

**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:** Soft Competencies and Satisfaction Levels for Software Engineers : A Unified Framework

**Year:** 2020

**Version:** Accepted version (Final draft)

**Copyright:** © Springer Nature Switzerland AG 2020

**Rights:** In Copyright

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

**Please cite the original version:**

Assyne, N. (2020). Soft Competencies and Satisfaction Levels for Software Engineers : A Unified Framework. In D. Winkler, S. Biffel, D. Mendez, & J. Bergsmann (Eds.), *Software Quality : Quality Intelligence in Software and Systems Engineering. Proceedings of the 12th International Conference, SWQD 2020, Vienna, Austria, January 14–17, 2020* (371, pp. 69-83). Springer. *Lecture Notes in Business Information Processing*. [https://doi.org/10.1007/978-3-030-35510-4\\_5](https://doi.org/10.1007/978-3-030-35510-4_5)

# Soft Competencies and Satisfaction Levels for Software Engineers: A Unified framework

Nana Assyne

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

**Abstract.** The importance of software engineers' competency has long been established as a key pillar for the development of robust software in order to achieve quality software. Software engineering competency research is not necessarily lacking. Nevertheless, the satisfaction derived from using software competency needs more investigation. The aim of this study is to identify soft competencies from empirical data and create satisfaction levels for software engineers' soft competencies. The result shows 63 soft competencies with three different satisfaction levels consisting of basic, performance and delighters. The paper contributes to the SEC research by highlighting the satisfaction levels of soft competency for the benefit of the educators (academia), software engineers (possessor) and users of software competency (practitioner).

**Keywords:** Soft competency, Software engineers' competencies, Competency satisfaction levels, Essential competencies.

## 1 Introduction

The competencies of software engineers have long been recognized as essential for the development of efficient and robust software [1]. According to IEEE software engineering competency is defined as the knowledge, skills and attitudes of software developers to fulfill a task in a software development project [2]. This includes both soft and hard competencies [3]. Lenberg et al. pointed out that research work on software engineering competency (SEC) is not necessarily lacking. Yet, most of the earlier research on SEC focused on technical or hard competencies as against soft or behavioral competencies [4]. Harris & Rogers, define soft skills or competencies as "work ethics, positive attitude, social grace, facility with language, friendliness, integrity and the willingness to learn" [5, p.19]. Thus, the identification and use of soft competencies help in the development of complex software, because the software development involves a combination of soft and hard competencies [3, 6].

Works of authors such as Broadbent et al., Moreno et al., and Colomo-palacios et al. have established that soft competency is essential for development of software [6–8]. More importantly, recent literature suggests an increase in the number of software soft competencies studies with emphasis on identification of soft competencies [4].

Holtkamp et al. argued that, soft competencies are crucial for the development of global software engineering [9]. Nonetheless, the satisfaction levels of these competencies have not been adequately explored. Accordingly, this paper reports the identification and satisfaction levels of soft SEC as part of a bigger research on SEC.

The knowledge or identification of competencies is one phase of competency equation. The other phase is the benefit derived in the using such competency. The second phases have not received much attention in SEC research. Thurner et al., argue for minimum or base competency as a basic requirement for students of software engineering [10]. We support the base competency requirement and advocate for further investigations to determine the various levels of satisfaction of competencies. We therefore argue for satisfaction levels of soft competencies for software engineers, and state our research questions as:

*RQ1: What are the different satisfaction levels derived from using a software soft competency?*

*RQ2: Which of these soft competencies are perceived as most valuable for Software engineering?*

Knowledge of soft competencies and their satisfaction levels serve as insurance for users (people or organizations who use the competencies possessed by the developers to produce a product or a service), educator (people who train the developers to acquire the competencies), and the engineers' (people who receive training and therefore possesses some competencies). Therefore, we have adopted the Kano model [11] and Competency Framework for Software Engineers (CFSE) [12] as a lens to develop satisfaction rankings that can be employed by (i) the possessor of the competencies, (ii) users of the competencies and (iii) by the trainer of competencies possessors. The rest of the paper is presented as follows: section 2 looks at the background and related works and discuss the research models; section 3 discusses the methodology; section 4 presents the results; and section 5 and 6 looks at the discussions and conclusions respectively.

## **2 Theoretical foundation and Related Works**

### **2.1 Soft competency**

According to Harris & Rogers, soft skill is or are skills that mostly do not require formal training [5]. Until recently, these skills were mostly self-taught and self-developed. They are mostly not industry specific. In addition, they mostly require emotional intelligence [13][14] E.g. communication flexibility, leadership, motivation, patience, persuasion, problem-solving abilities, teamwork, time management, work ethics.

Soft competency connotes skills that complement technical skills; therefore, it cannot be overlooked in the development of software engineering. [They complement technical skills and thus cannot be overlooked in software engineering]. They are considered to be essential for global software projects [9, p.136]. (Broadbent et al.

established that the biggest skill gaps for software engineers were business strategies and marketing of their services [7]. This was emphasized by Moreno et al. [6]. Other studies have argued that more attention must be given to social and inter-personal competencies [15] and emotional intelligence [16].

In proposing a body of skills (SWEBOS) for software engineering, Sedelmaier and Landes identified and structured soft competencies of software engineers into three categories [3]. These include (i) comprehension of the complexity of software engineering processes, (ii) awareness of problems and understanding of cause-effect relationships, and (iii) team competency including communication skills. Although, this provided useful information that facilitates software development practices, it fails to provide relevant information regarding the satisfaction levels derived for possessing or using a competency. Evidently, there is a gap in existing literature. Perhaps, this is because researchers in the area of behavioral of software engineering have been focusing on few concepts [4] and ignore other relevant issues such as the assurance for using or possessing a particular competency

To address this, this study seeks to identify and also create a satisfaction level of the competencies from perspective of users, educators, and engineers. This will complement research on SEC in general and soft competency research specifically. To enable us to achieve our research objectives, we make use of CFSE and Kano model. The next sub-sections discuss CFSE and the Kano model.

## **2.2 Kano model**

The Kano model is a quality function-deployment framework that helps developers of product or service to include customer's voice in the development phase. It has been applied mostly in the development of products. This is because it takes into consideration the views of both the customer and developer in the development of a product instead of a passive approach of only developers [17]. [17–23] used the Kano model for the development of ICT system and concluded that the model prioritizes user involvement. It assists in determining basic, performance and delighters of a product or service.

In our scenario, the customer is the software community (organization using the competencies) and the product or service is the competency. According to Kano et al., customer's decision-making options on product or service acquisition, are based on conscious and subconscious deliberations [11]. There is therefore the need to understand these deliberative conscious and subconscious processes of decision-making to help develop products or services. Kano et al., categorized these processes into three-requirement levels (basic, performance and delighters). Basic requirements relate to customer's expectations about a product or service. These requirements are classified as basic since their presence are not dynamic enough to change the options and opinion a customer has about the product. However, their absence may result in complaints from the customer. Performance requirements, on the other hand, are expected pre-requisites that customers know and they are essential influential factors on the customer's decision-making options on products or services. These are critical pre-requisite requirements that create high levels of satisfaction when employed

appropriately and otherwise if not used. The last requirement termed delighters are those requirements that do not engender any complaints from the customers when absent however surprises the customer when present. Delighters are sometimes referred to as attractive or “wow” factors [11].

### **2.3 Competency framework for software engineers**

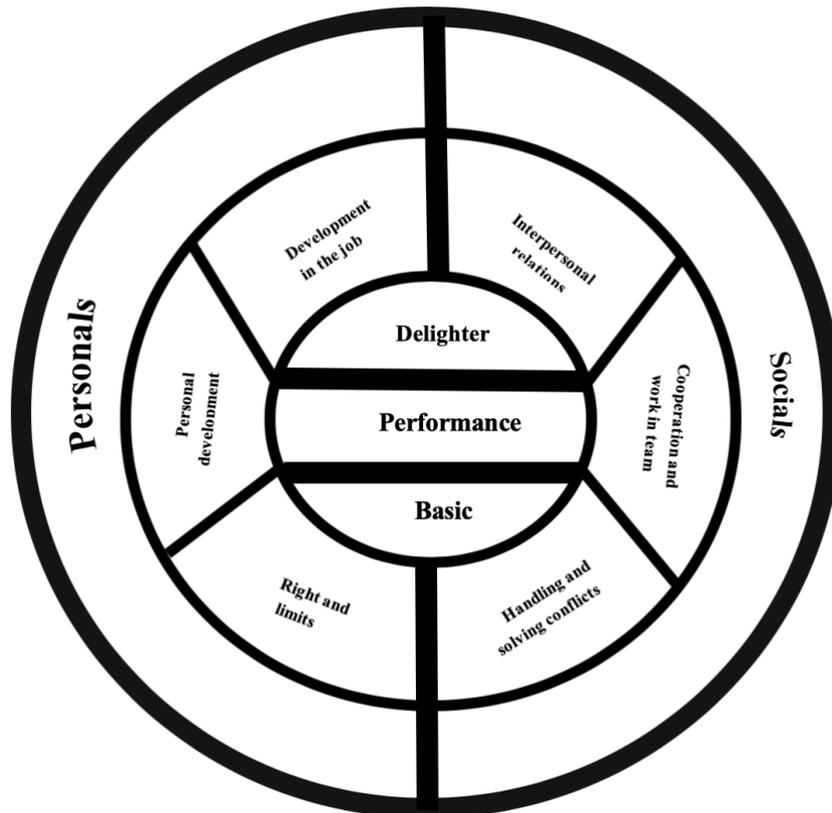
Competency Framework for Software Engineers (CFSE) is a framework that facilitates, identifies the training needs, and guides the design of software engineers' competencies. The design is based on the activities and interactions of engineers during the software development process. The constructs of this framework are under the main classification of competency (Hard and Soft). Hard competency category relates to the technical aspects of software engineering. These aspects are based on the definition of the SWEBOK roles in software engineering. They are project management, requirement analysis, software design, programming, validation and verification tests, configuration management, quality, tests, documentation and maintenance. The soft part of the categorization is classified into social and personal. Social aspects include interpersonal relations, cooperation and work in a team, and handling and conflicts resolution. Personals on the other hand includes development in the job, personal development, rights and limits. It can broadly be considered as “a set of knowledge, abilities and key behaviors, with special emphasis on the soft skills” [12].

The objective is to create a classificatory system that identifies and explains satisfaction levels of software engineers' competencies. Therefore, we consider the framework suitable. This is because it considers both soft and hard competency and this is the bigger objective we intend to achieve. Furthermore, the framework considers granularity, which is essential for fitting the work to the community. In line with the objective of this study, we focus on the soft competency aspect of the framework and merge it with Kano model. This resulted in a unified framework for identifying and classifying the satisfaction levels of soft competencies. For detailed analysis of the individual meanings of critical variables of CFSE, readers can refer to the original paper of [12]. The detail of the proposed framework for this paper is explained in the next section.

## **3 A Unified framework of Soft competency satisfaction levels for software engineers (UFSCSL)**

As mentioned earlier, the framework is derived from the CFSE and Kano model. From the CFSE we made use of the soft competency category since our aim is to identify and classify only the soft competency. From among frameworks such as [24–27] for identifying software engineering competencies CFSE framework is the one that has more granularity, thus making it easy for in-depth analysis. In addition, the Kano model has been used for research work in software engineering, but not for analyzing competencies. Thus, this provides a means to chart a new path for competency research. The soft part of the CFSE framework is first categorized into socials and personals and

each have lower granularity as shown in figure 1. The variables of the Kano model (basic, performance, and delighters) were included (see section 2.4), to provide the satisfaction levels for the competencies. See figure 1 for the “soft satisfaction levels of software engineers” framework.



**Fig. 1.** Unified framework of soft competency satisfaction levels for software engineers (UFSCSL)

To use the UFSCSL, first, the competencies are identified and classifying using the variables in [12] within the frameworks. Then each competency identified or classified is subjected to the metrics of Kano model to determine its satisfaction levels. Thus, given as basic, performance and delighter competencies for 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 Methodology

### 4.1 Data Collection

An exploratory qualitative study was adopted. Specifically, [28, 29] qualitative research guide was employed to extrapolate the required data. We agree with the philosophy that an individual's behavior is influenced by the meanings attached to events [30]. Thus, one hundred and thirty-eight (138) participants were drawn from workers in various positions within the industry: practitioner/software engineers/managers/supervisors/mentor. All participants were from software industries based in Norway. A semi-structured interview was used for data collection. Interviews were face-to-face and focused on expected skills of a software developer. Each interview session lasted for about 1 hours. The interview was conducted with the support of assistants. Table 1 represents the distribution of respondents' characteristics.

**Table 1.** Respondents Characteristics

Years of Experience	Category	Freq
	1 - 5	40
	6 - 10	17
	11 - 15	7
	16 - 20	15
	21 - 25	11
	26 - 30	13
	31 - 35	2
	36 - 40	2
	unspecified	31
Background	Software	72
	Hardware	11
	Research /university	11
	Others	19
	Unspecified	25

### 4.2 Data analysis

A thematic analysis offers an accessible means for organizing and describing a dataset under specific themes. Currently, there is no widely agreed way of going about how to use the method [31]. The soft competency satisfaction framework was therefore adopted to guide the analysis.

Both inductive analysis and deductive analysis were used. The coding of the data was done without any pre-defined framework. This enable the themes to emerge from the data. The framework (UFSCSL) was then applied to further code the theme that emerged from the data. Two categories were used on the bases of the epistemology of this research. That is, we were aware of the competencies that have been identified and exist in literature, but our epistemology was that within those identified there will be different satisfaction levels. Hence, we employed both categories in this paper. We

outline the following steps below based on the outcome of our analysis and guided by the steps of [31].

#### **Step 1 Familiarization of the data**

The interview was conducted with the help of assistants, with the aim of capturing large groups of respondents. Each interviewer transcribed his or her own interview. The author of this paper acquainted himself by reading through the transcribed scripts. During this stage, notes were taken in cases where there were difficulties in understanding aspects of the data. Further discussions were made with the head of data collection to resolve any ambiguity in the data.

#### **Step 2 Generating initial codes**

Initial codes were generated from the data by extracting keywords. This was done without recourse to initial pre-defined coding framework. The total number of competencies that were identified from the transcribed data were six hundred forty-one.

#### **Step 3 Searching for themes**

After the initial code, all initial codes were grouped into themes, this facilitated the identification of themes. These themes were generated without resort to pre-defined coding framework. Three hundred sixty soft competencies were identified at this stage.

#### **Step 4 Reviewing themes**

The themes were compared with existing themes. That is, a pre-defining coding framework was also used. In this case, the Rivera-Ibarra et al. [12] CFSE framework was used.

#### **Step 5 Defining and naming**

Next defined themes and meanings were assigned. These names and meanings were reviewed with literature before the competencies were validated using the variables in the Kano model. This stage resulted in 22 basics, 26 performance and 16 delighter competencies.

#### **Phase 6 Producing the report**

The emerged themes that resulted from comparing data themes and themes from the framework were used to produce the results discussed in the next section.

## **5 Results**

### **5.1 RQ1: What are the different satisfaction levels derived from using a software soft competency?**

We present the result in Table 2 using the framework (UFSCSL) developed for this paper. The results show the individual competencies and their satisfaction levels, that is: basic, performance, and delighters. They were grouped according to the broader

theme of soft competency: social and personal. We also provided definitions using the classification levels from the Kano model for the competencies.

**Table 2.** soft competencies and their satisfaction Levels

<b>Socials</b>		
	<i>Satisfaction levels</i>	<i>Software engineer competencies</i>
Interpersonal relations	Delight	(i) communicate to outside world, and (ii) sociable
	Performance	(i) communication skill, (ii) adaptability, (iii) human skill, and (iv) interpersonal skill.
	Basic	(i) social skills and (ii) contributing to the society.
Cooperation and work in team	Delight	(i) Excellent teacher, (ii) see bigger picture, and (iii) leadership.
	Performance	(i) team work, (ii) team organizer, (iii) approachable, (iv) open and communicating, (v) learn from others, and (vi) voice your own opinions.
	Basic	(i) Cooperation, (ii) maturity, (iii) teach and share knowledge, and (iv) dedication to work.
Handling and solving conflicts	Delight	(i) humbleness, (ii) customer awareness, and (iii) understand customer needs.
	Performance	(i) meeting skills, and (ii) contact with clients.
	Basic	(i) Listen ears, (ii) compromise, and (iii) empathy.
<b>Personals</b>		
Development in the job environment	Delight	(i) unafraid, (ii) creative and brave, and (iii) think outside the box.
	Performance	(i) persistence, (ii) flexible, (iii) versatile, (iv) focus, (v) accuracy, (vi) analytical skills, (vii) logical mindset and keep and overview, and (viii) creativity.
	Basic	(i) Willingness to learn, (ii) curious, (iii) passionate about your job, (iv) ask questions, (v) confidence, (vi) honest and responsible.
Personal development	Delight	(i) can apply theories in application, (ii) see opportunity in systems, (iii) initiative, (iv) separate work and being available, and (v) self-sufficient.
	Performance	(i) precise and detail oriented, (ii) self-reliance, (ii) independence (iv) understand needs for further development, and (v) know the working environments.
	Basic	(ii) pragmatic, (iii) patience, and (iii) open to new ideas.
Right and limits	Delight	-
	Performance	(i) attention to detail
	Basic	(i) Introspection and admit error, (ii) admit ignorance, and (iii) interest in the field.

### *Basic*

From the interview data and the analysis, basic competencies are pre-requisite competencies that are necessary and are expected by the users of the competency. Mostly they are taken for granted. Users see these competencies as natural when delivered properly. However, when delivered poorly, users will complain.

### *Performance*

From the interview data and the analysis performance competencies are what users expect and can articulate. They are mostly in the minds of the users and when they are delivered well, they create more satisfaction. These competencies can be described as “uni-dimensional” competency, in that the satisfaction grows exponentially when executed properly.

### *Delighters*

From the interview data and the analysis, the delighter competencies are unexpected by the user. Mostly unexpected by the user but increases the delight and surprise when available however its absence may have no effect on user.

## **5.2 RQ2: Which of these soft competencies are perceived as most valuable for Software engineering?**

As mentioned earlier, delighters are attractive or wow factors that valuable for the development of a product [11]. Therefore, we present our delighter competencies as the most valuable or essential competencies for software engineering. The table 3 shows the competency based on Rivera-Ibarra et al. CFSE framework [12]. The table shows the competency category and the identified essential soft competency for software engineers that are useful for software development.

**Table 3.** Most valuable competencies.

<b>Competency area</b>	<b>Competency name</b>
<b>Socials</b>	
Interpersonal relations	Communicate to outside world
	Sociable
Cooperation and work in team	Excellent teacher
	See bigger picture
	Leadership
<b>Personals</b>	
Handling and solving conflicts	Humbleness
	Customer awareness
	Understand customer needs
Development in the job environment	Unafraid
	Creative and brave

	Think outside the box
Personal development	Can applied theories in application
	See opportunity in systems
	Initiative
	Separate work and being available
	Self-sufficient

## 6 Discussions

Following our analysis, we aimed to provide a satisfaction level for the competencies identified from our primary data. A total of 63 competencies emerge from our data. Out of that 29 was for social competencies and 34 was for personal competencies with three satisfaction levels.

**Table 4.** Total number of soft competencies based on the satisfaction levels classification

<i>Competency area</i>	<i>Satisfaction levels</i>	<i>Total Number</i>	
<b>Socials</b>			
Interpersonal relations	Delight	2	8
	Performance	4	
	Basic	2	
Cooperation and work in team	Delight	3	13
	Performance	6	
	Basic	4	
Handling and solving conflicts	Delight	3	8
	Performance	2	
	Basic	3	
<b>Personals</b>			
Development in the job environment	Delight	3	17
	Performance	8	
	Basic	6	
Personal development	Delight	5	13
	Performance	5	
	Basic	3	
Right and limits	Delight	0	4
	Performance	1	
	Basic	3	
<b>Total</b>			

Table 4 shows the number of competencies and number of satisfactions of the competency area. Under socials competency area cooperation and work in team had 13 competencies, interpersonal relations and handling and solving conflicts had 8 competencies each. The cooperation and work in team competency reflect the team competency category of Sedelmaier and Landes [3]. Under personals competency area,

development in the job environment had 17 competencies, followed by personal development with 13 and right and limits with 4 competencies.

**Table 5.** Identified soft competencies and prior work.

<b>Category</b>	<b>Identified soft competency</b>	<b>Comparison</b>
<b>Socials</b>		
Interpersonal relations	Sociable, communication skill, adaptability, human skill, interpersonal skill, social skills	Consistent with prior work
	Communicate to outside world, contributing to the society,	New observations
Cooperation and work in team	See bigger picture, leadership, team work, Cooperation, teach and share knowledge, team organizer, approachable, open and communicating, learn from others	Consistent with prior work
	Maturity, Excellent teacher, voice your own opinions. dedication to work	New observations
Handling and solving conflicts	Customer awareness, understand customer needs, meeting skills, contact with clients, empathy	Consistent with prior work
	Humbleness, compromise	New observations
<b>Personals</b>		
Development in the job environment	Unafraid, creative and brave, think outside the box, persistence, flexible, versatile, analytical skills, creativity, Willingness to learn, curious, ask questions, confidence, focus, accuracy, logical mindset and keep and overview, honest and responsible	Consistent with prior work
	Passionate about your job	New observations
	Separate work and being available, self-sufficient,	Consistent with prior work

Personal development	precise and detail oriented, self-reliance, independence, pragmatic, patience, initiative, open to new ideas.	
	Can apply theories in application, see opportunity in systems, understand needs for further development, know the working environments	New observations
Right and limits	Attention to detail,	Consistent with prior work
	Introspection and admit error, admit ignorance, interest in the field	New observations

Some of the identified competencies in the categories are consistent with exiting literature such as the work of [3, 15, 16, 32–34]. Table 5 highlights the new observations and comparison to prior work. With regard to satisfaction levels, competencies were identified in all the categories except rights and limits delighters. A total of 16 essential competencies using the Kano model was identified. These competencies are consistent with literatures such as [35, 36]. Furthermore we have been able to create a satisfaction level, that adds to the works of [10, 37] that made argument for based competencies.

On the essential soft competencies for software engineers, we have been able use model analysis to extrapolate the essential competencies that are in agreements with the work of [35, 36, 38]. Thus, providing a new way of identifying essential competencies.

The novelty in this work are: (i) observations of new soft competency from empirical data that are highlighted in table 4 and (ii) the Unified framework of soft competency satisfaction levels for software engineers (UFSCSL). The UFSCSL has the ability to identify soft competencies of software engineers and also provide a satisfaction levels of the competency. Thus, serving as insurance model for users, possessors and the educators. In short, the major stakeholders of software engineering competency development are considered in this framework.

The study has both practical and research implications. From the perspective of the users of competencies, they can use the classification to determine which competencies will be valuable for employment. On the part of the possessor, they can use the classification levels to evaluate what they possess. Furthermore, educators can use the classification levels to adjust their training. Additionally, the framework which was proposed (UFSCSL) can be used for constant evaluation on old competencies and also on new ones.

## 7 Conclusion

The study has analyzed, identified and created a classification that can be used by the software community. This was done by synthesized existing relevant literature. The empirical work was based on Kano et al. [11] and Rivera-Ibarra et al. [12] CFSE framework. The study resulted in the identification of competencies, classification levels and essential competencies of software engineers. The study charts a new path of identifying essential or valued competencies of software engineers by using Kano model that has on been applied on products and services. Further studies should be done to understand how competencies within the satisfaction level can change.

The scope of the data collection was limited to companies situated in Norway; it may therefore limit the ability to generalize the findings universally. Nevertheless, most of the companies that the interviewees worked for has global representation and dealings outside Norway. With the development of competency satisfaction levels, we call for further studies to understand how specific competencies evolves within the satisfaction level.

### **Acknowledgement**

*The author would like to acknowledge Prof. Pekka Abrahamsson for his support in providing the dataset for this research and Dr. Hadi Ghanbari for his guidance.*

## References

1. Weinberg, G. M. (1971). *The Psychology of Computer Programming* (Silver Ann.). New York, New York 10027 Usa: Dorset House Publishing.
2. IEEE. (2014). *Software Engineering Competency Model (SWECOM)*. IEEE. Retrieved from <http://www.dahlan.web.id/files/ebooks/SWECOM.pdf>
3. Sedelmaier, Y., & Landes, D. (2014). Software Engineering Body of Skills ( SWEBOS ). In *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 395–401). IEEE. doi:10.1109/EDUCON.2014.6826125
4. 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. doi:10.1016/j.jss.2015.04.084
5. Harris, K. S., & Rogers, G. E. (2008). Soft Skills in the Technology Education Classroom : What Do Students Need. *Technology Teacher, 68*(3), 19–25.
6. Moreno, A. M., Sanchez-segura, M., Medina-dominguez, F., & Carvajal, L. (2012). The Journal of Systems and Software Balancing software engineering education and industrial needs. *The Journal of Systems & Software, 85*(7), 1607–1620. doi:10.1016/j.jss.2012.01.060
7. Broadbent, M., Dampney, C. N. G., Lloyd, P., & Hansell, A. (1992). Roles , Responsibilities and Requirements for Managing Information Systems in the 1990s. *International Journal of Information Managemet, 72*, 21–38.
8. Colomo-palacios, R., Casado-lumbreras, C., Soto-acosta, P., García-peñalvo, F. J., & Tovar-caro, E. (2013). Computers in Human Behavior Competence gaps in software personnel : A multi-organizational study. *Computers in Human Behavior, 29*(2), 456–461. doi:10.1016/j.chb.2012.04.021
9. 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. doi:10.1016/j.jss.2014.12.010
10. Thurner, V., Schlierkamp, K., Bottcher, A., & Zehetmeier, D. (2016). Integrated development of technical and base competencies: Fostering reflection skills in software engineers to be. In *IEEE Global Engineering Education Conference, EDUCON* (pp. 340–348). Abu Dhabi, UAE: IEEE. doi:10.1109/EDUCON.2016.7474576
11. 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.
12. 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). doi:10.1109/CSEET.2010.21
13. Andrews, J., & Higson, H. (2008). Graduate Employability, ‘Soft Skills’ Versus ‘Hard’ Business Knowledge: A European Study. *Higher Education in Europe, 33*(4), 411–422. doi:10.1080/03797720802522627
14. Trivellas, P., & Reklitis, P. (2014). Leadership Competencies Profiles and Managerial Effectiveness in Greece. *Procedia Economics and Finance, 9*(Ebeec 2013), 380–390. doi:10.1016/S2212-5671(14)00039-2
15. Licorish, S. A., & Macdonell, S. G. (2013). Differences in Jazz Project Leaders ’

- Competencies and Behaviors: A Preliminary Empirical Investigation. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (pp. 1–8). IEEE. doi:10.1109/CHASE.2013.6614725
16. Noorman, M., Akmal, M., Osman, F., & Ibrahim, Z. (2015). Malaysian Computer Professional: Assessment of Emotional Intelligence and Organizational Commitment. In *Procedia - Social and Behavioral Sciences* (Vol. 172, pp. 238–245). Elsevier B.V. doi:10.1016/j.sbspro.2015.01.360
  17. 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), 48–63. doi:10.1016/j.cie.2007.11.014
  18. Gangurde, S., & Patil, S. (2018). Benchmark product features using the Kano-QFD approach: a case study. *Benchmarking: An International Journal*, 25(2), 450–470. doi:http://dx.doi.org/10.1108/MRR-09-2015-0216
  19. Huang, J. (2018). Application of Kano model and IPA on improvement of service quality of mobile healthcare Jui-Chen Huang, 16(2).
  20. 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. doi:10.1002/spip.249
  21. Liu, X. F. (2000). Software quality function deployment. *Potentials, IEEE*, 19(5), 14–16. doi:10.1109/45.890072
  22. Piaszczyk, C. (2011). Model Based Systems Engineering with Department of Defense Architectural Framework. *Systems Engineering*, 14(3), 305–326. doi:10.1002/sys
  23. Richardson, I. (2001). Software Process Matrix: A Small Company SPI Model. *Software Process: Improvement and Practice*, 6(Daft 1992), 157–165. doi:10.1002/spip.144
  24. Orsoni, A., & Colaco, B. (2013). A Competency Framework for Software Development Organizations. In *2013 UKSim 15th International Conference on Computer Modelling and Simulation* (pp. 507–511). IEEE. doi:10.1109/UKSim.2013.101
  25. Acuña, S. T., & Juristo, N. (2004). Assigning people to roles in software projects. *Software - Practice and Experience*, 34(7), 675–696. doi:10.1002/spe.586
  26. 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). IEEE. doi:10.1109/EduCon.2013.6530090
  27. Tuffley, D. (2012). Optimising virtual team leadership in Global Software Development. *IET Software*, 6(March 2011), 176–184. doi:10.1049/iet-sen.2011.0044
  28. Mason, J. (2002). *Qualitative Researching*. *Qualitative Research Journal* (Vol. 41). doi:10.1159/000105503
  29. Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, 17(1), 2–26. doi:10.1016/j.infoandorg.2006.11.001
  30. Wohlin, C., & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, 20(6), 1427–1455. doi:10.1007/s10664-014-9319-7
  31. Braun, V., & Clarke, V. (2006). Full-Text. *Qualitative Research in Psychology*, 3(2), 77–101. doi:10.1191/1478088706qp063oa

32. Kropp, M., Meier, A., & Perellano, G. (2016). Experience Report of Teaching Agile Collaboration and Values Agile Software Development in Large Student Teams. In *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)* (pp. 76–80). IEEE. doi:10.1109/CSEET.2016.30
33. Robal, T., Ojastu, D., Kalja, A., & Jaakkola, H. (2015). Managing Software Engineering Competences with Domain Ontology for Customer and Team Profiling and Training. In *2015 Portland International Conference on Management of Engineering and Technology (PICMET)* (pp. 1369–1376). Portland International Conference on Management of. doi:10.1109/PICMET.2015.7273171
34. Samuelson, T., Colomo-palacios, R., & Kristiansen, M. (2016). Learning software project management in teams with diverse backgrounds. In *Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality – TEEM 16*.
35. Turley, T., & Bieman, M. (1995). Competencies Nonexceptional of Exceptional and Software Engineers. *J. Systems Software*, *1995:28*(28), 19–38.
36. Chang, J., Wang, T., & Lee, M. (2016). Impacts of Using Creative Thinking Skills and Open Data on Programming Design in a Computer-supported Collaborative Learning Environment. In *2016 IEEE 16th International Conference on Advanced Learning Technologies* (pp. 396–400). doi:10.1109/ICALT.2016.78
37. 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). doi:10.1109/EDUCON.2014.6826240
38. Suhartono, J., Sudirwan, J., & Background, A. (2016). Academic Competence of Computer Science Graduate Degree from the Employer ' s Perspective. In *2016 International Conference on Information Management and Technology (ICIMTech)* (pp. 176–181). IEEE. doi:10.1109/ICIMTech.2016.7930325