The Human Factor in Software Engineering

Luis Fernández-Sanz and María-José García-García

Software Engineering (SE) is a well established and recognized discipline with a significant influence on the software development world and on Information Technology (IT) in general. Many papers have highlighted SE as the basis for a differentiated set of professional profiles in the field of computing, and have stressed the influence of Human Resources (HR) on software productivity, but neither the standards nor the many empirical studies so far produced have provided an in-depth and wide-ranging analysis of the various specialization roles involved in a project. In this paper we take the connection between productivity and staff profiles and training as a given, and present an overview of the facts and figures concerning the definition of roles and positions in software development.

Keywords: Human Resources, Job Profiles, Productivity, Software Engineering.

1 Introduction

Companies tend to say that "*human resources are our main asset*". This is especially true in software development where cost estimation models and methods recognize people as the main resource and cost: in fact, estimation models are based on the estimation of human effort (work hours or months) to calculate the cost of project [1]. "Software engineer" is of course recognized as a separated and clearly defined IT (Information technology) job profile [2], but the people involved in software development projects play a great variety of roles and it is normal to differentiate between a number of different positions. We therefore need to study the different characteristics (relevant to each position or role) that influence people's ability to contribute to productivity and quality in software development:

- Academic education and specific training courses: our studies of IT job applicant requirements in Spain [3] clearly show the rapid evolution of and changes in the required knowledge of technical matters, and the increasing need to demonstrate personal skills such as the ability to work in teams, communication skills, proactive attitude, etc..
- Specific experience and skills related to technical environments, programming languages, data base systems, etc.
- Deployment of best practices for personal development in software projects that influence productivity and quality: e.g. PSP [4].

Neither can we ignore the strong influence exerted by motivation and a culture of quality, the spirit of excellence, etc. which are especially dependent on an ethical attitude such as is set out in the IEEE code of ethics for software engineers (see <http://www.computer.org/certification/ethics.htm>).

Although the study of people's work is in itself an exciting discipline, we need to take a special interest in analysing the effect of a number of factors related to Human Resources (HR) on software development productivity. If people are the main cost driver, it is important to study all the relationships between developers and productivity, quality, or any other commonly established goal for software projects. Productivity in particular is a much debated subject because software development is regularly blamed for low productivity. In [5] for example, various experts comment on a report published in Business Week saying that software productivity has decreased by 0.9 percent a year in the United States (the lowest productivity of all other sectors or activities). Most experts point to the impact of technological advances, new and more powerful tools, and automation, but only in very few cases do they mention the

Authors

Luis Fernández-Sanz received a degree in Informatics Engineering from the *Universidad Politécnica de Madrid*, Spain, in 1989 and a PhD in Computer Science from the *Universidad del País Vasco*, Spain, in 1997 (receiving a special mention for his doctoral thesis). Since 2000 he has been the head of the Computer Systems Dept. at the *Universidad Europea de Madrid*, Spain. Since 1992, he has been editor of the Software Engineering section of ***Novática***, and editor and founder of REICIS, Spanish Journal of Software Quality, Engineering and Innovation, <http://www.ati.es/reicis>, both published by the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*). He has guest-edited several monographs for **Novática** and **UP**GRADE, and has authored or co-authored several books about Software Engineering and software measurement, as well as a number of papers in international journals and conferences. He leads the Software Quality Special Interest Group of ATI in which capacity he has acted as chair of the Spanish Conference on Software Quality and Innovation organized by ATI. <luis.fernandez@uem.es>

María-José García-García received a degree in Computational Mathematics from the *Universidad Complutense de Madrid*, Spain. She has worked at the *Universidad Europea de Madrid* since 1998, and since 2005 she has been acting as a coordinator of the Informatics Engineering degree course. Author of several papers in a number of journals and conferences, she is completing her doctoral thesis on computing professional profiles in software development. <mariajose.garcia@uem.es>

Cost driver	*Very low*	*Low*	*Nominal*	*High*	*Very high*
Analyst capability[2]	1.42	1.19	1.00	0.85	0.71
Programmer capability[3]	1.34	1.15	1.00	0.88	0.76
Platform experience	1.19	1.09	1.00	0.91	0.85
Application experience	1.22	1.10	1.00	0.88	0.81
Language and tools experience	1.19	1.09	1.00	0.91	0.85
Staff continuity[4]	1.29	1.12	1.00	0.90	0.81

Table 1: People-related Cost Drivers in COCOMO Models.

problems related to process improvement and the need for caution when incorporating new tools into organizations. And only one of them makes mention of the problem of communication barriers and the fact that the rate of adoption of new tools (and therefore processes) by developers is very low[1] . There were no references at all to HR, training, education, motivation, or experience.

2 HR and Productivity

Various papers have attempted to formalize the relationship between people and productivity by addressing such issues as the skill and experience of developers. The well-known COCOMO [6] and COCOMO II models [7] include the factors related to HR shown in Table 1.

Two facts become apparent when analysing this model (leaving aside any discussion regarding the accuracy of the productivity influence values assigned to each factor):

■ The negative influence of some factors (i.e. the low skills of analysts or programmers, or their little experience in the target environment) is greater than the corresponding positive influence on costs. For example, bad quality analysts may cause a 42% rise in costs over the nominal/standard effort but good quality analysts only reduce costs by 29%.

■ The guidelines used to rank personnel are based on subjective methods where criteria include percentile evaluation (e.g. analyst capability is ranked as very low if it is in the 15th percentile according to a subjective evaluation of different skills) and a number of ranges for experience (e.g. less than 2 months counts as very low for experience related cost drivers).

[1] Three common factors were detected: the lack of any industry-wide standard definition for software productivity, the increasing complexity of software applications, and the need for more formalized processes in the industry as a whole.

[2] "*Analysts are people who work on requirements, high-level design and detailed design*" [7]. Evaluation criteria include analysis and design ability, thoroughness and efficiency, communication and cooperation skills (experience is excluded)

[3] The evaluation should be based on the capability of the team rather than that of the individuals. Criteria are similar to those applied to analysts.

[4] Based on turnover rate: >48%/year counts as very low continuity, <3%/year counts as very high.

Another interesting example of the quantification of the influence of several factors in productivity is presented in [8] (see Table 1). We can see that the influence of factors is generally greater when there is a negative influence (e.g. inexperienced developers: -87%) than when that influence is positive (e.g. experienced developers: +55%). Of course, many other factors closely related to HR (not only skill and experience) have a real influence on productivity: e.g. office ergonomics, motivation/morale, etc. (See Table 2.)

An important aid to understanding the relationships between the different factors affecting software development projects is Abdel-Hamid's microworld-based software development model [9]. This is a good tool for simulating the real behaviour of development projects. It was developed on the basis of interviews with software project managers in five major organizations and an extensive database of empirical findings from relevant literature. It includes four main subsystems, one of which is devoted to human-resources management, dealing with hiring, assimilation and transfer of people. This subsystem also breaks the workforce down into various employee categories (for example; newly hired, experienced workers). It also reflects the fact that the training process to assimilate new employees causes veterans' productivity to decline, as they are usually the ones to train newcomers. Other factors such as workforce stability and size tend to evolve as the project completion date nears and the project progresses. Such parameters as hiring rate, turnover rate, and workforce experience mix influence the project team structure which, in turn, has an effect on such aspects as potential productivity, software development rate, and project losses. However, the full model of Abdel-Hamid has more than 100 causal links between the different elements of the development world.

This model assumes the following equation [10]:

Productivity = potential productivity - losses due to faulty processes

Potential productivity refers to when an individual or group makes the best use of their available resources and is a function of the nature of the task and the team's resources. Average productivity is a weighted average of two nominal productivity rates; one for experienced staff and one for newly acquired staff. There are also problems such as motivation (development and promotion, salary, responsibility,

Factors	Positive impact (+%)	Negative impact (-%)
Staff experience[5]	High +55%	Low –87%
Management experience	High +65%	Low –90%
Office	Ergonomics +15%	Crowded space –27%
Unpaid overtime	Yes +15%	No 0%
Work	Specialist +18%	Generalist –15%
Morale	High +7%	Low –6%
Organization	Hierarchical +5%	Matrix–8%

Table 2. Impact of Key Adjustment Factors on Productivity.

recognition, etc.) and communication overhead.

We should remember that each communication link requires effort and time; in fact, at least 30% of programmers' time is devoted to working alone [11][12] and a number of attempts have been made to devise better metrics for communication [13].

Several conclusions have emerged as consequence of a great deal of study and debate on this subject.

For example, the inclusion of people late in a project is often disruptive [14]. The new people need to learn the system, and their teachers are none other than those already doing the work, while new communication paths also need to be established.

As shown in [9], the learning curve of new environments has a strong influence on productivity. Several papers (with empirical data) have highlighted the different aspects that need to be addressed by people management [15][16].

Various and sometimes radical rules-of-thumb have emerged from some of these studies: e.g. it is better to fire an incompetent programmer than to bring in another one [17] or that a good programmer can be 5 to 10 times more productive than an average worker [16].

We can therefore see that HR factors have a real influence on productivity and that the various proposals mentioned above have attempted to quantify that influence. But we believe that a more detailed insight is needed to establish a solid basis for this type of study. There are a great many different job positions and roles involved in a software development project.

Each of them has a very different profile which needs to be analysed if any further studies on productivity are to make sense. In the following section we present an analysis of a number of references (process standards, methodologies, job classifications, etc.) in which these positions and roles are supposedly described.

[5] Experience is evaluated in terms of application type, programming language, and programming tools or environment.

[6] It should be noted that the ISO 15504 [17] life cycle processes description is closely based on ISO 12207.

[7] In each iteration, a mini-cascade cycle is followed by the following phases (workflows): Requirements capture, Analysis, Design, Implementation, Testing.

3 Profiles, Positions, and Roles in Software Development

In this overview we will take a look at a number of viewpoints regarding descriptions and profiles of the jobs and roles involved in software development projects. As can be seen below, our overview covers classifications that have arisen from process standards, software methodology handbooks, and job categories and certifications. Our analysis of this selection of documents is focused on finding similarities and differences in the different worker profiles.

3.1 Process Standards

In the discipline of software development, a number of standard process descriptions have been defined as a formal reference framework to describe activities during projects. The following are particularly worthy of mention:

- ISO/IEC 12207:1995[6] [18] establishes a framework for software life cycle (SLC) processes. It sets out in detail the processes, activities and tasks suggested for SLC and also describes how to tailor the standard for a specific project.
- IEEE Std 1074-1997 [19] provides a process for creating an SLC process model including detailed descriptions of suggested activities and processes.

There are practically no descriptions of the profiles of people involved in processes; only generic titles are used to refer to personnel profiles (such as acquirer, developer, user, etc.).

3.2 Software Development Methodologies

There are many well known methodologies in professional software development. In this analysis, we have focused on two of the most widely used methodologies in Spain: the Unified Software Development Process [20] and Métrica 3 [21].

Although strictly speaking this is not a methodology, USDP (Unified Software Development Process) is closely related to the application of UML (Unified Modeling Language) notation [22]. It describes a use-case driven, architecture-centric, iterative, and incremental process. For each phase[7] identified in the process, there is a set of activities and responsibilities that workers involved in the process should cover. In USDP, ten different roles are established:

System Analyst, Use-Case Specifier, User-Interface Designer, Architect, Use-Case Engineer, Component Engineer, System Integrator, Test Designer (Test Engineering), Integration Tester, and System Tester.

The responsibilities of these roles evolve throughout the process, so a team member may take on different roles during a project. The project leader should know individual competences so that the best suited people are assigned to their appropriate roles.

Métrica v3 is the official Spanish methodology[8] for national public administration contracts. It provides a framework for the systematic performance of activities involved in SLC. Métrica consider that SLC participants can be broken down into five generic professional profiles (Manager, Project Leader, Consultant, Analyst, Programmer). The description of each profile includes responsibilities and duties for each phase of SLC[9] and details the tasks associated with them. Table 3 shows the relationship between participants in the development process and the profiles identified in Métrica 3[10].

If we concentrate on the main processes for IS development (feasibility study, analysis, design, construction, implementation and acceptance, maintenance) and on the main tasks associated with these processes, the participant catalogue could be reduced to the first twenty roles/positions.

Similarities can be seen between the two methodologies/process proposals: although they do not include exactly the same tasks, the development processes identified in [21] could be deemed to be more or less equivalent to the phases described in [20] (see Table 4).

We can therefore compare workers' tasks in the two methodologies, according to the phases in which they take part. The results are shown in Table 5.

In terms of the processes in which the various categories participate, the similarity between the profiles of the two methods is quite marked in certain cases (Architect from USDP with Analyst in Métrica 3, System integrator from USDP with Programmer from Métrica 3). But there are also cases in which no direct match is possible, e.g., Métrica 3 Implementation team could be associated with any of three USDP workers (Test Designer - Test Engineering, Integration Tester, System Tester). And, of course, there are

[8] Documentation is available, in Spanish, at the Ministry of Public Administration's website:<http:// www.map.es/csi>.

[9] Each phase description includes participants and their main characteristics.

[10] In this analysis we do not include any profile for the position of executive manager, normally associated with company responsibilities and decision making.

ID	Development participants	Project Leader	Consultant	Analyst	Programmer
1	Project leader	J			
2	Implementation manager	J			
3	Maintenance manager	J			
4	Operations manager	J			
5	Systems manager	J			
6	Security manager	J			
7	Communications specialist		C		
8	Systems technician		C		
9	Communications technician		C		
10	Analyst			A	
11	Database administrator			A	
12	Project team			A	
13	Architectural team			A	
14	Implementation team			A	
15	Operations team			A	
16	Systems team			A	
17	Technical support team			A	
18	Training and education team			A	
19	Programmer				P
20	Systems operations team				
21	Software quality manager	J			
22	Consultant		C		
23	Computing consultant		C		
24	IT Consultant		C		
25	IS Consultant		C		
26	Software Quality Assurance Group			A	

Table 3: Métrica 3 Participants and Profiles.

Métrica 3	USDP
Feasibility study	Requirements capture
System analysis	Analysis
System design	Design
System construction	Implementation
System acceptation and implementation	Test

Table 4: Phases in Métrica 3 and USDP.

many profiles with no equivalent profile on the other side.

3.3 Professionalism: Job Classification and Certification

The International Standard Classification of Occupations [23] is a general set of occupational definitions for all workforce occupations. ISCO classifies jobs in terms of the type of work to be performed. The basic criteria used to define the groups used by the system are the levels of "skill" and "skill specialization" required to carry out the tasks and duties of the jobs.

Table 6 presents this classification scheme together with the job titles related to the classification group for each IT job. Our research focused on job titles involving software development (other computing positions are shown in grey).

The Standard Occupational Classification SOC 2000 [24] is a general set of occupational definitions for all workforce occupations. It was first published in 1990 (SOC 90) and has been revised and updated up to the present SOC2000. It describes typical entry routes and associated qualifications for each group and proposes a list of related job titles. Table 7 shows a summary of the group connected to IT. As we are concerned only with software development, once again there are a number of groups or titles that are not relevant to this article.

Although not linked to any official standardization organization, Career Space [25] has earned its status as a European benchmark for the IT profession. It is a consortium of major ICT companies working in partnership with the European Commission that has developed generic skills profiles relevant to key ICT jobs that cover the main job areas for which the ICT industry is experiencing skills shortages. Each core profile describes the jobs involved, setting out the vision, role, and lifestyle associated with them as well as the specific technology areas and tasks associated with each job and the level of behavioural and technical skills required to carry out the profiled jobs. Three of these profiles relate closely to the Software Development process & Test Engineering). In Table 8 there is a list of 53 associated job titles[11].

Another interesting slant on job descriptions is to be found in certification schemes. EUCIP (European Certification of Informatics Professionals) [26] is a qualification scheme for IT professionals that arose out of a CEPIS (the Council of European Professional Informatics Societies) initiative. One of the goals of this project is to define an "*industry-driven vocational structure and standards for the informatics profession*". There are 22 professional certifications planned but only four of them are available as yet (see those in bold type in Table 9).

If we compare Tables 6 to 9 there are many similarities to be found but, generally speaking, there are a large number of different classifications profiles and job classifications that are not easy to match.

Met.	Id.	Participants					
USDP	1	System Analyst	x				
USDP	2	Use-Case Specifier	x				
USDP	3	User-Interface Designer	x				
USDP	4	Architect	x	x	X	x	
USDP	5	Use-Case Engineer		x	X		
USDP	6	Component Engineer		x	X	x	X
USDP	7	System Integrator				x	
USDP	8	Test Designer (Test Engineering)					X
USDP	9	Integration Tester					X
USDP	10	System Tester					X
Metr. 3	1	Project leader	x	x	X	x	X
Metr. 3	2	Implementation manager					X
Metr. 3	3	Maintenance manager	x				X
Metr. 3	4	Operations manager			x		X
Metr. 3	5	Systems manager			x		X
Metr. 3	6	Security manager	x				
Metr. 3	7	Communications specialist	x				
Metr. 3	8	System technician	x			x	
Metr. 3	9	Communications technician				x	
Metr. 3	10	Analyst	x	x	x	x	
Metr. 3	11	Data base administrator			x	x	X
Metr. 3	12	Project team			x	x	
Metr. 3	13	Architectural team		x	x	x	
Metr. 3	14	Implementation team					X
Metr. 3	15	Operations team				x	X
Metr. 3	16	Systems team		x	x	x	
Metr. 3	17	Technical support team	x	x	x		X
Metr. 3	18	Training and education team				x	X
Metr. 3	19	Programmer				x	
Metr. 3	20	System operations team			x		

Table 5: Participants in Each Phase: Métrica 3 and USDP.

[11] Job titles related to Data Communications Engineering, Multimedia Design, ICT Management or IT Business Consultancy are not included.

Job titles	Classification
Analyst, communications/computers	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Analyst, database/computers	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Analyst, systems/computers	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Database administrator	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Designer, systems/computers	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Engineer, computer systems	COMPUTER SYSTEMS DESIGNERS AND ANALYSTS
Programmer	COMPUTER PROGRAMMERS
Programmer, communications	COMPUTER PROGRAMMERS
Programmer, database	COMPUTER PROGRAMMERS
Engineer, computer applications	COMPUTING PROFESSIONALS NOT ELSEWHERE CLASSIFIED
Assistant, computer/programming	COMPUTER ASSISTANTS
Assistant, computer/systems analysis	COMPUTER ASSISTANTS
Assistant, computer/users' services	COMPUTER ASSISTANTS
Operator, computer peripheral equipment	COMPUTER EQUIPMENT OPERATORS
Operator, computer peripheral equipment/high-speed printer	COMPUTER EQUIPMENT OPERATORS

Table 6: ISCO Job Titles.

4 Conclusion

This initial study provides us with a general overview of the huge variety of job titles used to define people working in software development. It would be interesting to have a more detailed breakdown of these job titles under the umbrella of a formal standard classification for professional profiles. Better still, it would take into account up to date information from the software industry (supply and demand of IT professionals) in order to stay in touch with present day requirements and descriptions of profiles, roles, and job titles. By using mechanisms such as the RENTIC[12] reports [27], we are engaged in a line of research to help determine the specific requirements for each position.

The RENTIC database includes data from more than 1,200 job offers and covers 187 different job titles. From this data, we have determined that, for example, the most important personal skills for analysts[13] are, ranked in order of frequency of demand: teamwork, initiative/proactivity, and autonomy/independence. Academia is interested in obtaining an insight into the labour market outlook in order to offer degree or specialized courses for each specific area. Researchers at the *Université du Québec à Montréal* (Canada), ACM and the IEEE Computer Society have been working on identifying the Software Engineering elements that are most widely accepted by both academia and industry.

This project, called SWEBOK(Software Engineering Body of Knowledge [28] was initiated in 1998, and in 2004 the IEEE Computer Society approved the 2004 edition of its guide to the Software Engineering Body of Knowledge. The outcome of this research has been used in Computing Curricula 2004[14] [29] and, naturally, in SE 2004 [30] (Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering).

There is a similar initiative in Europe [31] that analyses the situation of ICT Practitioner Skills/Competence Frameworks in an effort to create a European ICT Skills Meta-Framework for the proposed European Qualifications Framework (EQF). This generic research does not review specific software development environment or software development methodologies and standards, so it may be interesting as a basis for further analysis.

In any event, as McConnell [32] said, "*the benefits of creating a true profession of Software Engineering are compelling*", so it is important to promote a clear definition of the profiles involved in the software development process.

[12] Specific reports focused on requirements for each job offer in IT in the Spanish labour market carried out by Dr. L. Fernandez-Sanz since 1998.

[13] Including equivalent titles.

[14] A Guide to Undergraduate Degree Programs in Computing, still in progress, describing five undergraduate degree programs in Computer Engineering: Computer Science, Information Systems, Information Technology, and Software Engineering.

1136	**Information and communication technology managers**
	Computer manager
	Computer operations manager
	Data processing manager
	IT manager
	Systems manager
	Telecom manager
2131	**IT strategy and planning professionals**
	Computer consultant
	Software consultant
2132	**Software professionals**
	Analyst-programmer
	Computer programmer
	Software engineer
	Systems analyst
	Systems designer
3131	**IT operations technicians**
	Computer operator
	Database manager
	IT technician
	Network technician
	Systems administrator
	Webmaster
3132	**IT user support technicians**
	Help desk operator
	Helpline operator (computing)
	IT helpline support officer
	Support technician (computing)
	Systems support officer
5245	**Computer engineers, installation and maintenance**
	Computer engineer
	Computer maintenance engineer
	Computer service engineer
	Computer service technician

Table 7. SOC 2000 Unit Groups and Job Titles Relevant to The Software Development Process.

References

[1] N.E. Fenton. Software metrics: a rigorous and practical approach, PWS Thomson, 1997.

[2] L. Fernández, and M.J. García. "Software Engineering Professionalism", Upgrade, Vol. IV, num. 4, 42-46, 2003.

[3] L. Fernández. "Requisitos para el empleo en Nuevas Tecnologías de la Información: el estudio RENTIC". Novática, num. 161, 51-56, 2003.

[4] W.S. Humphrey. Introduction to the personal software process, Addison-Wesley, 1995.

[5] R. Groth. Is the Software Industry's Productivity Declining?, IEEE Software, 92-94, November/December, 2004.

[6] B.W. Boehm. Software engineering economics, Prentice-Hall, 1981.

[7] B. Boehm et al. Software cost estimation with COCOMO II, Prentice-Hall, 2000.

[8] C. Jones. Estimating software costs, McGraw-Hill, 1998.

[9] T. Abdel-Hamid. The Slippery Path to Productivity Improvement, IEEE Software, July, 43-52, 1996.

[10] I. Steiner. Group process and productivity, Academic Press, 1972.

[11] G.M. McCue. IBM's Santa Teresa Laboratory - Architectural Design for Program Development, IBM Systems Journal, vol 17, num.1, 4-25, 1978.

[12] D.E. Perry, N.A. Staudemayer, and L.G. Votta. "People, Organizations, and Process Improvement", IEEE Software July 1994: 36-45.

[13] A.H. Dutoit, B. Bruegge. Communication Metrics for Software Development, IEEE Transactions On Software Engineering, Vol. 24, num. 8, 615-628, 1998.

[14] D.E. Perry, N.A. Staudemayer, and L.G. Votta. People, Organizations, and Process Improvement, IEEE Software, July, 36-45, 1994.

[15] T. DeMarco, and T. Lister. Peopleware: Productive Projects and Teams, Dorset House, New York, 1987.

[16] G.M. Weinberg. The Psychology of Computer Programming, Van Nostrand Reinhold Computer, 1971.

[17] ISO/IEC 15504:1999 Information technology — Process assessment. International Standards Organization,1999.

[18] ISO/IEC 12207:1995 Information technology. Software life cycle processes, International Standards Organization, 1995.

[19] IEEE Std 1074-1997. IEEE Standard for Developing Software Life Cycle Processes. IEEE Computer Society, 1997.

[20] I. Jacobson, G. Booch, and J. Rumbaugh. The Unified Software Development Process, Addison-Wesley, 1999.

[21] Ministerio de Administraciones Públicas. Métrica3, <http://www.csi.map.es/csi/metrica3/index.html>, 2003.

[22] G. Booch, J. Rumbaugh, and I. Jacobson. The unified modeling language user guide, Addison-Wesley, 1999.

[23] International Labour Organization: International Standard Classification of Occupations (ISCO 88), <http://www.ilo.org/public/english/bureau/stat/class/isco.htm>.

[24] Office of National Statistics (ONS) UK: Standard Occupational Classification SOC 2000, <http://www.statistics.gov.uk/methods_quality/ns_sec/soc2000,asp>.

[25] New ICT curricula for the 21st century: Designing Tomorrow's. Education. Office for Official Publications of the European Communities, 2001. (Available at <http://www.career-space.com>.)

[26] Eucip: European Certification of Informatics Professionals. <http://www.eucip.com/index.htm>.

[27] L. Fernández. Requisitos para el empleo en Nuevas Tecnologías de la Información: el estudio RENTIC, Novática, nº161, enero-febrero 2003, pp. 51-56.

ID	Job title	Job Profile
1	Application Programmer	Software & Applications Development
2	Software (SW) Engineer	Software & Applications Development
3	System Developer	Software & Applications Development
4	Technical System designer	Software & Applications Development
5	SW Architect	Software & Applications Development
6	Maintenance & Support Specialist	Software & Applications Development
7	Integration Technician	Software & Applications Development
8	Software Programmer	Software Architecture and Design
9	Systems Developer	Software Architecture and Design
10	Systems Architect	Software Architecture and Design
11	Systems Architecture & Design Scientist	Software Architecture and Design
12	Systems Integrator	Software Architecture and Design
13	Network Designer	Software Architecture and Design
14	Computer Scientist	Software Architecture and Design
15	Systems Integrator	Integration & Test/ Implementation & Test Engineering
16	System Implementation Engineer	Integration & Test/ Implementation & Test Engineering
17	Integration System Engineer	Integration & Test/ Implementation & Test Engineering
18	Integration Engineer	Integration & Test/ Implementation & Test Engineering
19	Implementation and Test Specialist	Integration & Test/ Implementation & Test Engineering
20	Integration and Test Specialist	Integration & Test/ Implementation & Test Engineering

Table 8: Job Profiles and Titles Included in Career Space.

[28] SWEBOK. <http://www.swebok.org>. Université du Québec à Montréal, ACM and the IEEE Computer Society.

[29] Computing Curricula 2004. Joint Task Force Guide on the main Computing disciplines (for university Degree courses). IEEE-CS + ACM + AIS (US). <http://www.acm.org/education/Overview_Draft_11-22-04.pdf>.

[30] SE 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. <http://sites.computer.org/ccse/SE2004Volume.pdf>.

[31] CEN Workshop Agreement. European ICT Skills Meta-Framework. Final version, For entering the 60 days public comment period and formal approval by the Workshop members. November 2005.

[32] S. McConnell. Professional Software Development, Addison Wesley, 2003.

	Title	Knowledge Area
1	IS Manager	PLAN
2	IS Quality Auditor	PLAN
3	Enterprise Solutions Consultant	PLAN
4	**Business Analyst**	PLAN
5	Logistics & Automation Consultant	PLAN
6	Sales and Application Consultant	PLAN
7	Client Services Manager	PLAN
8	IS Project Manager	PLAN
9	IT Systems Architect	BUILD
10	**Information Systems Analyst**	BUILD / PLAN
11	Web & Multimedia Master	BUILD
12	Systems Integration & Testing Engineer	BUILD
13	**Software Developer**	BUILD
14	Database Manager	BUILD
15	X-Systems Technician	OPERATE
16	Telecommunications Engineer	OPERATE
17	Network Architect	OPERATE
18	Security Adviser	OPERATE
19	**Network Manager**	OPERATE
20	Configuration Manager	OPERATE
21	Help Desk Engineer	OPERATE
22	IT Trainer	OPERATE

Table 9: EUCIP Professional Certifications.

