2010 23rd IEEE Conference on Software Engineering Education and Training

Competency Framework for Software Engineers

José Gamaliel Rivera-Ibarra, Josefina Rodríguez-Jacobo,
José Alberto Fernández-Zepeda
Department of Computer Science, CICESE
Km. 107 Carretera Tijuana-Ensenada, Ensenada, B.C. 22860, Mexico
{jrivera, jacobo, fernan}@cicese.mx

and

Miguel Angel Serrano-Vargas
MS SPI Solutions
1310 Summers End Ct., St. Louis, MO 63026, USA
miguel.serrano@msspisolutions.com

Abstract

One of the main concerns of the software industry is to develop the talent of its human resources, since the quality and innovation of its products and services depend to a great extent on the knowledge, the ability and the talent that software engineers apply in the software development process. A competency framework defines a set of knowledge, skills, and behaviors that professionals must have to excel in their careers. A competency framework facilitates the identification of training needs and guides the design of a professional development program. In this paper we propose a competency framework for software engineers, whose design is based on the activities and interactions that they perform during the software development process.

1. Introduction

Since its origins, the software industry faces many problems such as badly planned projects, breach of the delivery date, erroneous budget estimation and planning, uncontrolled system functionality changes, inappropriate documentation that generates unsatisfied clients. Research on software engineering confronts this type of problems by designing quality standards, working methodologies and technologies to increase the productivity of the software development process and the quality of software products. In spite of the multiple advances of these researches, they are not enough to prevent a high percentage of failing projects [10]. One possible cause of failing projects is that the quality of the software products and services do not only depend on the standards, methodologies, and technologies that are in use in the software development process, but also there is a great influence of the knowledge, the capability, and the talent of software engineers to apply the concepts of software engineering and to interact with diverse technologies and heterogeneous working groups.

The literature identifies the principal factors that cause low performance [5, 9] and the professional obsolescence [19, 12, 8, and 21]. It also mentions that sometimes the complexity of the software development process exceeds the capabilities of software engineers [14, 18, and 20]. However, there is little research that focuses on the skills update and professional development of software engineers. In addition, not everybody considers that software developing implies a complex process that presents technical-psychosocial interactions; consequently, software engineers must have the capability to work in team and under pressure, to satisfy the customer requirements, to solve

1093-0175/10 $26.00 © 2010 IEEE
DOI 10.1109/CSEET.2010.21

conflicts, and other qualities (such as stress handling and self-control) that they rarely try to develop, but that are fundamental for a good performance [15].

The goal of the present research is to propose a competency framework that guides software engineers to identify their training needs and to plan their professional development. The proposed competency framework is based on both the engineering software activities and on the characteristics that are required for a software engineer to perform those activities. The present work is mainly based on the information obtained from the empirical studies realized by the authors since 1994 to 2001 in software engineering training groups with a psycho-social approach. Our group traditionally does applied research in software engineering considering both technical and social factors with the aim of providing solutions to problems in software organizations.

This paper is organized as follows. Section 2 provides basic concepts. Section 3 briefly describes our methodology. Section 4 classifies competencies and defines the procedure and criteria we used for selecting competencies. In Section 5 we describe our proposed competency framework. Section 6 explains some possible scenarios where we can apply the proposed competency framework. Finally, Section 7 provides some concluding remarks and some research directions for future work.

2. General concepts

David McClelland [13] was one of the first to introduce the concept of competency. He detects that knowledge tests are not sufficient to predict the labor success of a professional and suggests that professionals need another type of qualities to effectively perform their job functions. The literature defines the concept of competency from two principal approaches, the functional and the behavioral. On the one hand, the competences from the functional approach (from the term "*competence*") refer to the professionals' knowledge and the ability to apply it to perform their job functions. These competences are known as *hard skills* [4]. On the other hand, the competencies from the behavioral approach (from the term "*competency*") refer to the characteristics of the professionals' behavior that allow them to excel in the performance of their job functions. These competencies are known as *soft skills* [4].

In this paper, we use the term competency in a holistic approach [3]; that is, we integrate in this term the functional and behavioral approaches. Thus, the term *competency* refers to the set of knowledge, abilities, and behaviors that professionals put in action in a specific context and that allow them to excel in the performance of their job functions and to fulfill the quality criteria that their job functions demand [11].

1) The *knowledge* (to know how to do) refers to the understanding of technical information (tool, phenomenon, methodology, etc.) necessary to adequately perform a job function [22].
2) The *ability* (to be able to do) refers to the cognitive factors that represent the capability to effectively apply the knowledge on a specific job function [22].
3) The *behavior* (to want to do) refers to the attitude that a professional shows as an affective positive or negative reaction towards an object (abstract or concrete) and that determines the way the professional acts.

Describing a working role by using competencies, facilitates organization's personnel management activities such as recruitment, career plan design, personnel performance evaluation, profile position design, compensation program design, training, and development program design [2]. A *competency framework* (or competency model) defines the set of knowledge, abilities and behaviors that professionals need to excel in the performance of their job functions. To elaborate a *competency framework* for the software engineer, we need to define the activities, roles, and interactions that are present in the software development process. *Software engineering* [7] is "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software." According to the SWEBOK [1], the activities of software engineering consist of ten knowledge areas, which we list in Table 1. The activities of software engineering are related to different disciplines, we list some of them in Table 2.

Table 1. Software engineering knowledge areas.

Software requirements	Software configuration management	Software construction	Software engineering process	Software quality
Software design	Software engineering management	Software testing	Software engineering tools and methods	Software maintenance

Table 2. Related disciplines of software engineering.

Computer engineering	Project management	Management	Software ergonomics
Computer science	Quality management	Mathematics	Systems engineering

In this paper we focus on the most important roles of the software engineer (see Table 3).

Table 3. Roles of software engineers.

a) Project manager	*c*) Software designer	*e*) Validation and verification engineer	*g*) Quality engineer	*i*) Documentator
b) Analyst	*d*) Programmer	*f*) Configuration manager	*h*) Test engineer	*j*) Maintenance engineer

3. Methodology

We studied nine groups of master's students in computer science during seven years. The average number of students per group was seventeen. We used the ethnographic technique (field observations, interviews and questionnaires). We spent three months to carry out the observations and interviews for each group. This period was the time assigned to each group to develop a project for a software engineering class. The projects had real clients and the work was distributed by role according to Table 3. The observations and interviews focused on identifying the knowledge, attitudes and skills that the students showed to perform their activities according to the software development process; more specifically, we were interested in those competencies that permit students to successfully conclude their projects. For the analysis of the information we used grounded theory [23], which is a method that starts by collecting data, then it classifies the data, and finally generates a theory. We also used functional analysis to identify core competencies that are fundamental to successfully complete the projects. The competencies are generic and are independent of the person who executes the task. The last two years of the study were devoted to verify the competencies and identify the level required for each student to successfully carry out the assigned activities, based on level of expertise specified in Table 5.

4. Classification and selection of competencies

For our convenience, we classify competencies under the following three categories:

1) *Technical*. These competencies describe the job functions performed by the software engineers; for example, obtaining the client requirements.
2) *Social*. These competencies describe the abilities of software engineers to interact with their work environments; for example, the ability to work in team.
3) *Personal*. These competencies describe particular characteristics of the software engineer; for example, the ability to learn by him/herself.

These categories are interrelated and they collectively are the base to identify the competencies to keep the software engineer in a competent level.

4.1 Criteria for selecting competencies

The competencies that we select come from two different sources. First, from the methodology described of Section 3; and second, from considering three factors: the market's needs, the characteristics of software engineers, and the characteristics of the organizations that hire them. On the one hand, software engineers need to rely on key competencies to adapt to environment where technologies change constantly and rapidly and working groups are now more diverse and fragmented than ever. On the other hand, companies need to satisfy the market's needs and to be profitable.

The framework that we describe in this paper focuses to individual competencies and not to collective capabilities of organizations or groups.

4.2 Selection of competencies

The work environments with the features mentioned above demands from the software engineer competencies with special characteristics. However, the selected competencies are those that are useful in different areas and are necessary for the all software engineers. The desirable characteristics of these competencies are:

1) They must relate to economical and social measurable benefits.
2) They must be useful in a wide spectrum of contexts; that is, they must be applicable to multiple areas of life.
3) They must be of specific use for a role, occupation, or task that the software engineer performs in the software development process.

5. Proposed competency framework

A competency framework consists of a specific set of competencies with an integrated approach. Table 4 shows the most important competencies for software engineers. We sort them by using two hierarchical levels. The first level divides competencies in groups: technical, social and personal. The second level divides each group in subgroups.

5.1. Why selecting technical competencies?

Software engineers require to have certain familiarity with technical knowledge and tools, as well as to understand how they change, how they can be applied to the software engineer's work environment, and how they can be used to pursue wider goals. In this respect, technical knowledge is an instrument to achieve an active dialog between software engineers and their environment. We divide technical competencies in two main subgroups as shown in Table 4.

5.1.1. Technical knowledge: The competency is necessary to understand options, to form opinions, to make decisions, and to carry out informed and responsible actions. To use knowledge and information in an interactive way, the software engineer needs.

1) To recognize and determine what they do not know.

2) To identify, locate and access to appropriate sources of information.
3) To evaluate the quality, suitability, and the value of an information and its sources.
4) To organize knowledge and information.

5.1.2. Use of technology: To use technology, software engineers require new forms of interaction with their daily job duties. Identifying those tools that support every role, may improve the productivity of the organization; it can also improve the flow of information inside the organization.

5.2. Why selecting social competencies?

In environments where software engineers work, it is important to know how to handle the interpersonal relations for mutual benefit and to construct new forms of cooperation. The competencies in this category are necessary in order that software engineers learn, live and work with others [16]. We divide the personal competencies in three main subgroups as shown in Table 4. This classification is based on the work of the project DeSeCo [17].

5.2.1. Interpersonal relation: This competency allows software engineers to initiate, to support and to handle personal relations with friends, colleagues and clients. This competency supposes that individuals can respect and appreciate the values, beliefs, cultures and histories of others to create an environment in which they feel welcome, they feel included, and in which they can grow [15].

5.2.2. Cooperation and work in team: Software development processes require people working cooperatively as a team. Cooperation demands that every software engineer has certain qualities. Software engineers must be able to balance their commitment with the group, their goals with their own priorities, and must be able to share the leadership and support others [16].

5.2.3. Handling and solving conflicts: Conflicts happen in all the aspects of life, at home, the workplace or in the community. Conflicts are part of the social reality and an inherent part of human relations. A conflict arises when two or more individuals or groups confront each other because of their needs, interests, goals or divergent values. The key to effectively handle a conflict is to face it, not to deny it. For this reason, it is necessary to consider the interests and the needs of others and the solutions in which both parts win [15].

5.3. Why selecting personal competencies?

This competency permits software engineers to act autonomously to effectively participate in the development of society and to work well in different spheres of life, including their workplace, their familiar and social lives. The software engineer supposes the possession of a firm concept of him\herself and the ability of translating the needs and the desires in acts of will: decision, choice and action. We divide the personal competencies in three main subgroups (see Table 4). This classification is based on the work presented in DeSeCo [17].

5.3.1. Personal development: This competency applies the concept of project management to the personal life of software engineers. It needs they analyze their lives to observe the directions that they are following and emphasize general aspects that are derived from this analysis. For example they need to be able to define a project and establish a goal, to identify and evaluate resources, to estimate future results, among others.

5.3.2. Rights and limits: This competency is important to secure the interests of individuals. In spite of many rights and limits are established and protected in laws or in contracts, at the end, individuals are the ones who identify and evaluate their rights, needs and interests. This competency relates to self-guided rights and limits.

Table 4. Competencies of software engineers and required level of expertise of each competency per role

COMPETENCIES OF SOFTWARE ENGINEERS	Roles (see table 3)									
	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*
Technicals										
Technical knowledge										
Analytical and learning capability in the following areas:										
Project management	3	2	2	2	2	2	2	2	2	2
Requirement analysis	2	3	2	2	2	2	2	2	2	2
Software design	2	2	3	2	2	2	2	2	2	2
Programming	2	2	2	3	2	2	2	2	2	2
Validation and verification tests	2	2	2	2	3	2	2	2	2	2
Configuration management	2	2	2	2	2	3	2	2	2	2
Quality	2	2	2	2	2	2	3	2	2	2
Tests	2	2	2	2	2	2	2	3	2	2
Documentation	2	2	2	2	2	2	2	2	3	2
Maintenance	2	2	2	2	2	2	2	2	2	3
Use of technology										
Evaluation and selection of tools to support influenced areas	3	3	3	3	3	3	3	3	3	3
Adaptation and use of tools to support influenced areas	3	3	3	3	3	3	3	3	3	3
Socials										
Interpersonal relations										
Communication	3	3	2	2	3	2	3	3	3	2
Adaptability	3	3	2	2	3	2	3	2	2	2
Aptitude to relate	3	3	2	2	2	2	2	2	2	2
Sociability	3	3	2	2	3	2	3	2	3	2
Interpersonal sensibility	3	3	2	2	3	2	3	2	3	2
Cooperation and work in team										
Understanding of the dynamics of debates and the follow-up of an agenda	3	3	2	2	2	2	2	2	2	2
Desires to contribute	3	2	2	2	2	2	2	2	2	2
Leadership	3	2	2	2	2	2	2	2	2	2
Motivation	2	2	2	2	2	2	2	2	2	2
Decision making to allow different opinions	3	3	2	2	3	2	3	3	2	2
The skill of presenting ideas and listening to the ideas of others	3	3	2	2	3	2	3	3	2	2
Orientation to achievement	2	2	2	2	2	2	2	2	2	2
Handling and solving conflicts										
Effective managing of the emotions	3	3	2	2	3	2	3	3	2	2
Aptitude to listen to others	3	3	2	2	3	2	3	3	2	2
Resolution of conflicts and initiatory	3	2	2	2	2	2	3	2	2	2
The skill of negotiating	3	3	2	2	3	2	3	3	2	2
Judgment(Reason), common sense and realism	3	3	3	3	3	3	3	3	3	3
Empathy	3	3	2	2	3	2	3	3	2	2
Personals										
Development in the job environment										
Capability to learn alone	2	3	2	2	2	2	2	2	2	2
Capability to search information	2	3	2	2	3	2	3	3	2	2
Capability to take risks	3	2	2	2	2	2	2	2	2	2
Flexibility	3	3	2	2	3	2	3	3	2	2
verbal Reasoning	3	3	2	2	3	2	3	3	2	2
Stress resistance	2	2	2	2	2	2	2	2	2	2
Proactive	3	3	2	2	3	2	3	3	2	2
Responsibility	3	3	3	3	3	3	3	3	3	3
Personal development										
To identify areas of personal opportunity	3	3	3	3	3	3	3	3	3	3
To define a project and to establish a personal goal	3	3	3	3	3	3	3	3	3	3
To determine priorities and to refine the goals	3	3	3	3	3	3	3	3	3	3
To identify and to evaluate available and required resources	3	3	3	3	3	3	3	3	3	3
Balance the necessary resources to satisfy multiple goals	3	2	2	2	2	2	2	2	2	2
Monitor the progress, to make the adjustments during the project development	3	3	3	3	3	3	3	3	3	3
To learn of past actions to project future result	3	3	3	3	3	3	3	3	3	3
High auto-esteem	3	3	3	3	3	3	3	3	3	3
Entrepreneur	3	2	2	2	2	2	2	2	2	2
Personal commitment	3	3	3	3	3	3	3	3	3	3
Self-control	3	3	2	2	3	2	2	3	3	2
Optimism	3	3	2	2	2	2	2	2	2	2
Rights and limits										
Ability to understand his/her own interests and needs	3	3	3	3	3	3	3	3	3	3
To know the rules and written principles to identify his/her limits	3	3	3	3	3	3	3	3	3	3
Ability of arguing in order that his/her rights and needs are recognized	3	2	2	2	2	2	2	2	2	2
To suggest arrangements or alternative solutions	3	2	2	2	2	2	2	2	2	2

6. How to use the competence framework

The proposed competency framework defines a set of knowledge, abilities and key behaviors, with special emphasis in the soft skills. The organization that adopts this framework must define and integrate the specific competencies needed by software engineers to allow them to fulfill their goals. To simplify the definition of this competency framework for the software engineer, we present Table 4 which shows specific competencies in each of the main groups described above. Organizations and individuals may use this table as a guide and to adapt it to their needs. Software engineers do not need to possess a total domain of each competency. We are proposing a four level scale (shown in Table 5) to evaluate the required level of expertise of a specific competency.

Table 5. Level of expertise of a competency.

Level	Percentage of the requirements that a software engineer satisfies with respect the requirements the organization establishes for a specific competency.	Level	Percentage of the requirements that a software engineer satisfies with respect the requirements the organization establishes for a specific competency.
0	*less than 25%*	2	*more than 50% and at most 80%*
1	*more than 25% and at most 50%*	3	*more than 80%*

Each column on the right side of Table 4 represents a software engineering role (see Table 3). The numbers represent the recommended level of expertise (see Table 5) of a competency for each role. For space limitations, we cannot provide a further explanation on this table.

Next we present five possible scenarios in which this framework may be useful.

1) *Getting a job*. This scenario corresponds to the period when software engineers try to get their first job.
2) *Preserving the job*. Software engineers must develop the necessary competencies to effectively fulfill the requirements of the role they play to preserve their jobs.
3) *Progressing in the same organization*. Software engineers sometimes are motivated to climb to some job that demands more responsibilities and new competencies.
4) *Changing to another organization*. When software engineers change their jobs and organizations, they need new competencies to perform the new job.
5) *Hiring personnel*. The organization wants to select, to evaluate, and to assign its workers to support its efficiency, to grow, and to face the challenges of the future.

7. Concluding remarks and future work

The objective of the competency framework is to support software engineers to develop and to advance in their work during their labor life in the organization. To face to the current challenges, software organizations need a better way of developing the skills of their software engineers to solve complex mental tasks. The competencies involve the mobilization of practical and cognitive skills, creative skills, and other psycho-socials resources like attitudes, motivation and values.

The attitudes or behaviors of software engineers and those of the organization are the strongest barriers that software engineers have to face if they want to develop themselves as professionals. The disposition of software engineers to adopt and acquire new competencies is a more definitive factor for their development as professional [6]. In order that software engineers could contribute to the organization, it is necessary that they prepare, form, and acquire the necessary competencies and obtain satisfaction from it; for this reason, it is recommendable to regularly evaluate their performance, to determine if their labor goals are in conformity with the market and their long term plans as people.

We are now working on designing a conceptual model for the development of competencies for software engineers and a strategy for the implementation of this model.

8. References

[1] Abran, A., Moore, J., Bourque, P., Dupuis, R., and Tripp, L., Guide to the Software Engineering Body of Knowledge - 2004 Version – SWEBOK, IEEE-Computer Society Press, April 2005, p. 200.

[2] Aragón, A., *et. al*., Fundamentos de dirección y gestión de recursos humanos, Cengage Learning Editores, 2004, p. 297.

[3] Argüelles, A., Competencia laboral y educación basada en normas de competencia, Editorial Limusa, México 1996, p. 319.

[4] Armstrong, M., A Handbook of Human Resource Management Practice, Kogan Page Ltd, 9Rev Ed edition, 2003, p. 979.

[5] Felder, R., Felder, G., and Dietz, E., "The effects of personality type on engineering student performance and attitudes". Journal of Engineering Education, **91** (1), 2002, pp. 3-17.

[6] Hansen, E., Orientación profesional: Un manual de recursos para países de bajos y medianos ingresos, Montevideo: CINTERFOR/OIT. EMP/SKILLS, 2006, p. 137.

[7] IEEE Standards Collection: Software Engineering, IEEE Standard 610.12-1990, IEEE, 1993.

[8] Joseph, D., and Ang, S., "The threat-rigidity model of professional obsolescence and its impact on occupational mobility behaviors of IT professionals". Proceeding 22nd International Conference on Information Systems, 2001, pp. 567-573.

[9] Karn, J., and Cowling, A., "An initial study of the effect of personality on group cohesion in software engineering projects", Department of Computer Science, Research Report,CS-04-01. University of Sheffield, 2001.

[10] Klappholz, D., Lawrence, B., and Port, D., "Assessing attitude towards, knowledge of, and ability to apply, software development process", Proceeding of the16th Conference on Software Engineering Education and Training (CSEET'03), 2003, p. 268.

[11] Klein, J., Spector, M., Grabowski, B. and de la Teja, I., Instructor Competencies: Standards for Face-To-Face, Online, and Blended Settings, Information Age Publishing, 2004, p. 164.

[12] Loo, J., Grip, A., and Steur, M., "Skills obsolescence: causes and cures", Proceedings International Journal of Manpower, MCB UP Ltd, **22** (1/2), 2001, pp. 121-138.

[13] McClelland, D., "Testing for competence rather than for intelligence." American Psych. (28) 1973, pp. 1-14.

[14] McGuire, E., and Randall, K., "Process Improvement Competencies for IS Professionals: A Survey of Perceived Needs", Proceedings of the 1998 ACM SIGCPR conference on Computer personnel research, March 26-28, Boston Massachusetts, 1998, pp. 1-8.

[15] Rodríguez, J., Una estrategia de cambio basado en las personas en una empresa de desarrollo de software, Guadalajara, 2003.

[16] Rodríguez, J., Formación de grupo de desarrollo de software: un enfoque psicosocial, Editorial Ediciones Yoltéotl, Guadalajara, 2004.

[17] Rychen, D., "Key competencies for the knowledge society", Education-Lifelong Learning and the Knowledge Economy Conference, Stuttgart, October 10-11, 2002, p. 13.

[18] Todd, K., "The effects of roles and personality characteristics on software development team effectiveness", doctoral dissertation, Faculty of Virginia Polytechnic Institute and State University, 1998.

[19] Trimmer, K., Blanton, J., and Schambach, T., "An Evaluation of Factors Affecting Professional Obsolescence of Information Technology Professionals," Thirty-First Annual Hawaii International Conference on System Sciences-Volume 6 (HICSS 98), 1998, p. 406.

[20] Trower, J., and Moore, K., "A study of the impact of individual goals and team composition variables on team performance", Proceedings of the 1996 ACM SIGCPR/SIGMIS conference on Computer personnel research, Denver Colorado, 1996, pp. 206-213.

[21] Tsai, H., Deborha, C., and Nicole, H., "A cognitive view of how IT professionals update their technical skills", Proceedings of the Conference on Computer personnel research: Careers, culture, and ethics in a networked environment (SIGMIS'04), April 22-24, 2004, pp. 70-73.

[22] Valdez, B., Svolou, A., and Valdez, F., "A Holistic Approach to Process Improvement Using the People CMM and the CMMI-DEV: Technology Process, People, & Culture, The Holistic Quadripartite", Software Engineering Information Repository. SEI and CMU, 2008; https://seir.sei.cmu.edu/seir/ (accessed: May of 2008.)

[23] Strauss, A. and Corbin, J., "Basics of Qualitative Research: Techniques and procedures for developing grounded theory", Sage. 1998.