Competence gaps in software personnel. A multi-organizational study

Ricardo Colomo-Palacios [a], Cristina Casado-Lumbreras [b], Pedro Soto-Acosta [c,⇑], Francisco J. Garcia-Penalvo [d], Edmundo Tovar-Caro [e]

[a] Computer Science Department, Universidad Carlos III de Madrid, Av. Universidad, 30, 28911 Leganés, Madrid, Spain
[b] Department of Development Psychology and Education, Universidad Complutense de Madrid, Rector Royo Villanova s/n, 28040 Madrid, Spain
[c] Department of Management & Finance, Universidad de Murcia, Campus de Espinardo, 30100 Espinardo, Murcia, Spain
[d] Computer Science Department, GRIAL Research Group, Universidad de Salamanca, Plaza de los caídos s/n, 37008 Salamanca, Spain
[e] Department of Software Engineering, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain

Abstract:

Today, the innovation and quality of the software industry's products and services depend to a great extent on the knowledge, ability and talent applied by software engineers. At the same time, human aspects are recognized as one of the main problems associated with software development projects. More specifically, inefficiencies usually come from inadequate verification of software engineers' competences. Another issue is the lack of an established career for software engineers, which adds difficulties to evaluate competences. With these challenges in mind, this paper presents a study conducted in the software industry to test competence gaps among software practitioners, comparing the 360-degree feedback results and self-evaluations with that of standard competence levels. The results of this research may be very valuable to organizations immersed in software development projects.

Keywords: Competence Gaps, Software workers, Career Ladder, Software Engineering

1 Introduction

Despite advances in technology and major shifts in economy, human resources remain an organization's most valuable resource (Saraswathy et al., 2011). Personnel have been proved to be crucial in the software industry, since software engineering involves people collaborating to develop better software (Lanubile et al., 2010). Recently, Colomo-Palacios et al. (2011a), following the path described by DeMarco and Lister (1987), identified human resources management as one of the main issues in software development. In fact, one of the software industry concerns is related to the development of its human resources talent because the quality and innovation of its products and services depend to a great extent on the knowledge, ability and talent applied by software engineers through the software development process (Rivera-Ibarra, Rodríguez-Jacobo & Serrano-Vargas, 2010). Therefore, information technology (IT) human resources are gaining importance in today's changing and more and more competitive environment (López-Fernández, Martín-Alcázar & Romero-Fernández, 2010) and have become crucial for the software engineering process (Polančič, Heričko & Pavlič, 2011).

In this scenario, management of people in software development projects is particularly critical (Liu et al., 2011) because human aspects are the source of the main problems associated with software development projects and there is abundant empirical evidence which confirms that (Hazzan & Hadar, 2008). Software development is recognized as a human centric and sociotechnical activity (Casado-Lumbreras et al., 2011) affected by personnel factors, which employs software practitioners who possess high levels of education, specific skills as well as the ability to apply skills to identify and solve problems (Ryan & O'Connor, 2009).

Individual differences have been identified as one of the paradigms for research analyzing human factors in software development (Curtis, 2002). These individual differences could rely, among other factors, on competencies (Sharp et al., 2009; Turley & Bieman, 1995). Although software practitioners through their competencies are the enablers of the Knowledge Society (Hernández-López et al. 2010), since they provide knowledge workers with the tool to perform their work, inadequate verification of software engineers' competencies is usually one of the main problems within software development projects (McConnell, 2003). Studies in the literature (e.g. Levy-Levoyer, 1996; Martin & Staines, 1994) have introduced taxonomies which differentiate between technical and generic competences. However, although software engineers may lack from technical competences, only a few studies in the literature (e.g. Colomo et al., 2011b) have analyzed discrepancies between standard competence level and real competence levels regarding generic competences. In addition, presumably discrepancies in competences may vary regarding the employees' position along the software engineering career. There is therefore a need to further investigate into software engineers' competences, considering both types of competences as well as the software engineering career levels. To address these issues, this paper presents a study conducted in the software industry to test competence gaps among software practitioners. These gaps are calculated comparing the 360-degree feedback results with that of standard competence levels. This assessment method comprises a process in which peers, supervisors and other external sources provide anonymous feedback (Atwater & Brett, 2005). This tool provides reports from multiple sources and has become a fundamental tool in personnel and human resources management (Massingham, Nguyen & Massingham, 2011).

Another issue is related to performance evaluation. In fact, Scullen et al. (2000) suggested that performance evaluation is not free from biases and later Patiar and Mia (2008) reported several investigations in which leniency and halo effects occurred. Thus, self-evaluations are expected to be above the 360-degree evaluations, producing the so called leniency effect.

Considering the above-mentioned points, the key research questions that motivated our work were:

- Do software engineers present more deficiency of technical competences than generic competences?
- Do discrepancies in competences differ along the software engineering career?
- Are self-evaluations systematically above 360-degree evaluations?

The remainder of the paper is organized as follows. Section 2 includes an overview of the origins and taxonomies of competencies, section 3 outlines the main software career ladders and section 4 describes the study conducted, including its research design, sample and main results. Finally, conclusions and future work are depicted in section 5.

2 Competence concept

Competences have gradually become a strategic issue in areas of academic research, business and education (Aramo-Immonen et al., 2011). Back in the 1990s the competence approach already marked a new and the importance of competences started to receive attention in the organizational context (Matthewman, 1995). Since that date, organizations are adopting more and more competence-centric human resource management approaches as standard practices (De Leenheer, Christiaens & Meersman, 2010).

The term "competence" has been applied in reference to many different behavior domains (Waters & Sroufe, 1983). Anderson and Messick (1974) catalogued 29 diverse competences ranging from specific skills (fine motor dexterity) to abstract concepts such as consolidation of identity. Though competence is often used in the sense of performance, however that is not entirely accurate (Bassellier, Horner Reich, & Benbasat, 2001). According to McClelland (1973), competence comprises the relation between humans and work tasks, that is, the issue is not about knowledge and skills itself, but knowledge and skills required to perform a specific job or task in an efficient way (McClelland, 1973). McClelland (1987) suggested that competences must become the basis for predicting individual performance more effectively in organizations.

Several authors (e.g. Levy-Levoyer, 1996; Martin & Staines, 1994) set up taxonomies in which particular or technical competences are referred as those that are necessary to carry out a very specific task in a particular job position, which include knowledge, abilities and skills. By contrast, universal or generic competences are those that, though not linked to a specific activity or function, do make possible the competent performance of the tasks related to the work position, since they involve characteristics or abilities of the individual's general behavior. These competences permit individuals to better adapt to changes in a more efficient and rapid way (Levy-Levoyer, 1996). Generic competences may be crucial not only for the success of IT projects (Sukhoo et al., 2005), but also for a wider range of organizational contexts, including all knowledge workers (Rimbau-Gilabert et al., 2009).

The identification of competencies that match job requirements has become a major issue in human resources development. In any context, competences needs should be identified in terms of gaps, which inform whether deficiencies in competences exist and, at the same time, reduce managers' subjectivity and preference in identifying managerial competence needs (Wickramasinghe & De Zoyza, 2009). A gap arises when a competence possessed by an individual is below than what is required for the expert to perform the task efficiently (Agut & Grau, 2002). According to Latham (1988) a perceived gap could sometimes be an expression of preference but, in any case, this information is useful. In this sense, when an organization finds a competence gap, this circumstance could be tackled, for instance, by redesigning jobs (Naquin & Holton, 2003). In sum, the competence approach and, more precisely, the competence gap analysis is useful for both organizations and practitioners.

3 A career ladder for software practitioner

In the information technology/information systems world, there is a specific career path, which includes several positions: programmer, analyst, IT manager and, eventually, the chief information officer (CIO) (Lee, 2005). However, the software engineering career is much less established (Downey, 2010), no clear role definitions have been agreed (Downey. 2009). In spite of this, significant initiatives such as the People Capability Maturity Model (People-CMM) (Curtis, Hefley, & Miller, 2009) point out the importance of establishing professional career structures defined, documented and driven by organizations. Recently, Colomo-Palacios et al. (2010) presented a career ladder for software professionals. This approach was based on an analysis, which focuses on extracting similarities between definitions for each professional profile from different sources such as industry practices (e.g. Curtis, Hefley, & Miller, 2009) and the technical literature (e.g. McConnell, 2003). The professional career is established starting from seven consecutive positions, giving concrete form to different levels of technical and general competences. Figure 1 shows the seven steps of the People-CMM career (Curtis, Hefley, & Miller, 2009) and the mappings to the software engineering technical career:

INSERT FIGURE 1

Once the software engineering career ladder is defined, the second step is to determine the set of competences, generic and technical, needed to perform the work in a competent way. The set of technical competencies were extracted from the Software Engineering Body of Knowledge (SWEBOK) areas (Abran & Moore, 2004), while generic competences were adapted from the Spanish White book for university degrees in computer science (Casanovas et al., 2004). Table 1 presents the standard level of generic and technical competences per role, which were extracted from two different sources: Colomo-Palacios et al. (2011b) and Colomo-Palacios et al. (2010).

The description of the scale (Likert-type) for all competences is as follows:

1 = Low Level

2 = Medium Level

3 = High level

4 = Very High level

INSERT TABLE 1

4 The study

4.1 Experimental Design

Authors applied the competence evaluation analysis to a set of professional roles: C, D, E and F. The experimental groups were required to have personnel with a minimum work experience of 1 year. This restriction was set in order to allow that 360-degree evaluations had sufficient data. The 360-degree feedback obtains comprehensive evaluations by considering all those that may reasonably comment on the performance of the individual evaluated, including self-assessment, assessment from below (subordinates/staff), assessment from peers or co-workers as well as assessment from external agents (Church, 2000). In sum, authors opted for the 360-degree evaluation because of its good results (Wood & Payne, 1998) and previous use for IT personnel evaluation (Jiang et al., 2001).

The groups' composition regarding work profiles presented a structure of seven professionals, which obey the hierarchical structure shown in Figure 2:

INSERT FIGURE 2

With regard to evaluations, results from an individual were based on the following stakeholders:

- His or her supervisor.
- One or more subjects sharing the same professional role.
- One or more subordinates.

To provide a consensus in a single indicator, an overall measure was calculated using the following formula:

C. Level = ((Avg. supervisor rate) + (Avg. peers rate) + (Avg.subordinates rate)) /3

This research instrument was pretested with 10 different practitioners with similar characteristics as those in the sample. The objective of the pretest was to improve procedures and ensure the proper design and procedural implementation of the experiment. The results from the pretest showed no particular bias.

4.2 Sample Description

The study was carried out over a period of three weeks. Participants were obtained from those who responded positively to a personal invitation, the sample consisted of 22 subjects, 5 women (23%) and 17 men (77%), from three different Spanish software corporations (a multinational IT consultancy corporation, a national IT consultancy corporation and a bank). The average age was 32.3. In average, subjects have 7.6 years of working experience. Subjects were classified on the previously mentioned roles, obtaining the following distribution: C (3), D (6), E(6) and F(7).

4.3 Threats to Validity

Two threats to the validity exist: internal or external. With respect to internal validity, the threat comes from the fact that the respondents may not have a comparable level of knowledge or expertise. Given that respondents were in all cases chosen because of their expertise and experience, authors made sure that experts possessed a comparable level of knowledge and expertise.

Regarding external validity, authors assumed two possible threats. The first is the small number of respondents, which makes difficult the generalization of results. The second is the fact that the sample was not taken randomly. Even though these threats exist, the sampling method and the number of respondents may be acceptable given the exploratory nature of our study.

4.4 Results

Results from the 360-degree feedback are presented in table 2 for all the competences, including the number of responses per role (N), average rating (Av) and standard deviation (Sd).

INSERT TABLE 2

With regard to technical competences, the role that presents more technical competences is “D”, presenting an overall rating of 27.9 points, followed by "E" with 25.2, "C" with 24.7 and, finally, "F" with 20.7. Software Construction, Software Requirements and Software Design are the most valued technical competences. Only one case presents a standard deviation over 1 point “Software Design” (role D) with 1.02 points.

Regarding generic competences, the role that leads the list is "C" with 68.5 points, followed by "D" with 64.8, "E" with 62.5 and, finally, the "F" with 61.1 points. On the top of the generic competences' ranking is the "Capacity for Analysis and Synthesis" with 13.2 points, followed by "Organization and Capacity Planning". At the bottom of the list are "Environmental sensibility" and "Leadership" with 8.8 and 8.9 points, respectively. Similar to what occurs with the technical skills, only a case presents a standard deviation above the unit, "Working in an international context" (role D), with a score of 1.02 points.

To analyze the differences between standard competence levels and the evaluations (real data) table 3 is calculated. The role that presents the most significant negative competence gap is "C" with and overall gap of 21 points, followed by "D" with 19.3, "E" with 10.4 and "F" with 0.2. These gaps are rooted on the greater exigencies usually requested to higher levels. With respect to the competences, two technical competences, Software Engineering Process and Software Engineering Tools, present the greatest gaps, with 7.6 and 4.7 points, respectively. In general, technical competences present greater gaps, with none of the technical competences presenting positive gap. In contrast, positive gaps appeared for several generic competences such as Understanding of other cultures and customs, Environmental sensibility and Interpersonal skills.

INSERT TABLE 3

Our first research question intends to respond whether software engineers present more deficiency of technical competences than generic competences. Adding total discrepancies and calculating the average mean for the case of 360-degree evaluations, average discrepancies of 26.7 points and 83.5 points were obtained for the technical and generic competences, respectively. Therefore, technical competences present a greater percentage of negative discrepancy than generic competences for the case of 360-degree evaluations.

Our second research question is concerned with examining whether discrepancies differ along the software engineering career. Table 4 shows information related to negative discrepancies by professional role, indicating the number of respondents. Results showed that "F" profile, with 7 subjects, is the position with less negative discrepancies, presenting a total number of discrepancies of 51 and an average mean of discrepancies of 7.3 points. D profile is the one that presented the greatest number of discrepancies both in terms of technical and generic competences as well as in average. Hypothesis 3 is confirmed and, thus, it can be concluded that subjects working in F positions present less negative discrepancies.

INSERT TABLE 4

Table 4 shows the differences between self-evaluations and 360-degree evaluations. It is important to notice the distinct number of cases between the two groups (3,267 for the 360-degree evaluations and 726 for self-evaluations) which indicates that additional analyses were needed. Our third research question is analyzing whether self-evaluations are systematically above 360-degree evaluations for technical and generic competences. Descriptive statistics indicate that differences between both types of evaluations may exist. Although, according to the results obtained, differences only exist for the case of generic competences.

INSERT TABLE 5

Furthermore, we explored whether discrepancies among technical competences existed. Table 6 shows that, for the 360-degree evaluations, software construction competence is the technical competence that presented fewer discrepancies on average, though the competence that had the fewest number of total discrepancies is software requirements with 7.16. With regard to self-evaluations, the competence that presented fewer discrepancies on average (as well as by total number) was software requirements.

INSERT TABLE 6

5 Conclusions

Personnel aspects have been recognized as the source of the main problems associated with software development projects. In this context, the management of human resources in software development projects is particularly critical. Inadequate verification of software engineers' competences is usually one of the main problems within software development projects. Another issue is the lack of an established career for software engineers, which adds difficulties to evaluate competences. These issues have not been sufficiently investigated and require further research. To address these gaps in the research, this paper presents a study conducted in the software industry to test competence gaps among software practitioners. More specifically, standard competence levels are compared with real competence levels from a set of software engineering organizations.

The results showed that the professional role that presents more technical competences is "D", followed by "E" and "C". Regarding generic competences, the role that leads the list is "C", followed by "D" and "E". These findings are coherent with the competence levels defined, since the D position is on the top of the technical competences pyramid, whereas C profile is on the top of the trapezoid formed by the generic competences. The implementation of the competence evaluation method has contributed to the knowledge of the subjects who took part in the experiment. Results from evaluations confirm that, generally, self-evaluations are above those from superiors, peers and subordinates. This finding is in line with existing research and the literature refers to it as the leniency effect (Agut & Grau, 2002; Atwater & Brett, 2005). However, it cannot be concluded that subjects from higher hierarchical level give lower punctuations than those from lower levels.

Furthermore, regarding negative competence discrepancies, results corroborate that technical competences present a greater percentage of negative discrepancies than generic competences for the case of 360-degree evaluations. Despite this result, in software engineering, the human dimension sometimes has greater importance than the technical dimension (Constantine, 2001), since it is an activity based on intellectual and social interaction. In fact, there is abundant empirical literature which confirms that human aspects are the source of the main problems associated with software development projects (Hazzan & Hadar, 2008). In addition, results show that subjects from lower positions present less negative discrepancies than those from higher hierarchical levels both for technical and generic competences. The software engineering competence is the technical competence that presents the large number of negative discrepancies for all types of evaluations and, therefore, the one that requires more training. This circumstance may be explained due to the relatively recent research in software engineering, which started in the beginning of the 1990s. Contrary to what hypothesized, software construction was not the technical competence that presented the fewer number of negative discrepancies. The technical competence that presented the fewer number of negative discrepancies was software requirements.

Acknowledgements

We would like to thank the two anonymous reviewers for their highly constructive comments and suggestions which allowed us to furthering the work.

References

Abran, A., & Moore, J.W. (2004). SWEBOK—Guide to the Software Engineering Body of Knowledge. 2004 Version, IEEE CS Professional Practices Committee, 2004.

Agut, S., & Grau, R. (2002). Managerial competency needs and training requests: the case of the Spanish tourist industry. *Human Resource Development Quarterly, 13*(1), 31-51.

Anderson, S. & Messick. S. Social competency in young children. *Developmental Psychology, 10*(2), 282-293.

Aramo-Immonen, H., Bikfalvi, A., Mancebo, N., & Vanharanta, H. (2011). Project Managers' Competence Identification. International *Journal of Human Capital and Information Technology Professionals, 2*(1), 33-47.

Atwater, L.E., & Brett, J.F. (2005). Antecedents and consequences of reactions to developmental 360° feedback. *Journal of Vocational Behavior, 66*(3), 532-548.

Bassellier, G., Reich, B.H. & Benbasat, I. (2001). IT Competence of Business Managers: A Definition and Research Model. *Journal of Management Information Systems, 17* (4), 159-182.

Casado-Lumbreras, C., Colomo-Palacios, R., Soto-Acosta, P., & Misra, S. (2011). Culture dimensions in software development industry: The effects of mentoring. *Scientific Research and Essays, 6*(11), 2403-2412.

Casanovas, J., Colom, J.M., Morlán, I., Pont, A. & Ribera, M., (2004). Libro Blanco sobre las titulaciones universitarias en Informática en España [[White Book: University degrees in computer engineering]. Madrid: ANECA.

Church, A.H. (2000). Do higher performing managers actually receive better ratings? A validation of multirater assessment methodology. *Consulting Psychology Journal: Practice* *and Research, 52*(2), 99-116.

Colomo Palacios, R., Tovar Caro, E., García Crespo, A., & Gómez Berbís, J.M. (2010). Identifying technical competences of it professionals: the case of software engineers. *International Journal of Human Capital and Information Technology Professionals, 1*(1), 31-43.

Colomo-Palacios, R., Casado-Lumbreras, C., Tovar, E., & Soto-Acosta, P. (2011a). A step towards human capital management in the software industry based on generic competencies. International Journal of Strategic Change Management, 3(4), 247-259.

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P., & Sabbagh, M. (2011b). Software product evolution for Intellectual Capital Management: The case of Meta4 PeopleNet. *International Journal of Information Management, 31*(4), 395-399.

Constantine, L. L. (2001). Peopleware Papers: The Notes of the Human Side of Software, Prentice Hall.

Curtis, B. (2002). Human factors in software development. In Marciniak, J.J. (Ed.): Encyclopedia of Software Engineering, pp.598-610, Willey & Sons.

Curtis, B., Hefley, W. E., & Miller, S. A. (2009). People Capability Maturity Model (P-CMM®) Version 2.0. (No. CMU/SEI-2009-TR-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellow University.

De Leenheer, P., Christiaens, S., & Meersman, R. (2010). Business semantics management: A case study for competency-centric HRM. *Computers in Industry, 61*(8), 760-775.

DeMarco, T., & Lister, T. (1987). Peopleware: Productive Projects and Teams. New York: Dorset House.

Downey, J. (2009). Career Paths for Programmers: Skills in Senior Software Roles. Cambridge Scholars Publishing, Newcastle: England.

Downey, J. (2010). Careers in software: is there life after programming?. In Proceedings of the 2010 Special Interest Group on Management Information System's 48th annual conference on Computer personnel research on Computer personnel research. ACM Press.

Hazzan, O., & Hadar, I. (2008). Why and how can human-related measures support software development processes. *The Journal of Systems and Software*, *81*(7), 1248–1252.

Hernández-López, A., Colomo Palacios, R., García Crespo, A., & Soto-Acosta, P. (2010). Trust Building Process for Global Software Development Teams. A review from the Literature. *International Journal of Knowledge Society Research, 1*(1), 65-82.

Jiang, J. J., Klein, G., Roan, J., & Lin, J. (2001). IS service performance: self perceptions and user perceptions. *Information & Management, 38*(8), 500-506.

Lanubile, F., Ebert, C., Prikladnicki, R. & Vizcaíno, A. (2010). Collaboration Tools for Global Software Engineering. *IEEE Software, 27*(2), 52-55.

Latham, G.P. (1988). Human resource training and development. *Annual Review of*

Psychology, 39, 545-589.

Lee, C.K. (2005).Transferability of Skills over the IT Career Path. In J. E. Moore and S. E. Yager, Eds., 2005 ACM SIGMIS CPR Conference, Atlanta, Georgia, ACM Press, p. 85-93.

Levy-Leboyer, C. (1996). La gestion des compétences [Competence management]. Paris: Les Editions d'Organisation.

Liu, J.Y.C., Chen, H.G., Chen, C.C., & Sheu, T.S. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management, 29*(5), 547-556.

López-Fernández, M., Martín-Alcázar, F., & Romero-Fernández, P. M. (2010). Human resource management on social capital. *International Journal of Human Capital and Information Technology Professionals, 1*(2), 36–48.

Martin, G., & Staines, H. (1994). Managerial competences in small firms. *Journal of Management Development, 13*(7), 23–34.

Massingham, P., Nguyen, T.N.Q., & Massingham, R. (2011). Using 360 degree peer review to validate self-reporting in human capital measurement. *Journal of Intellectual Capital, 12*(1), 43 – 74.

Matthewman, T. (1995). Trends and development in the use of competency frameworks. *Competency, 2*(49), 1-20.

McClelland, D.C. (1987). Human Motivation. Cambridge University Press.

McClelland, D.C. (1973) Testing for competence rather than for 'intelligence'. *American Psychologist, 28*, 1–14.

McConnell, S. (2003). Professional Software Development. Addison-Wesley, Reading, MA.

Naquin, S.S., & Holton, E.F. (2003). Redefining state government leadership and management development: a process for competency based development. *Public Personnel Management, 32*(1), 23-46.

Polančič, G., Heričko, M., & Pavlič, L. (2011). Developers' perceptions of object-oriented frameworks – An investigation into the impact of technological and individual characteristics. *Computers in Human Behavior, 27*(2), 730-740.

Rimbau-Gilabert, E., Miyar-Cruz, D., & López-de Pedro, J.M. (2009). Breaking the boundary between personal- and work-life skills: parenting as a valuable experience for knowledge workers. *International Journal of Knowledge and Learning, 5* (1), 1-13.

Rivera-Ibarra, J.G., Rodríguez-Jacobo, J., & Serrano-Vargas, M.A. (2010). Competency framework for software engineers. In Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training (CSEE&T), IEEE Press, pp. 33 - 40

Ryan, S., & O'Connor, R.V. (2009). Development of a team measure for tacit knowledge in software development teams. *The Journal of Systems and Software, 82*(2), 229–240.

Saraswathy, R., Thamaraiselvan, N., Senthilarasu,, B., & Sivagnanasundaram, M. (2011).

Facades of Attractive Employer in Indian IT Industry: Existing Employee Perspective. *International Journal of Human Capital and Information Technology Professionals, 2*(1), 80-98.

Sharp, H., Baddoo, N., Beecham, S., Tracy, H., & Robinson, H. (2009). Models of motivation in software engineering. *Information and Software Technology, 51*(1), 219-233.

Sukhoo, A., Barnard, A., Eloff, M.M., Van der Poll, J.A., & Motah, M. (2005). Accommodating Soft Skills in Software Project Management. *Issues in Informing Science and Information Technology, 2*, 691-704.

Turley, R.T., & Bieman, J. M. (1995). Competencies of exceptional and nonexceptional software engineers, *Journal of Systems and Software, 28*(1), 19–38.

Waters, E. & Sroufe, L. (1983). Social Competence as a Developmental Construct. *Developmental Review, 3*(1), 79-97.

Wickramasinghe, V., & De Zoyza, N. (2009). A comparative analysis of managerial competency needs across areas of functional specialization. *Journal of Management Development, 28*(4), 344 – 360.

Wood, R. & Payne, T. (1998). Competency-based Recruitment and Selection. Wiley & Sons.