Fahmy, Aziz Deraman, Jamaiah Yahaya and Abdul Razak Hamdan, "Human Competency Assessment for Software Configuration Management", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 69-78, Vol. 5, No. 5, 20th March 2020, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2021.05.008, Available: http://aetic.theiaer.org/archive/v5/v5n5/p8.html.

This study aims to identify how the competency of SCM practitioners can be assessed. There are two challenges in meeting this aim. The first challenge is to identify a set of SCM competency criteria as there is yet a standard set of competency criteria for software practitioners. The second challenge is to formulate a competency assessment framework for SCM. Major competency frameworks focus more on other software engineering areas and not explicitly to SCM. Both challenges are addressed in this paper.

2. Software Configuration Management

Software Configuration Management (SCM) is commonly used to manage the evolution of software systems. It is defined as "*a supporting-software life cycle process that benefits project management, development and maintenance activities*" [2]. SCM can be traced back to 1950s when poorly documented engineering changes in the aerospace industry posed a problem to spacecraft production. When *Software Managers* faced the problem of managing software change in the 1970s, SCM was adapted in the software development process [11].

SCM research and development (R&D) activities has been highly focused on addressing the problems at hand in software development for example programming in the large in the 1980s [12]; development of object-oriented systems in the 1990s [13]; constructions of web services in the 2000s [14]; and configurations of late binding systems post 2010 [15]. The corpus of research has been in the development of tools for software engineering. This has translated into wealth of SCM-related tools, making SCM implementation highly dependent on these tools. To date, SCM tools are pervasive and available to other areas such as web services and programming environments.

Although the role of human in software development has been acknowledged as early as the 1970s [16], R&D activities related to human factor in SCM have focused on the *process* for example collaboration [17]; education [18]; and debugging [19], rather than the *success* of SCM implementation. One aspect of human factor is *competency*. Competency is a set of characteristics that improve the efficiency and performance of a practitioner. The lack of (human) competency has been identified as one of the challenges in software development [10]. The quality of software products is highly dependent on the knowledge, abilities, and talent of the project team [20]. Traditionally, the competency of software practitioners was built on personal qualities and developed through experience. To date, formal education and professional training have become more significant in the development of competency.

Competency has many facets including knowledge, skills, abilities and personal attributes. It can be divided into *specific* [21] and *generic* [22] competencies. Unfortunately, there is no one size fits all or silver bullet for determining the competencies for software practitioners as organizational functions require a combination of competencies in order to be performed effectively [23]. In addition, there is no widely accepted competency framework for systems engineering. Major competency frameworks including PCMM [24]; INCOSE [25], MITRE [26], and SwA [27] are not suitable for this study as they focus more on other software engineering areas than SCM.

PCMM adapts the maturity framework of the *Capability Maturity Model* which aims to improve software development processes. As the scope of this study is SCM based on IEEE classification, PCMM is not suitable for assessing the competency of practitioners as it falls under the areas of *Software Process* and *Software Construction*. INCOSE presents description and indicators of knowledge and experience for competency in four levels namely *Awareness*, *Supervised Practitioner*, *Practitioner*, and *Expert*. However, there is no specific mention of SCM in the framework although the competency for *Planning*, *Monitoring* and *Controlling* is listed in the *System Engineering Management* category. This reduces the suitability of this framework for this study. In MITRE, SCM is discussed in the *Systems Engineering Planning* and *Management* section although no specific mention for tools utilization is made in the competency levels. The only mention of tools appear in the *Intermediate Behaviour* with the "*recommendation of configuration management and deficiency reporting tools*". As SCM implementation is highly dependent on tools, the lack of competency for tools utilization reduces the suitability of MITRE for this study. SwA falls under the *Software Quality* area in the IEEE classification of software engineering key areas and not SCM, thus reducing its suitability for this study. Although the *Software Engineering Competency Model* (SWECOM) [28] explicitly focuses on SCM, tasks remain in identifying suitable competency level for each skill set, and mapping specific skill set to a particular SCM process.

3. Identification of Competency Criteria

SCM competency criteria was identified in two surveys.

3.1. Questionnaire

The first survey identified SCM practice in software organizations throughout Malaysia. Software practitioners from the public sector, industry and *Institutes of Higher Learning* (IHLs) were selected as respondents. Questionnaire was selected as the survey tool as respondents are geographically scattered. The questionnaire was based on the standard SCM process [2] and the content was validated by a group of experts representing the target respondents. Construct validity was not carried out as the questionnaire was based on a standard document [29-30].

The questionnaire had 40 questions in 6 sections, and a total of 19 practitioners took part in the survey; 3 from the public sector, 11 from the industry, and 5 from IHLs. Results of the survey revealed 4 main SCM components namely *People*, *Process*, *Tools*, *Documentation* and their relationships as viewed by practitioners. Successful SCM implementation is dependent on the competency *People* to implement *Process*, operate *Tools* and generate *Documentation*. SCM implementation involves 6 inter-related process (*Planning*, *Identification*, *Control*, *Accounting*, *Auditing*, *Delivery*). The use of SCM tools are prevalent in software projects, as well as general tools for project reporting, progress assessment and project monitoring purposes. In addition, 3 types of SCM tools that are commonly used are *software versioning*, *software building*, and *software release* tools. The types of documents generated in software projects are in line with international standards. Minor variations are noted due to organizational culture and constraints.

3.2. Interview Sessions

Based on the questionnaire results, it is discovered that IHLs have the highest number of late project delivery, requiring an additional of 2.6 months for completion (on average). As such, the second survey specifically targeted IHLs to identify key factors that inhibit and promote SCM implementation. Open-ended [31] and unstructured [32] interview questions were used in order to get as much insights to significant areas in SCM. A total of 12 respondents from 5 IHLs participated in this survey in 9 interview sessions comprising of individual and group sessions, carried out at the respondents' organization. Results of the survey were summarized into SCM implementation issues and success factors.

People-related issues reported include conformance to directives, ineffective communication, encumbering key performance indicators, management's commitment towards timely completion of software projects, procedures for change request, bureaucracy, and additional task assignments. *Process*-related issues reported include ineffective communication, ambiguous requirements, inefficient change request procedure, manual change request process, frequent restructuring exercise, lack of infrastructure support for implementation, and lack of dedicated SCM manager to oversee implementation. Issues related to *Tools* include the adoption of technological change into the work process, tools suitability, tools utilization, diversity of target platforms for testing, implementation, and maintenance purposes, and vendor support. Issues related to *Documentation* include ineffective communication, poor documentation, lack of a dedicated SCM team, lack of understanding of SCM process and outcomes, lack of management's commitment, and vague organization directions.

Major contributing factors include a dedicated SCM manager to oversee implementation, efficient communication between all stakeholders involved, conformance to directives, training, clear organization direction, and management's commitment towards SCM process and implementation. Other contributing factors reported were the foundation of a *Change Control Board*, appropriate tools to support implementation, the use of SCM external consultant, the need for a suitable infrastructure to be in place, availability of SCM tools, SCM awareness, dedicated development and maintenance teams, clear requirements to minimize the number of change requests, and high morale of staff.

3.3. Competency Formulation

Filtering the survey results and leaving out non SCM factors such as budget and team size; and combining similar factors such as tools suitability and utilization; 10 significant factors to SCM were derived, illustrated in Table 1.

Table 1. Competency in SCM

SCM Success Factors	**Competency**				
	Qualification	Experience	Professionalism	Training	SCM Skills
Awareness	✓	✓	✓		
Change Request Procedure	✓	✓	✓		✓
Communication		✓	✓		
Competency	✓	✓	✓	✓	✓
Documentation	✓	✓	✓		✓
Infrastructure Support	✓	✓		✓	✓
Management's Commitment	✓		✓		
Requirements	✓	✓	✓	✓	
Technological Change	✓	✓	✓	✓	
Tools	✓	✓	✓	✓	✓

These factors are represented by a set of 5 competency criteria namely *Knowledge*, *Experience*, *Professionalism*, *Training*, and *SCM Skills*. These criteria are key for addressing SCM issues and promoting successful implementation.

The *Malaysian Qualifications Framework* [33] and the *Malaysian Program Standards for Computing* [34] were referred to in defining and refining the competency for Knowledge. Knowledge refers to the level of formal education in Computing and represented by education level and expertise in software engineering key areas. Education level refers to qualification obtained; area of expertise refers to the field of qualification; and key areas refer to the 13 software engineering key areas. SWECOM and SWEBOK [2] were referred to in defining the competency for Experience. Experience refers to the mastery of key areas through involvement in software development projects. The *Software Engineering Code of Ethics and Professional Practice* [35] was referred to in defining the competency for Professionalism. It refers to the professional conduct of a practitioner and has been adopted by the *IEEE Computer Society* and ACM. Training refers to the completion of workshops, seminars, and other learning mediums in software engineering key areas. SWECOM was referred to in defining the competency for Skills. It refers to the ability to implement SCM namely *Planning*, *Conducting* and *Managing*.

4. Competency Assessment Framework

SCM competency assessment evaluates the competency of practitioners in implementing SCM process, utilizing tools and generating documentation. Assessment also illustrates the organization's project management practice and highlights areas of concern. The assessment framework is illustrated in Figure 1.

Competency assessment can be carried out to all type software projects (new, current, completed, outsourced). Assessment for new software project would facilitate planning activities such as tools procurements and training needs. Assessment for current project is prompted by discrepancies in the project's schedule, amongst other things. As such, assessment is carried out to gain insights as to why the discrepancies occur and highlight areas for improvements. Assessment for completed projects is mainly a review exercise to determine what worked and what did not with regards to *Human*, *Process*, *Documentation* and *Tools*. Relevant data from past projects are collected and the results are used to plan future projects more effectively. Assessment for outsourced projects provides alternative input in the vendor selection process. Additional data can be included in the assessment, tailored to the requirements and needs of the organization. Competency assessment can be carried out through self-estimation surveys, interviews or work audit. In this study, a self-estimation questionnaire was developed and administered.

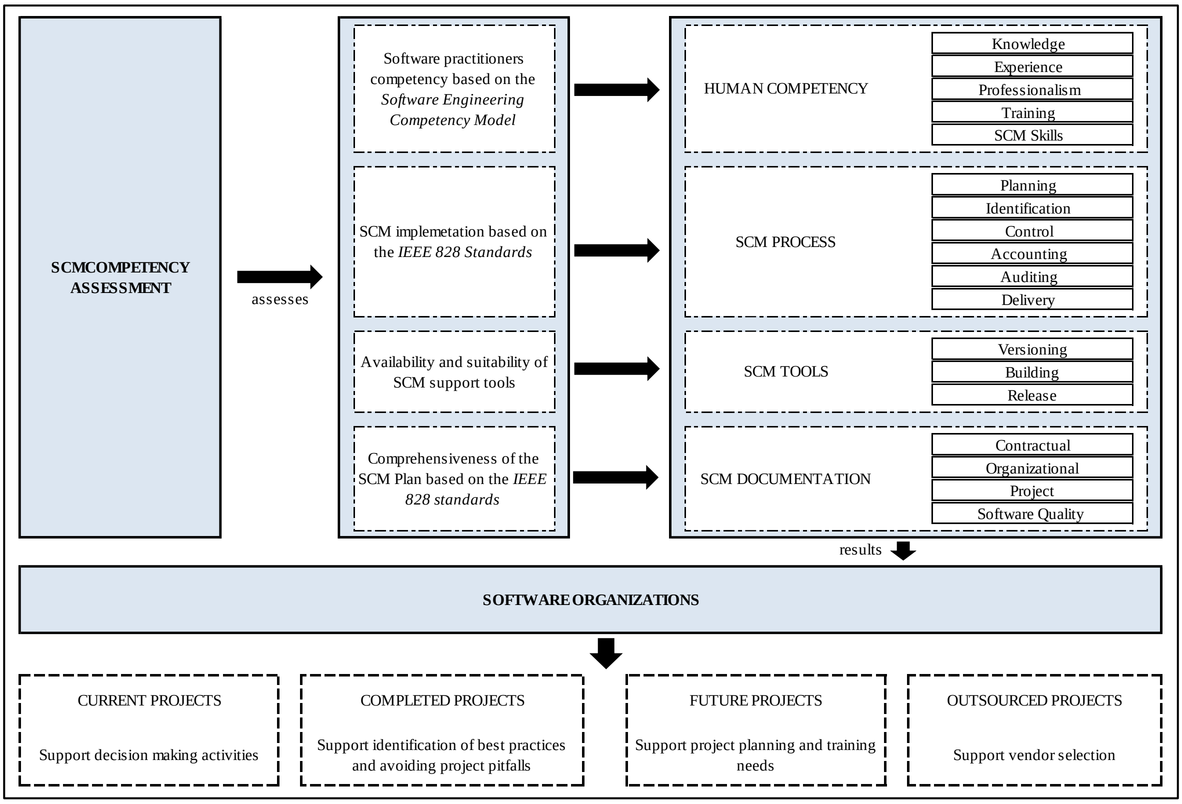
SCMCOMPETENCY ASSESSMENT
assesses
Software practitioners competency based on the *Software Engineering Competency Model*
SCM implemetation based on the *IEEE 828 Standards*
Availability and suitability of SCM support tools
Comprehensiveness of the SCM Plan based on the *IEEE 828 standards*
HUMAN COMPETENCY
Knowledge
Experience
Professionalism
Training
SCM Skills
SCM PROCESS
Planning
Identification
Control
Accounting
Auditing
Delivery
SCM TOOLS
Versioning
Building
Release
SCM DOCUMENTATION
Contractual
Organizational
Project
Software Quality
results
SOFTWARE ORGANIZATIONS
CURRENT PROJECTS
Support decision making activities
COMPLETED PROJECTS
Support identification of best practices and avoiding project pitfalls
FUTURE PROJECTS
Support project planning and training needs
OUTSOURCED PROJECTS
Support vendor selection

Figure 1. Competency Assessment Framework for Software Configuration Management

4.1. Human

Competency is determined based on academic qualification, professional experience, professional conduct, training, and SCM skills. Five levels of competency are adopted based on SWECOM and Dreyfus model of acquisition [36] namely *Software Engineer*, *Technical Leader*, *Practitioner*, *Entry-Level Practitioner*, and *Technician*. Determination of competency is as follows:

- Software Engineer holds at least a first degree in Computing, has the proficiency of an Expert, with at least 5 years of Experience.
- Technical Leader holds at least a first degree in Computing, has the proficiency of Proficient, with at least 5 years of Experience.
- Practitioner holds at least and a first degree in Computing, has the proficiency of Competent, with at least 5 years of Experience.
- Entry Level Practitioner holds at least a first degree in Computing, has the proficiency of a Beginner, with less than 5 years of Experience.
- Technician holds at least a diploma in Computing, has the proficiency of a Novice, with any years of Experience.

4.2. Process

Competency is determined based on the extensiveness of SCM process implementation by the organization. There are 6 SCM-related process based on the IEEE-828 standard (*Planning*, *Identification*, *Control*, *Accounting*, *Auditing*, *Delivery*) [37]. Determination of competency is as follows:

- Minimum competency for managing the Planning process is a Practitioner, as a practitioner *"participates in determining impact of constraints imposed by policies, contract, and SDLC"*.

- Minimum competency for managing the Identification process is a Practitioner, as a practitioner *"participates in identifying CIs and the relationships among them" and "participates in developing software release plans"*.
- Minimum competency for managing the Control process is a Technical Leader, as a technical leader *"appoints members and convenes the Change Control Board"* and *"tailors and adopts mechanisms for requesting, evaluating, and approving changes"*.
- Minimum competency for managing the Accounting process is a Technical Leader, as a technical leader *"leads the Change Control Board in making decisions on change requests"* and *"ensures that approved changes are made and documented"*.
- Minimum competency for managing the Audit process is a Technical Leader, as a technical leader *"establishes and maintains the mechanisms for generating audit reports"*.
- Minimum competency for carrying out the Delivery process is a Practitioner, as a practitioner *"leads the building and verifying of software releases"*.

4.3. Tools

Competency is determined based on the tools used in the SCM project. Three level of competency is adopted based on SWECOM namely the *specification, acquisition* and *utilization* of tools. Determination of competency is as follows:

- Minimum competency for specifying tools is a Technical Leader, as a technical leader *"specifies SCM tools"* and *"establishes and maintains the mechanisms for generating audit reports"*.
- Minimum competency for acquiring SCM tools is a Practitioner, as a practitioner *"procures SCM tools"* and *"develops and configures tools for generating audit reports"*.
- Minimum competency for utilizing SCM tools is a Technician, as a technician *"operates SCM tools"*, *"operates tools to generate status and audit reports"*, and *"operates tools to build software releases"*.

4.4. Documentation

Competency is determined based on the documents generated in SCM. Two levels of competency are adopted based on SWECOM namely *generating* and *updating* documents. Determination of competency is as follows:

- Minimum competency for generating SCM documents is a Practitioner, as a practitioner *"develops and maintains the SCMP"*, *"uses established procedures for populating and maintaining the library"*, and *"uses established mechanism to record and report information"*.
- Minimum competency for updating SCM documents is an Entry Level Practitioner, as an entry level practitioner *"assists in developing, updating and maintaining the SCM Plan"*, *"implement and document approved changes to CIs"*, and *"provides audit reports as scheduled and requested"*.

To complement the assessment results and facilitate software organizations in taking corrective actions, several areas of concern can be derived such as training and professionalism. These informations would serve as a starting point to identify why a certain process, tool, or document is not implemented, utilized or generated although the minimum competency is met.

5. Validation

The proposed SCM competency assessment framework was validated through expert reviews. The identification and selection of experts were based on the industry. A total of 11 experts was selected in this study: 2 from the public sector, 3 from the industry, 4 from IHLs and 2 international experts. List of experts is as follows:

- Expert 1 is a Principal Assistant Director at the Information Technology Section, Attorney General's Chambers; holds a postgraduate degree in Information Systems, a degree in Information Technology; has 17 years of professional experience; and has 2 professional certificates in Software Engineering.

- Expert 2 is a Senior IT Officer at the Information Management Division, Ministry of Tourism and Culture; holds a degree in Computer Science; and has 12 years of professional experience.
- Expert 3 is a Senior Assistant Director at the Centre for Information Technology Development & Services, Universiti Malaysia Sarawak; holds a postgraduate degree in Information Technology, a degree in Information Systems, a diploma in Information Technology; has 19 years of professional experience; and has 3 professional certificates in Information Technology.
- Expert 4 is a Senior Information Technology Officer at the Centre for Information & Communication Technology, Universiti Malaysia Perlis; holds a degree in Information Technology; has 14 years of professional experience; and has 3 professional certificates in Information Systems.
- Expert 5 is a Senior IT Officer at the Information Technology Centre, Universiti Tun Hussein Onn Malaysia; holds a degree in Computer Science; and has 15 years of professional experience.
- Expert 6 is the Head of IT Service Centre at University College TATI; holds a postgraduate degree in Computer Science, a degree in Information Technology and a diploma in Information Technology; and has 15 years of professional experience.
- Expert 7 is a Business Strategy Manager at Telekom Malaysia Bhd; holds a degree in Information Systems; and has 18 years of professional experience. The nature of business includes communication & networking, data centre & web hosting, mobile & wireless and software development & system integration.
- Expert 8 is an Account Manager at Epson Malaysia Sdn. Bhd.; holds a diploma in Computer Science; and has more than 10 years of professional experience; The nature of business includes communication & networking, consultancy & professional services, creative design & content and distributor & retailer.
- Expert 9 is a Manager at Top IT Industries Sdn. Bhd.; holds a diploma in Computer Science; and has more than 10 years of professional experience. The nature of business includes communication & networking, creative design & content, maintenance, and software development & system integration.
- Expert 10 is the CEO of Rex Black Consulting Services, Texas, USA; has over 30 years of software and system engineering experience; and Chair of the Agile Working Group and past president of the International Software Testing Qualifications Board.
- Expert 11 is a System Software Engineer and Tester Consultant at the Grand Software Testing, Denver, USA; has over 35 years of professional experience in software testing; lead editor and author of international standards including IEEE and ISO.

Semi-structured interviews sessions followed by a short questionnaire were administered to the experts, covering SCM components, competency criteria, and benefits of competency assessment to organizations. The interview questions were developed based on a brainstorming session with Computing post-graduate students at *Universiti Malaysia Terengganu*. The questions were then drafted into questionnaire and sent to a group of 5 computing lecturers (experts) at University College TATI. The final set of questions had 16 items, covering SCM components, competency criteria, and benefits of competency assessment to organizations. A 5-point Likert scale was used to record the responses. Validation revealed the soundness of the SCM components (*Human*, *Process*, *Tools*, *Documentation*); the competency criteria (*Knowledge*, *Experience*, *Proficiency*, *Training*, *Professionalism*); and the feasibility of competency assessment in different software development environments (*new*, *existing*, *completed*, *outsourced*).

6. Conclusion

This study started-off with the aim to identify how the competency of SCM practitioners can be assessed, leading to the identification of SCM competency criteria and the development of an SCM competency assessment framework. This study has successfully identified a set of SCM-specific competency criteria based on two surveys carried out and mapped against standard documents including the *Malaysian Qualifications Framework*, the *Program Standards for Computing* and the ACM/ IEEE *Software Engineering Code of Ethics and Professional Practice*. The selection of competency criteria was validated by experts from the public sector, industry, institutes of higher learning and international organizations.

A comprehensive SCM competency assessment framework has also been developed to determine the competency of practitioners, the extensiveness of SCM process implementation, the utilization of support tools, and the comprehensiveness of the SCM plan. The framework is based on international standards including SWECOM and SWEBOK. The framework was validated by experts from the public sector, industry, institutes of higher learning and international organizations.

This work has shed light to the roles played by human in SCM, complementing similar works such as the effect of knowledge to software quality [7-8], and contributed to the areas of SCM model and framework, where similar works include the impact of change to software quality and development [1] and component-based SCM model [38].

Acknowledgement

The Ministry of Education, Universiti Malaysia Terengganu (FRGS-59561) and University College TATI.

References

[1] Irina Brudaru and Andreas Zeller. (2008). What is the Long-Term Impact of Changes? In Proceedings of the 2008 international workshop on Recommendation systems for software engineering. Association for Computing Machinery, New York, NY, USA, pp. 30–32. DOI:10.1145/1454247.1454257.

[2] Pierre Bourque and Richard Fairley. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0 (3rd ed.). IEEE Computer Society Press, Washington, DC, USA. ISBN:9780769551661.

[3] Pulse of the Profession 2015: Capturing the Value of Project Management. Available http://www.pmi.org/learning/pulse.aspx.

[4] Chintan Amrit, Maya Daneva, Daniela Damian. (2014). Human Factors in Software Development: On its Underlying Theories and the Value of Learning From Related Disciplines. A guest editorial introduction to the special issue, Information and Software Technology, 56(12): 1537-1542. DOI:10.1016/j.infsof.2014.07.006.

[5] Jingdong Jia, Pengnan Zhang and Luiz Fernando Capretz. (2016). Environmental Factors Influencing Individual Decision-Making Behavior in Software Projects: A Systematic Literature Review. In Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering. Association for Computing Machinery, New York, NY, USA, pp. 86–92. DOI:10.1145/2897586.2897589.

[6] Rafael Prikladnicki, Marcelo Perin and Sabrina Marczak. (2016). Virtual Team Configurations that Promote Better Product Quality. In Proceedings of the 10th ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement. Association for Computing Machinery, New York, NY, USA, Article 18, pp. 1–5. DOI:10.1145/2961111.2962638.

[7] Kevin North, Shane Bolan, Anita Sarma and Myra Cohen. (2015). Gitsonifier: Using Sound to Portray Developer Conflict History. In Proceedings of the 10th Joint Meeting on Foundations of Software Engineering. Association for Computing Machinery, New York, NY, USA, pp. 886–889. DOI:10.1145/2786805.2803199.

[8] Luiz Fernando Capretz. (2014). Bringing the Human Factor to Software Engineering. IEEE Software, 31(02): 104-104. DOI:10.1109/MS.2014.30.

[9] Cintan Amrit. (2008). Improving Coordination in Software Development Through Social and Technical Network Analysis. Enschede: University of Twente. DOI:10.3990/1.9789036527767.

[10] Phillip Holtkamp, Ivan Lau and Jan Pawlowski. (2014), How Software Development Competences Change in Global Settings - An Explorative Study. Journal of Software Evolution and Process, 27: 50–72. DOI:10.1002/smr.1701.

[11] Jacky Estublier, David Leblang, Andre van der Hoek, Reidar Conradi, Geoffrey Clemm, Walter Tichy and Darcy Wiborg-Weber. (2005). Impact of Software Engineering Research on the Practice of Software Configuration Management. ACM Transactions on Software Engineering and Methodology 14(4): 383–430. DOI:10.1145/1101815.1101817.

[12] William Bryan, Stanley Siegel and Gary Whiteleather. (1981). An Approach to Software Configuration Control. SIGMETRICS Performance Evaluation Review 10(1): 33–47. DOI:10.1145/1010627.807907.

[13] Hal Render and Roy Campbell. (1991). An Object-Oriented Model of Software Configuration Management. In Proceedings of the 3rd international workshop on Software configuration management. Association for Computing Machinery, New York, NY, USA, pp. 127–139. DOI:10.1145/111062.111079.

[14] Daniel Junqueira, Thiago Bittar and Renata Fortes. (2008). A Fine-Grained and Flexible Version Control for Software Artifacts. In Proceedings of the 26th annual ACM international conference on Design of communication. Association for Computing Machinery, New York, NY, USA, pp. 185–192. DOI:10.1145/1456536.1456576.

[15] Santiago Perez De Rosso and Daniel Jackson. (2013). What's Wrong with git? A Conceptual Design Analysis. In Proceedings of the 2013 ACM international symposium on New ideas, new paradigms and reflections on programming & software. Association for Computing Machinery, New York, NY, USA, pp. 37–52. DOI:10.1145/2509578.2509584.

[16] Gerald Weinberg. (1985). The Psychology of Computer Programming. John Wiley & Sons, Inc., USA. ISBN: 9780442292645.

[17] Anita Sarma, Josh Branchaud, Matthew Dwyer, Suzette Person and Neha Rungta. (2014). Development Context Driven Change Awareness and Analysis Framework. In Companion Proceedings of the 36th International Conference on Software Engineering. Association for Computing Machinery, New York, NY, USA, pp. 404–407. DOI:10.1145/2591062.2591100.

[18] Pekka Makiaho, Timo Poranen and Ari Seppi. (2014). Version Control Usage in Students' Software Development Projects. In Proceedings of the 15th International Conference on Computer Systems and Technologies. Association for Computing Machinery, New York, NY, USA, pp. 452–459. DOI:10.1145/2659532.2659646.

[19] Gerald Bortis and Andre van der Hoek. (2013). PorchLight: A Tag-Based Approach to Bug Triaging. In the 35th International Conference on Software Engineering, San Francisco, CA, pp. 342-351. DOI:10.1109/ICSE.2013.6606580.

[20] Jose Rivera-Ibarra, Josefina Rodríguez-Jacobo, Jose Fernández-Zepeda and Miguel Serrano-Vargas. (2010). Competency Framework for Software Engineers. In the 23rd IEEE Conference on Software Engineering Education and Training, Pittsburgh, PA, pp. 33-40. DOI:10.1109/CSEET.2010.21.

[21] Franz Weinert. (2001). Concept of Competence: A Conceptual Clarification. In Dominique Rychen & Laura Salganik (Eds.), Defining and selecting key competencies (p. 45–65). Hogrefe & Huber Publishers.

[22] John Thompson, Roger Stuart and Phillip Lindsay. (1996). The Competence of Top Team Members: A Framework for Successful Performance. Journal of Managerial Psychology, 11(3): 48-66. DOI:10.1108/02683949610113593.

[23] Steven Abraham, Lanny Karns, Kenneth Shaw and Manuer Mena. (2001). Managerial Competencies and the Managerial Performance Appraisal Process. Journal of Management Development, 20(10): 842-852. DOI:10.1108/02621710110410842.

[24] Bill Hefley, Bill Curtis and Sally Miller. (2009). The People CMM: A Framework for Human Capital Management (2nd edition). ISBN: 9780321553904.

[25] INCOSE. (2010). Systems Engineering Competencies Framework 2010-0205. San Diego, CA, USA: International Council on Systems Engineering. Available https://www.sebokwiki.org/wiki/Systems_Engineering_Competencies_Framework_2010-0205.

[26] MITRE Systems Engineering Competency Model. (2007). The MITRE Corporation. Available https://www.mitre.org/sites/default/files/publications/10_0678_presentation.pdf.

[27] Thomas Hilburn, Mark Ardis, Glenn Johnson, Andrew Kornecki and Nancy Mead. (2013). Software Assurance Competency Model. CMU/SEI-2013-TN-004. Software Engineering Institute, Carnegie Mellon University. Available http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=47953.

[28] Software Engineering Competency Model. (2014). Available https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-competency-model.

[29] Marie-Claude Boudreau, David Gefen and Detmar W. Straub. (2001). Validation in Information Systems Research: A State-of-the-Art Assessment. MIS Quarterly, 25(1): 1–16. DOI:10.2307/3250956.

[30] Detmar W. Straub. (1989). Validating Instruments in MIS Research. MIS Quarterly, 13(2): 147-169. DOI:10.2307/248922.

[31] John Creswell. (2009). Research Design: Qualitative, Quantitative and Mixed Methods Approaches. Thousand Oaks, CA: Sage Publications. ISBN:9781412965576.

[32] Judith Bell and Stephen Waters. 2014. Doing Your Research Project. McGraw-Hill Education. ISBN:9780335264469.

[33] Malaysian Qualification Framework. (2011). Available https://www.mqa.gov.my/pv4/mqf.cfm

[34] Malaysian Qualifications Agency. (2014). Programme Standards: Computing. Available http://www2.mqa.gov.my/qad/garispanduan/Computing.pdf.

[35] Don Gotterbarn, Keith Miller and Simon Rogerson. (1997). Software Engineering Code of Ethics. Communications of the ACM 40(11): 110-118. DOI:10.1145/265684.265699.

[36] Stuart Dreyfus and Hubert Dreyfus. (1980). A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition. University of California, Berkeley. Available https://apps.dtic.mil/dtic/tr/fulltext/u2/a084551.pdf.

[37] IEEE Standard for Configuration Management in Systems and Software Engineering. (2012). In IEEE Std 828-2012 (Revision of IEEE Std 828-2005), pp.1-71. DOI:10.1109/IEEESTD.2012.6170935.

[38] Hong Mei, Lu Zhang and Fuqing Yang. (2001). A Software Configuration Management Model for Supporting Component-Based Software Development. SIGSOFT Software Engineering Notes 26(2): 53–58. DOI:10.1145/505776.505790.