

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Assyne, Nana

Title: Hard Competencies Satisfaction Levels for Software Engineers : A Unified Framework

Year: 2019

Version: Accepted version (Final draft)

Copyright: © 2019 Springer Nature Switzerland AG

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Assyne, N. (2019). Hard Competencies Satisfaction Levels for Software Engineers : A Unified Framework. In S. Hyrynsalmi, M. Suoranta, A. Nguyen-Duc, P. Tyrväinen, & P. Abrahamsson (Eds.), ICSOB 2019 : 10th International Conference of Software Business (pp. 345-350). Springer. Lecture Notes in Business Information Processing, 370. https://doi.org/10.1007/978-3-030-33742-1_27

Hard Competencies Satisfaction Levels for Software Engineers: A Unified framework

Nana Assyne

Faculty of Information Technology, University of Jyväskylä, Finland
nana.m.a.assyne@student.jyu.fi

Abstract. Software engineer's/developer's competency has long been established as a key pillar for the development of software. Nevertheless, the satisfaction levels derived from using a competency needs more investigation. The aim of this paper is to propose a framework for identifying hard competencies and their satisfaction levels. The paper contributes to the software engineering competency research by highlighting the satisfaction levels of hard competence for the benefit of the educators (academia), software engineers and users of software competence (practitioner).

Keywords: Hard competency, Technical competency, Software engineers' competencies, Competence satisfaction levels.

1 Introduction

Software are the principal driving force for hardware that currently run our daily lives. As such software development is propelled by the competency of the software developers. Competency is said to be the combination of abilities, knowledge, and skills for performing an assigned task. Competency then includes both soft and hard competencies [1]: a hard skill is or are the skill(s) one needs to be able to perform a job or assignment. Hard skills are teachable and acquired mostly through formal training and studies, and are sometimes referred to as technical skill. Often for example a trainee is required to be smart or must possess a good IQ to acquire the required skill. Thus, hard/technical skills are pre-requisite skills required by software engineers/developer in software development process.

Whereas both practical and empirical knowledge on technical competencies of software developers is not lacking, competency study has become an important and fundamental strategic area for academic research. Colomo-palacios et al. identify the competency levels relevant to software engineering of professional profiles [2]. Turley and Bieman in an attempt to identify non-exceptional and exceptional competencies of software engineers, also provided the technical competencies of software engineers [3]. Yet – there is paucity of studies that examines the satisfaction levels derived for possessing or using a competence.

Though the works of [4]; [2] and [5] establish the essence of hard or technical competence to software development, if we do not know the satisfaction level derived

as assurance for the possessor or the user, beneficiary cannot know which competency will be demanded or be needed. Our initial study looked at [2] work, which examined relevant levels of profile of software engineers and professional. Also the work of [6] assesses base competencies necessary for software engineering students. We do agree with the said work and argue further that it gives credence to the software engineering competency. However, we are of the view that additional satisfaction levels of the competency will provide assurance for both possessor and users in the software engineering community. Thus, there is a need to provide strategic frameworks for the various satisfaction levels of hard or technical competencies of software developers. This paper forms part of broader research on software developer's competency study.

The goal of this paper is to use existing models to create classification levels for the benefit of the users and possessors of software engineering competency. We therefore set our research question as: how do we determine the benefit or satisfaction of a competency of technical or hard competencies for software developers, thus, the research question for this paper is:

What are the different satisfaction levels derived from using a software technical or hard competency?

Research on software competency is not necessarily lacking in software engineering studies [7], however, in this study the Kano model, which is the main framework for this study is being used for the first time on competencies as against its original use on products. To structure this study to fit into previous studies for practical use, we also made use of Competency Framework for Software Engineers (CFSE) [8]. The framework has two main areas, that is soft and hard competency. Since this paper focuses on hard competency, we make use of that as part of the framework. This paper is structured as follows: section 2 discusses the theoretical foundations, section 3, methodology and the proposed framework, section 4, conclusions and future work.

2 Theoretical foundations

2.1 Kano model

The Kano model provides a quality function-deployment framework that aids products or service developers to take into consideration the customer's voice and preferences in the development phase instead of a passive approach of only developers [9–15] employed the Kano model for ICT system development and established that the model highlights user involvement. The model assists in determining basic, performance and delighters of a product or service.

In this paper, we conceptualize the customer as the software community (organization using the competencies) and the product or service as the needed competency. According to Kano et al. [16], customer's decision-making options on product or service acquisition, are founded on conscious and subconscious deliberations. For effective product and or service development there is the need to understand these deliberative conscious and subconscious processes of decision-

making. Kano et al.'s categorization of these processes into three-requirement levels (basic, performance and delighters) is relevant. For instance, basic requirements emanate from customer's expectations about a product or service, since their presence are immutable to influence customer options and opinion about the product. However, their absence may result in complaints from the customer. By extension, performance requirements, are expected pre-requisites knowledge factor vital in influencing customer decision-making options. These are critical pre-requisite requirements when appropriately adopted yields high levels of satisfaction. Meanwhile, at the delighter level, product and service developers are required to include surprise elements often referred to as 'wow' factors to entice, attract and influence customer choice options and preferences [16].

2.2 Competence framework for software engineers

Competency Framework for Software Engineers (CFSE) is a framework proposed by [17]. It identifies the training needs of software community and also serves as a guide for competency identification. The framework is divided into two main categories with sub-categories under main categories. The main categories are hard and soft competency. The soft competency category has socials and personals. The hard competency category has subcategories similar to roles for software development identified in SWEBOK. These includes project management, requirement analysis, software design, programming, validation and verification tests, configuration management, quality, tests, documentation and maintenance.

Our study, forms part of a broader software engineering competency study, which aims at creating classification maps for the satisfaction levels of software engineers' competencies. Specifically, in this paper, we focus on hard competency. Since CFSE serve the purpose of identifying hard soft competencies, we make use of the hard category side. This framework provides a granularity which align closes with the roles of software engineering. Thus, we make use of hard category aspect and the kano model to create our desired framework for the study. The result will be a unified framework to identify and classify the satisfaction levels of hard competencies for the use of the software engineering community.

3 Methodology and proposed framework

According to [18] framework as design science artifact requires some iteration in the validation of the process in developing. Justification for the need of the artifact has been presented through using literature, but it also requires stakeholder input, Thus, we present the proposed model for validation in this conference.

3.1 Propose Model: A Unified framework of Hard competency satisfaction levels for software engineers (UFHCSL)

This framework originates in the Kano model and CFSE. The Kano model as quality function-deployment model has been used for research work in software engineering.

Our study is the first to apply the Kano model on human resources as a means to determine the competency satisfaction levels of software engineers. CFSE is a framework for identifying competencies of software engineers, and there are more compatible frameworks available, such as [19–22] which provide a means to identify competencies of software developers. However, in line with our objectives, the CFSE framework provides required granularity and align with the roles of software engineering, we think the work of Rivera-ibarra et al. (2010) is suitable for our objectives.

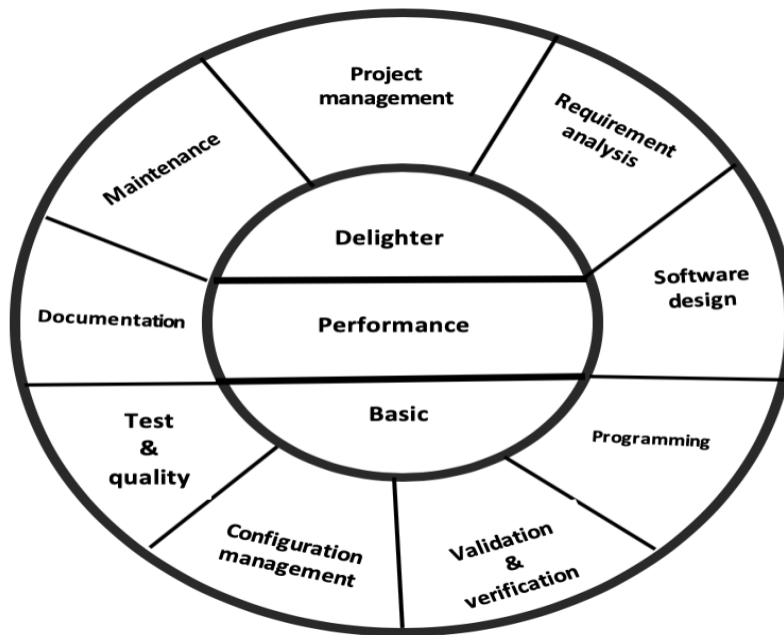


Figure 1. Unified framework of hard competency satisfaction levels for software engineers

To use the presented framework (UFHCSL), hard competencies are identified and classified using the hard category in [17] framework, followed by competency identification or classification subjected to the metrics of Kano model (we provide the metrics as table 1) to determine its satisfaction levels. The Categorization metrics is divided into three main parts (satisfaction levels): 1) basic, 2) performance and 3) delighter competencies. In each part a number of parameters are considered e.g. socials (interpersonal relations, cooperation and work in a team, and handling and conflicts resolution) and personals (development in the job, personal development, rights and limits).

4 Conclusion and future work

The proposed framework UFHCSL uses the kano model and the CFSE framework to create framework that can be use to identify hard competencies of software developers, their satisfaction levels and the most valued competencies of the developers. This framework add to the work of [23]. Thus, we have provided a framework that can be beneficial to educators, competency users, and possessors of hard competencies. The future work will be to use empirical data to evaluated the framework.

References

1. Sedelmaier, Y., & Landes, D. (2014). Software Engineering Body of Skills (SWEBOS). In *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 395–401)
2. Colomo-palacios, R., Carlos, U., & Madrid, I. I. I. De. (2010). The Case of Software Engineers Identifying Technical Competences of IT Professionals : *International Journal of Human Capital and Information Technology Professionals, 1*(March), 31–43
3. Turley, T., & Bieman, M. (1995). Competencies Nonexceptional of Exceptional and Software Engineers. *J. Systems Software, 1995:28*(28), 19–38.
4. Patel, A., Benslimane, Y., Bahli, B., & Yang, Z. (2012). Addressing IT Security in Practice : Key Responsibilities , Competencies and Implications on Related Bodies of Knowledge. In *2012 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 899–903)
5. Manawadu, C. D., Johar, M. G. M., & Perera, S. S. N. (2015). Essential Technical Competencies for Software Engineers : Perspectives from Sri Lankan Undergraduates. *International Journal of Computer applications, 113*(17), 27–34
6. Thurner, V., Axel, B., & Andreas, K. (2014). Identifying Base Competencies as Prerequisites for Software Engineering Education. In *IEEE Global Engineering Education Conference (EDUCON)* (pp. 1069–1076)
7. Lenberg, P., Feldt, R., & Wallgren, L. G. (2015). Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and Software, 107*, 15–37
8. Holtkamp, P., Jokinen, J. P. P., & Pawlowski, J. M. (2015). Soft competency requirements in requirements engineering, software design, implementation, and testing. *Journal of Systems and Software, 101*, 136–146
9. Lee, Y. C., Sheu, L. C., & Tsou, Y. G. (2008). Quality function deployment implementation based on Fuzzy Kano model: An application in PLM system. *Computers and Industrial Engineering, 55*(1)
10. Gangurde, S., & Patil, S. (2018). Benchmark product features using the Kano-QFD approach: a case study. *Benchmarking: An International Journal, 25*(2), 450–470
11. Huang, J. (2018). Application of Kano model and IPA on improvement of

- service quality of mobile healthcare Jui-Chen Huang, *16*(2).
12. Lehtola, L., & Kauppinen, M. (2006). Suitability of requirements prioritization methods for market-driven software product development. *Software Process Improvement and Practice*, *11*(1), 7–19
 13. Liu, X. F. (2000). Software quality function deployment. *Potentials, IEEE*, *19*(5), 14–16
 14. Piaszczyk, C. (2011). Model Based Systems Engineering with Department of Defense Architectural Framework. *Systems Engineering*, *14*(3), 305–326
 15. Richardson, I. (2001). Software Process Matrix: A Small Company SPI Model. *Software Process: Improvement and Practice*, *6*(Daft 1992), 157–165
 16. Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Kano. *Attractive Quality and Must-Be Quality. The Journal of the Japanese Society for Quality Control*, *14*, 39–48.
 17. Rivera-ibarra, J. G., Rodríguez-jacobo, J., Fernández-zepeda, J. A., & Serrano-vargas, M. A. (2010). Competency Framework for Software Engineers and. In *2010 23rd IEEE Conference on Software Engineering Education and Training* (pp. 33–40)
 18. Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst.*, *24*(3), 45–77
 19. Linck, B., Ohrndorf, L., Kiel, T. D. L., Magenheimer, J., & Neugebauer, J. (2013). Competence model for informatics modelling and system comprehension. In *2013 IEEE Global Engineering Education Conference (EDUCON)* (pp. 85–93)
 20. Tuffley, D. (2012). Optimising virtual team leadership in Global Software Development. *IET Software*, *6*(March 2011), 176–184
 21. André, M., Baldoquín, M. G., & Acuña, S. T. (2011). Formal model for assigning human resources to teams in software projects. *Information and Software Technology*, *53*, 259–275
 22. Schulte, C., Magenheimer, J., Kathrin, M., & Budde, L. (2017). The Design and Exploration Cycle as Research and Development Framework in Computing Education. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 867–876)
 23. Rivera-Ibarra, J. G., Rodríguez-Jacobo, J., & Serrano-Vargas, M. A. (2010). Competency Framework for Software Engineers. In *2010 23rd IEEE Conference on Software Engineering Education and Training* (pp. 33–40) 1