Balancing software engineering education and industrial needs

Ana M. Moreno[a], Maria-Isabel Sanchez-Segura[b], Fuensanta Medina-Dominguez[b,*], Laura Carvajal[a]

[a] *School of Computing, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain*

[b] *Computer Science Department, Carlos III Technical University of Madrid, Avda. De la Universidad, 30, 28911 Leganés, Madrid, Spain*

ABSTRACT

In the world of information and communications technologies the demand for professionals with software engineering skills grows at an exponential rate. On this ground, we have conducted a study to help both academia and the software industry form a picture of the relationship between the competences of recent graduates of undergraduate and graduate software engineering programmes and the tasks that these professionals are to perform as part of their jobs in industry. Thanks to this study, academia will be able to observe which skills demanded by industry the software engineering curricula do or do not cater for, and industry will be able to ascertain which tasks a recent software engineering programme graduate is well qualified to perform. The study focuses on the software engineering knowledge guidelines provided in SE2004 and GSwE2009, and the job profiles identified by Career Space.

1. Introduction

The US Department of Labor foresees that employment demand for IT-related professionals will grow much faster than for average occupations through the year 2016. The Bureau of Labor Statistics predicts an increment of 32.45% for computer software engineers in the US from 2008 to 2018 (Bureau, 2008a, 2008b). On the other hand, it is also estimated that the US will have half of the qualified IT professionals needed through to 2012 (Longo, 2006). Those data, although specific for US, reflect a situation easily extrapolable to other parts of the world.

In this context, how to educate software engineers to do their jobs efficiently and properly remains a crucial open question for the future of the profession having in mind the relevance of software engineering (SE) practices in the IT world.

Several researchers have already highlighted the existing gap between software education and industry needs (Lee and Han, 2008; Kim et al., 2006; Aasheim et al., 2009). Additionally, some empirical studies have addressed this issue. One of the first was the report published in 2000 by Lethbridge analysing the relevance and depth of the knowledge that a representative set of US and Canadian software professionals had received as part of their graduate education. The study identified a significant mismatch between software education and industry in terms of the knowledge needed by software engineers to perform the tasks required by industry. More recent studies, including surveys by Kitchenham et al. (2005) or Surakka (2007), again identified such a fissure.

Aligning SE education with industry is a sizeable challenge. Lethbridge et al. (2007) take the view that filling this gap is one of the most critical tasks to be addressed in SE education. In their opinion, the task is complicated by several open questions, including "What industrial practices are currently not being taught? How effective are these practices? Which practices should be taught to undergraduates?"

Nowadays, different approaches are being taken in an attempt to answer the above questions. Some higher education institutions are working with panels of experts composed of industry and education representatives whose goal is to assess the definition of their SE graduate or undergraduate programmes. Complementary international *de facto* standards for SE education, like SE2004 (Software Engineering, 2004) Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE, 2004) or GSwE2009 (Graduate Software Engineering 2009—GSwE, 2009), have been developed with the relevance of industrial needs in mind.

Our aim is to contribute to this debate. To effectively fill this gap, it would be necessary, on one hand, to guarantee that the educational programmes provide the knowledge required for the job profiles suggested by industry and also assure that this knowledge is taught in a manner enabling future professionals to correctly tackle the problems that they will face during their professional career (Loftus et al., 2011). In this paper we will focus on the first point and analyse whether SE educational standards set out

* Corresponding author. Tel.: +34 91 624 91 06; fax: +34 91 624 91 29.
E-mail addresses: ammoreno@fi.upm.es (A.M. Moreno), misanche@inf.uc3m.es (M.-I. Sanchez-Segura), fmedina@inf.uc3m.es (F. Medina-Dominguez), laura.carvajal@upm.es (L. Carvajal).

the knowledge to be taught to satisfy the job profiles proposed by industry. In other words, we will explore the possible breach between the skills that industry demands for software engineers and the international recommendations for SE undergraduate and graduate education, SE2004 and GSwE2009, respectively.

The results of this study are useful on two fronts. On one hand, they can be used by individual educators to develop new undergraduate or graduate SE programmes based on the international standards SE2004 or GSwE2009. Using our results they can identify which industrial requirements are not completely addressed by the above standards and, therefore, need to be reinforced, for example, by means of elective or complementary courses. On the other hand, industry can also benefit from this study by identifying possible skill gaps among SE graduates. This information is potentially useful for assigning roles and tasks to employees, as well as for planning industrial training.

This paper is structured as follows. Section 2 discusses the related work in this area, whereas Sections 3–5 detail the sources that we will use to analyse the gap between academia and industry, namely, SE2004 and GSwE2009 from the academic point of view, and the information and communications technology (ICT) skills identified by the Career Space consortium on the industry side. In Section 6, we examine whether the SE2004 and GSwE2009 cores provide knowledge that is useful for performing each Career Space task related to each of the three software development-related profiles: Software and Applications Development, Software Architecture and Design and IT Business Consultancy. Finally, Section 7 describes the conclusions obtained of our analysis.

2. Background

There are a number of different studies that suggest that there is a gap between skills acquired by information technology (IT) graduates and skills demanded by industry. For example, Lee and Han (2008) analysed the IS 2002 model curriculum from the perspective of industry, and suggested some skills, like problem solving or knowledge of business, that should be included in the design of future IS curricula. Kim et al. (2006) also analysed the content of IS curricula sampling employees at one manufacturing firm. They found that project management was the highest ranked skill and that topics like security or end-user computing should be given more emphasis in IS curricula. Aasheim et al. (2009) studied whether academia had a different perception of the importance of various skills for entry-level IT workers than IT managers. They found that IT managers place more importance than faculty on issues related to hardware concepts, operating systems, leadership skills or entrepreneurial traits. However, both groups ranked broader categories of skills—interpersonal skills, personal skills, technical skills, organizational skills and work experience—in the same order of importance.

Lethbridge (Lethbridge, 2000) conducted the seminal study in this field on the specialized area of software development. He worked with professionals from Canada and the United States to identify the gaps between the knowledge required from a practice viewpoint and the education and training received (Lethbridge, 2000). More recently, Kitchenham et al. adapted the surveys used by Lethbridge and ran a similar study with recent SE graduates from several UK universities (Kitchenham et al., 2005). Table 1 lists the knowledge for which the gap between higher education received and importance in industry is widest in the two surveys. Kitchenham et al. point out that a possible source of the differences in the two surveys is the surveyed population. Lethbridge's survey deals with professionals with experience in industry, whereas Kitchenham interviewed recent graduates. On the other hand, the issue of web development was not as relevant in 2000 as it has been in the last decade. This may explain why this issue ranks top in Kitchenham et al.'s study.

Table 1
Comparison of the knowledge gap between studies reported in Lethbridge (2000) and Kitchenham et al. (2005).

Lethbridge (2000)	Kitchenham et al. (2005)
HCI/user interfaces	Web-based programming
Real-time system design	Project management
Software cost estimation	Configuration and release management
Software metrics	Multimedia
Software reliability and fault tolerance	Security and cryptography
Requirements gathering and analysis	Computer graphics

In any case both studies conclude that mathematical topics appear to be taught in more depth than required in industrial practice. This result should be taken to mean not that this knowledge is not useful but that it should be delivered with a more practical slant. This would help students to perceive its relevance for logical reasoning and hence software practice. On the other hand, general business topics, like management, leadership, negotiation or delivery of presentations, are quite important and should be given more emphasis by educators.

A third study in this category was performed by Surakka (Surakka, 2007) in the Finnish context. He surveyed software developers, professors and lecturers, and master students about the relevance of different matters. His results coincide with Lethbridge's and Kitchenham's outcomes regarding the excessive importance attached to mathematics-related topics in academia, and with Kitchenham's findings as to the relevance of web-related areas in industry. SE-specific issues like design, testing or configuration management appear to be considered as equally relevant by all three surveyed groups.

Complementarily, a number of studies have been conducted to define the critical skills that graduates need to perform IT-related jobs properly. Abraham et al. (2006) worked with non-IT US companies finding that skills associated with business were crucial as such skills were more likely to be retained in-house. Technical skills, on the other hand, were what they were looking for in new recruits.

Wu et al. (2007) investigate critical IS professional activities, as well as knowledge requirements for three management levels (*i.e.*, supervisory, middle and top) at two kinds of Taiwan companies (manufacturing and services). They identified 20 IS management activities and found that the importance of such professional activities was viewed in significantly different ways by each level of IS management; but this was unchanged in the two types of industry. From the computer programmer's perspective, Bailey and Stefaniak (2001) surveyed IT professionals at US companies who identified a total of 85 skills (53 technical, 20 soft and 12 business concepts) that were relevant for this role. Of them, technical and soft skills are viewed as the most relevant for IT professionals.

The analysis of keywords in Internet job postings has been also used to determine the critical skill requirements of IT professionals. Koong et al. worked with Monster.com and HotJob.com and split the different jobs into seven categories in which programming languages and web development accounted for about 51% of the identified job skills (Koong et al., 2002). Prabhakar et al. (2005) worked with Monster.com over a three-year period and again highlighted the relevance of web-related skills. Kovacs and Davis (2008) analysed keywords from Internet job postings of a US regional trade association. They agreed with the previous studies in identifying web and web-related applications skills as the keywords in highest demand. Although they do not rank highest in this study; keywords related to communications and project management are among the top ten. This result matches Gallavin et al.'s findings (2004); where most skills listed in job advertisements were technical. We found this situation somewhat inconsistent with organizational claims that soft skills are important for entry-level professionals.

Another report that tries to identify useful knowledge for the software industry was generated by the Gartner Group (Gartner, 2005). It highlights four kinds of knowledge that will be crucial for future IT professionals:

- Technical knowledge. How does technology work? What are its effects? How does it interact with other technologies? What are its dependencies?
- Business-specific knowledge. What makes this company tick? Business-specific knowledge breaks down further into knowledge of enterprise objectives, operational activities, social and knowledge networks, and cultural behaviour.
- Core process knowledge. What processes fuel this company's competitive edge? In other words, which processes make this company unique?
- Industry knowledge. What forces, markets and models characterize this industry? Which parties or industries are traditional or emerging buyers and sellers? How does regulation affect this industry? Which industries does this industry resemble?"

The relevance of non-technical knowledge leads to an interesting thought that Davey and Tatnall (2008) sums up as the need to move from a focus on data to a focus on the business.

Another interesting study that details the skills needed by IT professionals to perform their jobs was conducted by the Career Space industrial consortium (Career Space, 2001). The European Union put in place this project designed to foresee the roles, skills and competencies required by the ICT industry in Europe in the near future. This initiative was motivated by the shortage of skilled ICT people in Europe, that is, "people with a fluent digital age language; people skilled in developing and deploying new technologies; people who are experts in communication with business, professionals and customers; people who are comfortable with technology yet able to understand others needs and be their guide and counsel". Even though this European study was conducted a decade ago, the situation has not improved significantly in either Europe or elsewhere in the world. For example, Lethbridge et al. (2007) also draw attention to this shortage of software professionals indicating that only 40% of computer industry workers actually have a computing-related education. Most are not educated in key portions of the SE body of knowledge, such as requirements, architecture, testing, human factors and project management. Most practitioners are merely skilled at programming in a few popular languages or at using specific technology products, such as database management and web development tools.

In this context, our research focuses on a complementary view of the relationship between SE education and industrial needs. As already mentioned, we will try to provide detailed information about how *de facto* SE education standards contribute to the provision of the knowledge needed to perform software tasks identified by industry. To do this, we will use the Career Space report as a source of the industrial needs to be addressed on the following grounds:

- Although it was developed a decade ago, it was developed with a visionary perspective of IT, which is still valid nowadays. This visionary perspective is able to cover emergent profiles or tasks, like particular global skills, as we will discuss later, or recent development methods, like agile approaches.
- The generic profiles provided by Career Space fit one of the most used occupational classifications, the United Kingdom's Standard Occupational Classification (SOC, 2000).
- As far as we know, there is no other recent publication that gives an overall picture of software engineering profiles.
- The roles and tasks defined by Career Space are detailed enough to be related to the core knowledge recommendations provided by standard educational guidelines.

3. Software engineering undergraduate curriculum guidelines

The 2004 Software Engineering Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering were instituted with the aim of "providing guidance to academic institutions and accreditation agencies about what should constitute an undergraduate software engineering education" (SE, 2004).

SE2004 was built around three major initiatives that engaged a large number of volunteers, as well as all the members of the Steering Committee. The first of these initiatives was the development of a set of target curriculum outcomes and a statement of what every SE graduate should know. The second initiative involved determining and specifying the knowledge to be included in an undergraduate SE programme. The third initiative was the construction of a set of curriculum recommendations, describing how a software engineering curriculum, incorporating SEEK, could be structured in various contexts (SE, 2004).

In actual fact, they defined what was termed Software Engineering Education Knowledge (SEEK). SEEK represents a body of core knowledge or essential material that professionals teaching SE agree is necessary for anyone to obtain an undergraduate degree in this field. So, even though SEEK does not represent a complete curriculum, it does provide the foundation for the design, implementation, and delivery of core educational units that make up a software engineering curriculum.

The body of SEEK is organized hierarchically into three levels. Knowledge areas (KA) represent particular subdisciplines of SE that are generally recognized as significant parts of the body of SE knowledge that an undergraduate should learn. Each area is broken down into smaller divisions called units, which represent individual thematic modules within an area. Finally, each unit is further subdivided into a set of topics, which are the lowest level of the hierarchy. The contents can be packaged into different course names, generating specific curricula that cover the same core of SE knowledge.

Table 2 sums up the SEEK knowledge areas. The specific topics to be covered in each unit are outlined in SE (2004). Table 2 also shows the percentage of in-class time a student should spend on each KA, defined in terms of Bloom's taxonomy (Knowledge, K, Comprehension, C and Application, AP; Bloom et al., 1956). Bloom's taxonomy is a classification of learning objectives within education proposed in 1956 by a committee of educators chaired by Benjamin Bloom. It refers to a classification of the different objectives that educators set for students (learning objectives). SE2004 works with Bloom's taxonomy at topic level. In this paper, we have abstracted these values to unit level by generalizing the rates for the topics of each unit. SE 2004 works with what are termed contact hours, that is, the minimum number of hours of in-class time needed to present the material to students. For the purpose of comparison with GSwE, we have translated contact hours to percentage of in-class time, as shown in Table 2.

SE2004 also suggests a set of 10 specialization areas, like network-centric systems, information systems and data processing, financial and e-commerce systems, etc. Those areas suggest

Table 2
SEEK knowledge areas and knowledge units.

SEEK knowledge area	SEEK units	Bloom's taxonomy	Percentage of in-class time/unit	Percentage of in-class time/area
Computing Essentials	Computer science foundations	C/AP	28%	35%
	Construction technologies	C/AP	4%	
	Tools	C/AP	1%	
	Formal construction methods	K/C	2%	
Mathematical and Engineering Fundamentals	Mathematical foundations	C/AP	11%	18%
	Engineering foundations for software	K/C	5%	
	Engineering economics for software	K/C	2%	
Professional Practice	Group dynamics/psychology	K/C	1%	7%
	Communication skills	AP	2%	
	Professionalism	K/C	4%	
Software Modelling and Analysis	Modelling foundations	K/C	4%	11%
	Types of models	C/AP	2%	
	Analysis fundamentals	C/AP	1%	
	Requirements fundamentals	K/C	1%	
	Eliciting requirements	C	1%	
	Requirements specification & documentation	K/AP	1%	
	Requirement validation	C/AP	1%	
Software Design	Design concepts	C/AP	1%	9%
	Design strategies	C/AP	1%	
	Architectural design	K/AP	2%	
	Human-computer interface design methods	C/AP	2%	
	Detailed design	C/AP	2%	
	Design support tools and evaluation	K/AP	1%	
Software Verification and Validation	V&V terminology and foundations	K/AP	1%	9%
	Reviews	AP	1%	
	Testing	AP/C	4%	
	Human-computer UI testing and evaluation	C/AP	1%	
	Problem analysis and reporting	C/AP	1%	
Software Evolution	Evolution process	K	1%	2%
	Evolution activities	K	1%	
Software Process	Process concepts	K/C	1%	3%
	Process implementation		2%	
Software Quality	Software quality concepts and culture	K	0.5%	3%
	Software quality standards	K	1%	
	Software quality process	K/C	1%	
	Process assurance	K/C	0.5%	
Software Management	Management concepts	K	0.5%	4%
	Project planning	C/AP	1%	
	Project personnel and organization	K	1%	
	Project control	K/C	0.5%	
	Software configuration	K/C	1%	

elective computing knowledge that students taking those specialities should learn.

4. Graduate software engineering curriculum

The Curriculum Guidelines for Graduate Degree Programs in Software Engineering (GSwE2009) were developed with the aim of enabling the creation of new and improving existing professional master programmes in SE. The project was started up in 2007 by the Stevens Institute of Technology and sponsored by the US Department of Defense. To date, both IEEE Computer Society and ACM support the initiative, and about 100 experts from industry, government and academia have participated in the project.

One of the main outputs of GSwE2009 is the Core Body of Knowledge (CBOK) to be covered by a professional SE master programme. This knowledge is designed to cover about 50% of the master programme contents. The reason behind this decision is to provide a flexible framework for the development of graduate SE programmes. The other 50% should be covered by detailing the CBOK contents or focusing on a chosen application domain.

Like SE2004, CBOK is organized around knowledge areas, which are decomposed into units and further into topics. Table 3 shows knowledge areas and units, along with the in-class time for each KA as a percentage of the entire programme, as well as the recommended level to which a student should learn each KA according to Bloom's taxonomy. GSwE2009 uses the Comprehension (C),

Table 3
CBOK knowledge areas and units.

CBOK knowledge area	CBOK units	Bloom's taxonomy	Percentage of contact hours
A. Ethics and Prof. Conduct	1. Social, legal, and historical issues	C	1–2%
	2. Codes of ethics and professional conduct	C/AP	
	3. The nature and role of SE standards	C	
B. Systems Engineering	1. Systems Engineering Concepts	C	2–3%
	2. Systems Engineering Life Cycle Mgmt.	C	
	3. Requirements	C/AP	
	4. System Design	C/AP	
	5. Integration and Verification	C	
	6. Transition and Validation	C	
	7. Operation, Maintenance and Support	C	
C. Requirements Engineering	1. Fundamentals of Requirements Engineering	C/AP	6–8%
	2. Requirements Engineering Process	C	
	3. Initiation and Scope Definition	AP	
	4. Requirements Elicitation	AP	
	5. Requirements Analysis	AN	
	6. Requirements Specification	AP	
	7. Requirements Validation	AP	
	8. Practical Considerations	C/AP	
D. Software Design	1. Software Design Fundamentals	C/AP	9–11%
	2. Key Issues in Software Design	AP	
	3. Software Structure and Architecture	AP/AN	
	4. Sw. Design Quality Analysis and Evaluation	AP	
	5. Software Design Notations	AP	
	6. Software Design Strategies and Methods	AP/AN	
E. Software Construction	1. Software Construction Fundamentals	AP	1–3%
	2. Managing Construction	AP	
	3. Practical Considerations	AP	
F. Testing	1. Testing Fundamentals	AP	4–6%
	2. Test Levels	AP	
	3. Testing Techniques	AP	
	4. Test-Related Measures	AP/AN	
	5. Test Process	C/AP	
G. Software Maintenance	1. Software Maintenance Fundamentals	C	3–4%
	2. Key Issues in Software Maintenance	AP	
	3. Maintenance Process	AP	
	4. Techniques for Maintenance	AP	
H. Configuration Management	1. Management of the CM Process	C/AP	2–3%
	2. Configuration Identification	AP	
	3. Configuration Control	AP	
	4. Configuration Status Accounting	AP	
	5. Software release Management and Delivery	AP	
I. SE Management	1. Software Project Planning	AP	7–8%
	2. Risk Management	AP	
	3. Sw. Project Organization and Enactment	AP	
	4. Review and Evaluation	C	
	5. Closure	C	
	6. Software Engineering Measurement	AP	
	7. Engineering Economics	C	
J. SE Process	1. Process Implementation and Change	C/AP	3–4%
	2. Process Definition	C	
	3. Process Assessment	AP	
	4. Product and Process Measurement	AP	
K. Software Quality	1. Software Quality Fundamentals	AP	3–4%
	2. Software Quality Management Processes	AP	
	3. Verification and Validation (V&V)	AP	

Application (AP) and Analysis (AN) levels. Notice that, although the Analysis level applies to only a few units, the highest level in undergraduate recommendations is AP. GSwE2009 also differs from SE2004 as to the recommended contact hours for each area and unit. GSwE2009 works with percentages of hours for each area, which apply only to the core of the programme. The programme core represents approximately 50% of the curriculum. In this context, the percentages are set to be considered as general high-level guidance, not as a precise curriculum specification.

As in SE2004, the KAs outlined in Table 3 are intended to characterize the core content of a master's programme in SE; they are not intended to depict or to imply the organization of curricula and courses. Actually this CBOK can be packaged as different master programmes with different structures and course names.

5. Career space skills profile

Career Space was a European Union initiative in which a consortium of eleven major ICT companies (BT, Cisco Systems, IBM Europe, Intel, Microsoft Europe, Nokia, Nortel Networks, Philips Semiconductors, Siemens AG, Telefonica S.A., Thales), and the European Information and Communications Technology Industry Association

Table 4
Generic job profiles for Career Space.

Telecommunications	Radio Frequency Engineering
	Digital Design
	Data Communications Engineering
	Digital Signal Processing
	Applications Design
	Communications Network Design
Software & Services	Software & Applications Development
	Software Architecture and Design
	Multimedia Design
	IT Business Consultancy
	Technical Support
Products & Systems	Product Design
	Integration & Test/Implementation & Test Engineering
	Systems Specialist
Cross Sector	ICT Marketing Management
	ICT Project Management
	Research and Technology Development
	ICT Management
	ICT Sales Management

(EICTA), developed several initiatives to address the shortage of well-trained IT professionals in Europe. One of their responsibilities was to define the skills and competencies required by the European ICT industry in the near future. To do this, the consortium outlined a set of core profiles describing examples of job titles, setting out the vision, role and professional lifestyle associated with each job profile. The specific technology areas and tasks associated with each job were also outlined, as was the level of behavioural and technical skills required to do the job. Table 4 shows the identified job profiles grouped into four categories: telecommunications, software and services, products and systems and cross sector.

Software & Services, and particularly Software & Applications Development, Software Architecture and Design, and IT Business Consultancy are the job profiles that we have used for this study. The first two are clearly related to the SE field. We have included the IT Business Consultancy profile. This should not be taken to mean that IT business consultancy is a subdiscipline of software engineering. Sometimes, however, the skills needed to develop an IT business consultancy project overlap with the skills required by a software engineer. Software engineers and project managers might also aspire to be IT business consultants. It is then worthwhile analysing the relationship between the tasks described by industry for the IT Business Consultancy profile and the knowledge acquired by a software engineer educated in conformance with SE2004 or GSwE2009. We understand that Multimedia Design refers to a specific application domain and can, as such, be considered in curricula models or in elective contents. On the other hand, the Technical Support profile is not part of the classical SE discipline, as, according to Career Space, it involves jobs like computer operator, help desk operator or product support specialist.

As already mentioned, different information is provided for each job profile. Since we set out to balance industry needs with the standard SE curriculum guidelines, we will use the tasks associated with each job and relate them to the corresponding curriculum knowledge recommendations.

Table 5 shows the representative tasks associated with each of the selected job profiles. We have coded each task according to the profile to which it belongs (for example, Software & Applications Development tasks have been labelled Dev-X, Software

Table 5
Tasks associated with each job profile according to Career Space.

Software & Applications Development (Dev)	
Dev-01	Applying modern design methods and associated development tools
Dev-02	Developing code and testing algorithms and/or real-time control aspects in a modular way of working that follows the planned structure
Dev-03	Analysing system routines/modules, performance, memory size, etc., of (embedded) technical systems (when applicable)
Dev-04	Supporting project management
Dev-05	Building the system and the sub-systems according to the design and the development structure and modular set-up
Dev-06	Building prototypes of (parts of) the system
Dev-07	Co-operating with the systems architect and/or system designer
Dev-08	Designing the module test(s), assisting in the design of the integration and installation of tests. Executing the system integration, integration testing and installation
Dev-09	Developing and/or applying a version control procedure, installation procedure and making a full documentation set. Adding relevant documents like release bulletins
Dev-10	Executing the technical introduction, the installation, final testing system
Dev-11	Evaluating and arranging the maintenance & support
Dev-12	Specifying user requirements and functional requirements
Dev-13	Drawing up the plan of action for the design, the code development and other phases of the software development cycle
Software Architecture and Design (Arch)	
Arch-01	Establishing market requirements or enterprise needs
Arch-02	Building architecture
Arch-03	Developing clear, concise, accurate and coherent models of the requirements
Arch-04	Extending analysis models to solve system constraints
Arch-05	Designing solutions
Arch-06	Designing and testing prototypes
Arch-07	Defining detailed specifications
Arch-08	Creating maintenance and implementation plans
Arch-09	Enhancing products
IT Business Consultancy (BC)	
BC-01	Defining business requirements for the IT solution
BC-02	Defining IT strategy for the business (which might be, for instance, the best ways to capitalize on the latest Internet or mobile phone technologies). Participating in business needs planning & strategy process
BC-03	Identifying and defining opportunities to simplify, improve or redesign business processes using IT solutions
BC-04	Analysing, planning, configuring and developing IT solutions
BC-05	Overseeing and co-ordinating various aspects of the solution including information flow, data security, business recovery, system implementation, and change management
BC-06	Defining and ensuring implementation of standards and processing across the organization in support of the solutions

Table 6
SE2004 and GSwE2009 support for Career Space software professional tasks.

		SE2004	GSWE2009
Software & Applications Development (Dev)			
DEV-01	Applying modern design methods and associated development tools	●	⬬
DEV-02	Developing code and testing algorithms and/or real-time control aspects in a modular way of working that follows the planned structure	●	⬬
DEV-03	Analysing system routines/modules, performance, memory size, etc., of (embedded) technical systems (if applicable)	◕	○
DEV-04	Supporting project management	●	●
DEV-05	Building the system and the subsystems according to the design and the development structure and modular setup	⬬	◕
DEV-06	Building system prototypes (parts)	⬬	◕
DEV-07	Cooperating with the systems architect and/or system designer	◕	●
DEV-08	Designing the module test(s), assisting in the design of the integration and installation of tests. Executing the system integration, integration testing and installation	●	⬬
DEV-09	Developing and/or applying a version control procedure, installation procedure and making a full documentation set; adding relevant documents like release bulletins	◕	⬬
DEV-10	Executing the technical introduction, the installation, final testing system	○	◕
DEV-11	Evaluating and arranging maintenance & support	◕	●
DEV-12	Specifying user requirements and functional requirements	●	⬬
DEV-13	Drawing up the plan of action for the design, code development and other phases of the software development cycle	◕	●
Software Architecture and Design (Arch)			
ARCH-01	Establish market requirements or enterprise needs.	○	◕
ARCH-02	Building architecture	◕	●
ARCH-03	Developing clear, concise, accurate and coherent requirements models	●	⬬
ARCH-04	Extending analysis models to solve system constraints	○	◕
ARCH-05	Designing solutions	●	⬬
ARCH-06	Designing and testing prototypes	●	⬬
ARCH-07	Defining detailed specifications	◕	●
ARCH-08	Creating maintenance and implementation plans	◕	●
ARCH-09	Enhancing products	◕	●
IT Business Consultancy (BC)			
BC-01[1]	Defining business requirements for the IT solution	○	○
BC-02[2]	Defining IT strategy for the business (which might be, for instance, the best ways to capitalize on the latest Internet or mobile phone technologies);participating in business needs planning & strategy process	○	○
BC-03	Identifying and defining opportunities to simplify, improve or redesign business processes using IT solutions	○	○
BC-04	Analysing, planning, configuring and developing IT solutions	○	○
BC-05	Overseeing and coordinating various aspects of the solution including information flow, data security, business recovery, system implementation, and change management	◕	◕
BC-06	Defining and ensuring implementation of standards and processing across the organization in support of the solutions	◕	●

Architecture and Design tasks as Arch-X and IT-Business Consultancy tasks as BC-X).

6. SE2004 and GSwE2009 knowledge coverage

In the following, we examine the core knowledge to which SE undergraduate and postgraduate programme graduates will have had access in order to perform the tasks proposed by the software industry. As already mentioned, we will focus on SE programmes conforming to SE2004 and GSwE2009 recommendations, respectively. Notice that a student that has taken a GSwE2009-based graduate programme will not necessarily have taken a SE2004-based undergraduate programme. For this reason, these analyses are conducted for the two recommendations separately.

In summary, we will examine whether the SE2004 and GSwE2009 cores provide knowledge that is useful for performing each Career Space task related to each of the three software development-related profiles: Software and Applications Development, Software Architecture and Design and IT Business Consultancy.

This analysis is based on the KA and Unit descriptors proposed by SE2004 and GSwE2009, as our focus is to provide a global analysis of what knowledge these standards provide to cover key tasks in the software industry. As we will see later, we propose to replicate this study for specific programmes with particular compulsory and elective contents and percentages in future research.

Table 6 summarizes graphically how SE2004 and GSwE2009 support the different tasks of the Career Space profiles under study. This relationship is represented by the existence in the corresponding standard of the core knowledge needed to perform the respective task. When a standard does not provide any of the knowledge required to perform a particular task, it will be labelled with a non-coverage icon (○); when the knowledge is clearly insufficient, it will be marked with a limited coverage icon (◕); and when the knowledge is likely to be sufficient, we use the coverage icon (●). If the knowledge is sufficient and delivered at application or analysis level, we use an extra coverage icon (⬬). A qualitative analysis such as this is usual in studies that compare knowledge from different sources (see, for example, Mead et al., 2010).

Only a professional with years of experience might be 100% guaranteed to perform a task properly. Consequently, by looking at the extent of coverage provided by the SE2004 or GSwE2009 cores to perform a task, we are approaching the issue from the angle of the academic knowledge required to tackle that task. As already mentioned, this study sets out to examine the extent to

Table 7
Relationship between SE2004 and GSwE2009 KAs and the Software and Applications Development profile.

Career Space profile: Software and Applications Development	SE2004 KA	GSWE2009 KA
Dev-01: Applying modern design methods and associated development tools	Software Design	Software Design
Dev-02: Developing code and testing algorithms and/or real-time control aspects in a modular way of working that follows the planned structure	Software Verification and Validation Computing Essentials	Testing and Software Construction
Dev-03: Analysing system routines/modules, performance, memory size, etc. of (embedded) technical systems (when applicable)	Software quality	-
Dev-04: Supporting project management	Software Management	Software Engineering Management
Dev-05: Building the system and the sub-systems according to the design and the development structure and modular set-up	Computing Essentials	Software Construction
Dev-06: Building prototypes of (parts of) the system	Computing Essentials	Software Construction
Dev-07: Co-operating with the systems architect and/or system designer	Professional Practice	System Engineering
Dev-08: Designing the module test(s), assisting in the design of the integration and installation of tests. Executing the system integration, integration testing and installation	Software Verification and Validation	Testing
Dev-09: Developing and/or applying a version control procedure, installation procedure and making a full documentation set. Adding relevant documents like release bulletins	Software Management	Configuration Management
Dev-10: Executing the technical introduction, the installation, final testing system	Software Verification and Validation	Testing
Dev-11: Evaluating and arranging the maintenance & support	Software Management	Software Maintenance
Dev-12: Specifying user requirements and functional requirements	Software Modelling and Analysis	Requirements Engineering
Dev-13: Drawing up the plan of action for the design, the code development and other phases of the software development cycle	Software Management	Software Engineering Management

which the SE2004 and GSwE2009 cores provide this knowledge. We do not intend to determine whether a graduate of an SE2004- or GSwE2009-based programme is able to perform a particular task. This would be a far from easy undertaking, as other factors influence this finding. In the introduction of the paper, for example, we discussed the importance of an adequate pedagogical approach to deliver the knowledge (neither SE2004 nor GSwE2009 include this information, nor is it specified in the description of specific SE programmes). Additionally, there would be no guarantee of ability to perform without work experience. Therefore, this parameter should also be considered complementary to our study.

Sections 6.1–6.3 detail SE2004 and GSwE2009 support for the Career Space profiles listed in Table 6. Section 6.4 provides a more global discussion of the results.

6.1. *Software & Applications Development Profile*

Table 7 shows the SE2004 and GSwE2009 KAs that provide knowledge for performing the corresponding Career Space tasks. The details of these KAs, units, Bloom's taxonomy, etc. have been shown in Sections 3 and 4, respectively, and are discussed below related to each task.

6.1.1. *Dev-01. Applying modern design methods and associated development tools*

Both SE2004 and GSwE2009 suggest core knowledge to cover this competence as shown in Table 7. In SE2004 this knowledge is acquired in the Software Design KA, which is taught at the comprehension and application level (see Section 3). In GSwE2009, it is again acquired in the Software Design KA, albeit at the application level in this case, and, specifically, in the Software Design Strategies and Methods and Software Structure and Architecture Units, also at the analysis level (Section 4). In both cases, the minimum in-class time recommended in the curriculum is 9%. Therefore, although, as expected, both recommendations could provide useful knowledge for performing this task, this knowledge is acquired in more depth in GSwE2009-based programmes. For this reason, Table 6 indicates that SE2004 provides coverage and GSwE2009 extra coverage for this task.

6.1.2. *Dev-02. Developing code and testing algorithms and/or real-time control aspects in a modular way of working that follows the planned structure*

Programming and testing knowledge is required to support this competence. This knowledge is located in the Testing-related Units of the Software Verification and Validation KA, as well as the Computing Essentials KA of SE2004. This knowledge is acquired at the comprehension and application levels. In the case of GSwE2009, this knowledge was acquired through the Testing and Software Construction KA. In both KAs, the knowledge is proposed at application level and is extended to the analysis level for Test-Related Measures. The minimum in-class time suggested in both SE2004 and GSwE2009 is about 4%. As above, the knowledge to perform this task is acquired in more depth in graduate programmes. Thus, this task is denoted with coverage and extra coverage for SE2004 and GSwE2009, respectively, in Table 6.

6.1.3. *Dev-03. Analysing system routines/modules, performance, memory size, etc., of (embedded) technical systems (if applicable)*

It is not easy to pinpoint specific knowledge to address this task in the SE2004 KAs. Generally, this knowledge could be acquired through the Software Quality Concepts or Software Quality Process Units within the Software Quality KA (see Table 7). Taking into account, however, that this knowledge should be acquired at the knowledge and comprehension level and the in-class time is minimal, there is no guarantee that this will be sufficient for performing this task. GSwE2009 does not include any knowledge to address this task. Thus, this task has been labelled with partial coverage and no coverage, respectively, in Table 6.

6.1.4. *Dev-04. Supporting project management*

The SE2004 knowledge core includes project management knowledge in the Software Management KA (see Table 7). The percentage of allocated in-class time is 4% of the core, and it is acquired at the knowledge, comprehension and very low application levels. SE2004 also includes knowledge that may serve as support for this task, such as communication skills acquired in the Professional Practice KA at the application level with a very low in-class time of 2%. Therefore, an entry-level professional graduating from these programmes would be able to perform this job, although further practical knowledge about project management would be an advantage.

As regards GSwE2009, as shown in Table 7, the knowledge related to this task is acquired in the Software Engineering Management KA at the knowledge and mostly application level with an in-class time of 7–8%. However, the GSwE2009 core does not include the communication skills that may also be necessary to support project management. Although an advantage, these communication skills are not strictly necessary for this task because, these days, much of the information for project management can be gathered automatically from development environments. So, in view of the importance placed on project management knowledge in GSwE2009, this has been considered adequate for an entry-level professional graduating from such programmes to be able to properly perform this task.

Thus, Table 6 indicates that both recommendations provide coverage for this task.

6.1.5. *Dev-05. Building the system and the sub-systems according to the design and the development structure and modular set-up*

As seen in Table 7, SE2004 provides knowledge for developing this task within the Computing Essentials KA, and, specifically, the Computer Science Foundations Unit (with an in-class time of 28% of the SE2004 core) and Construction Technologies Unit (with 4% of in-class time). This knowledge is delivered to the levels of comprehension and application. The GSwE2009 accounts for this knowledge within the Software Construction KA (see Table 7). This KA covers foundations for building software, software management and the practical concerns. This knowledge is also acquired at the application level; however, its percentage in-class time is negligible (1–3%).

Undergraduate SE programmes cover all the knowledge for supporting this type of tasks. As graduate programmes have a greater focus on higher life-cycle tasks, the knowledge acquired will not be sufficient unless professionals have received a previous undergraduate education in SE. However, Table 6 does not account for this assumption, and the respective task has been marked with partial coverage for GSwE2009.

6.1.6. *Dev-06. Building prototypes of (parts of) the system*

Like task Dev-05, the knowledge required to complete this task is acquired within the Computing Essentials KA in SE2004 and the Software Construction KA in GSwE2009, and the same discussion therefore applies.

6.1.7. *Dev-07. Co-operating with the systems architect and/or system designer*

The knowledge core covered by SE2004 focuses on meeting software development needs, and does not specifically focus on systems. On this ground, although it proposes knowledge related to soft skills, such as communication, within the Professional Practice KA, this knowledge cannot be guaranteed to be sufficient for performing this task, as it would also be important to have technical knowledge about the particularities of these roles. On the other hand, although, as already mentioned, there is no reference in the GSwE2009core to communication skills, it does contain a System Engineering KA accounting for 2–3% of in-class time. This KA focuses on technical knowledge, which, in the case of the System Design Unit, is acquired at comprehension and application level. On this ground, we understand that a graduate of a GS2WE2009-based programme is more likely to be able to perform this task. For this reason, Table 6 labels this task with partial coverage and coverage, respectively.

6.1.8. *Dev-08. Designing the module test(s), assisting in the design of the integration and installation of test. Executing the system integration, integration testing and installation*

SE2004 provides knowledge to cover this task within the Testing Unit of the Software Verification and Validation KA. This knowledge accounts for 4% of the core. Similarly, GSwE2009 provides this type of knowledge within the Testing KA with an in-class time of 4–6%. The knowledge differs as to the level at which it is acquired, being comprehension and application for SE2004 and application and analysis for GSwE2009. For this reason, this task has been marked with coverage and extra coverage in Table 6, respectively.

6.1.9. *Dev-09. Developing and/or applying a version control procedure, installation procedure and making a full documentation set. Adding relevant documents like release bulletins*

The knowledge to meet the requirements of this task is acquired by learning the Software Management KA, specifically the Software Configuration Unit, in SE2004, whose allocated workload is, however, negligible (less than 1%). On the other hand, Configuration Management is a standalone KA in GSwE2009, and the Configuration Control Unit contains knowledge on version control, also with a minimum workload allocation (2–3%). The difference between the two guidelines lies in the level of depth with which the knowledge is acquired, this being comprehension and knowledge in SE2004 and application in GSwE2009.

In Table 6, this task has been denoted with partial coverage for SE2004 and coverage for GSwE2009, as the knowledge is acquired at a greater practical level and with more in-class time in this case. Even so, it is debatable, in view of the importance of version control and configuration management in industry, whether an in-class time of 3% would be sufficient.

6.1.10. *Dev-10. Executing the technical introduction, the installation, final testing system*

SE2004 recommends the inclusion of knowledge on deployment within the Testing Unit in the Software Verification and Validation KA. However, this knowledge is catalogued as optional (which has been shown in Table 7 with this KA in parenthesis). Therefore, there is no guarantee that graduates of programmes based on this guideline will have the knowledge necessary to perform this task. GSwE2009, on the other hand, includes application-level knowledge of acceptance tests within the Test Levels Unit of the Testing KA with an in-class time of approximately 1%, where no reference is made to installation topics as specified in the task.

Based on the core, then, a recent graduate of a SE2004-based programme cannot be guaranteed to have the knowledge to carry out this task. On the other hand, a recent graduate of a GSwE2009-based programme will have partial knowledge. On this ground, this task has been denoted with no coverage and partial coverage, respectively, in Table 6.

6.1.11. Dev-11. Evaluating and arranging maintenance and support

The knowledge for performing this task is basically supplied in GSEW2009 within a special-purpose Software Maintenance KA (as shown in Table 7). However the in-class time suggested is of 3–4% generally acquired at a level of application. In SE2004, on the other hand, this knowledge is embedded within the Software Management KA in the Management Concepts Unit with an in-class time of less than 0.5% and acquired at the knowledge level.

Therefore, the core gives no guarantee that SE2004-based programme graduates will have acquired sufficient knowledge to perform this task. This does not, however, apply to GSEW2009. Therefore, Table 6 denotes this task with partial coverage and coverage, respectively.

6.1.12. Dev-12. Specifying user requirements and functional requirements

SE2004 contains a KA called Software Modelling and Analysis, which provides the knowledge required to perform this task and has an allocated in-class time of 11% of the core. Within GSwE2009 there is also a Requirements Engineering KA, delivering this knowledge with an in-class time of 6–8%. The difference between the two lies in the level of acquisition, which is knowledge, comprehension or application in SE2004, as opposed to application or even analysis in GSwE2009.

Thus, recent graduates of courses designed according to the SE2004 core have basic knowledge and skills, whereas recent graduates of programmes of study based on GSwE2009 have more internalized and consolidated skills with respect to this task. For this reason, this task has been marked with coverage and extra coverage in Table 6.

6.1.13. Dev-13. Drawing up the plan of action for the design, code development and other phases of the software development cycle

This task is accounted for in SE2004 by the Software Management KA, specifically the Project Planning Unit, where the skills are acquired at comprehension and application level, and in the Project Personnel and Organization Unit, where they are taught to knowledge level. Each Unit has an allocated in-class time of 1% of the core. As regardsGSwE2009, this task is covered in the Software Engineering Management KA, specifically the Software Project Planning Unit, where the knowledge is acquired at the application level, again with an in-class time of 1% of the core.

Recent graduates of courses based on the GSwE2009 core have more practical skills for performing this task than recent graduates of programmes of study based on the SE2004 core. However, an in-class time of 1% may turn out to be insufficient for tackling the potential problems related to this topic in industry, meaning that we have denoted this task with partial coverage and coverage, respectively, in Table 6.

6.2. Software Architecture and Design Profile

Table 8 summarizes the relationships between the Software Architecture and Design Career Space profile and SE2004 and GSWE2009 KAs. The details of these relationships are described below.

6.2.1. Arch-01. Establish market requirements or enterprise needs

The core of SE2004 includes a KA called Software Modelling and Analysis, containing a unit called Types of Models. This unit covers the specific topic of enterprise modelling, where the skills to perform this task are acquired at the knowledge level. However, this topic is optional within the core, meaning that there is no guarantee that it will be included in programmes. On the other hand, GSwE2009 has a KA called Systems Engineering, containing a Requirements Unit, which accounts for requirements extraction. Therefore, this knowledge could contribute to supporting this task for particular domains, although the in-class time is negligible (0.4% of the core at comprehension and application level). Therefore, Table 6 denotes this task with no coverage and partial coverage for SE2004 and GSwE2009, respectively.

6.2.2. Arch-02. Building architecture

In the SE2004 core, this task is covered by the Architectural Design Unit of Software Design KA. It has an in-class time of 2% within the core and is acquired at the knowledge and application level. On the other hand, GSwE2009 includes the knowledge required to perform this task in the Software Structure and Architecture Unit of the Software Design KA, also with an in-class time of about 2%. However, the level of learning is analysis and application in this case.

The profile of recent graduates of both educational programmes based on the SE2004 and GSwE2009 core could be adequate for this task, although GSwE2009 graduates acquire this knowledge in greater depth (analysis and application level). In our view, an in-class time of only 2% (which is the minimum suggested in both recommendations) might not be enough to deal with this task in a real situation. On this ground, this task has been specified as having partial coverage and coverage in Table 6.

6.2.3. Arch-03. Developing clear, concise, accurate and coherent models of the requirements

In terms of software development, this task is equivalent to properly modelling requirements, meaning that this task is covered in the SE2004 core by the knowledge acquired after studying the Software Modelling and Analysis KA with a coverage of 11% at the comprehension and knowledge level. On the other hand, the knowledge for performing this task is acquired in GSwE2009's Requirements Engineering KA with an in-class time of 6–8% acquired at the application level.

Table 8
Relationship between SE2004 and GSwE2009 KAs and Software Architecture and Design Profile.

Career Space profile: Software Architecture and Design	SE2004 KA	GSWE2009 KA
Arch-01: Establishing market requirements or enterprise needs	Software Modelling and Analysis	Systems Engineering
Arch-02: Building architecture	Software Design	Software Design
Arch-03: Developing clear, concise, accurate and coherent models of the requirements	Software Modelling and Analysis	Requirements Engineering
Arch-04: Extending analysis models to solve system constraints	Software Modelling and Analysis	Requirements Engineering
Arch-05: Designing solutions	Software Design	Software Design
Arch-06: Designing and testing prototypes	Software Design Software Verification and Validation	Software Design Testing
Arch-07: Defining detailed specifications	Software Modelling and Analysis	Requirements Engineering
Arch-08: Creating maintenance and implementation plans	Software Management	Software Maintenance
Arch-09: Enhancing products	Software Management	Software Maintenance

Table 9
Relationship between SE2004 and GSwE2009 KAs and Business Consultancy profile.

Career space profile: Business Consultancy	SE2004 KA	GSWE2009 KA
BC-01: Defining business requirements for the IT solution	–	–
BC-02: Defining IT strategy for the Business (which might be, for instance, the best ways to capitalize on the latest Internet or mobile phone technologies). Participating in business needs planning & strategy process	–	–
BC-03: Identifying and defining opportunities to simplify, improve or redesign business processes using IT solutions	–	–
BC-04: Analysing, planning, configuring and developing IT solutions	–	–
BC-05: Overseeing and co-ordinating various aspects of the solution including information flow, data security, business recovery, system implementation, and change management	Software Management	SE Management
BC-06: Defining and ensuring implementation of standards and processing across the organization in support of the solutions	Software Quality	Software Quality

In both cases, recent graduates of educational programmes based on the core of SE2004 or GSwE2009 acquire knowledge to perform this task. It is true, though, that GSwE2009-based programme graduates have more practical knowledge of how to perform the task. On this ground, this task was defined with coverage and extra coverage in Table 6.

6.2.4. Arch-04. Extending analysis models to solve system constraints

The Requirements Fundamentals Unit of the Software Modelling and Analysis KA in the SE2004 core sets out knowledge for expressing requirements that take into account project but not system constraints. In this case, then, we can say that this task is not supported.

On the other hand, the core of GSwE2009 proposes a Requirements Engineering KA including the Fundamentals of Requirements Engineering Unit that takes into account knowledge of system design constraints. This knowledge is acquired at the comprehension and application level with an in-class time of approximately 1% of the core. Therefore, a recent graduate of a programme based on GSwE2009 can be considered to have some knowledge for developing this task.

On this ground, this task has been marked with no coverage and partial coverage, respectively, in Table 6.

6.2.5. Arch-05. Designing solutions

The knowledge for tackling this task is similar to what we discussed under Arch-1 Applying modern design methods and associated development tools. Within the SE2004 core, this task is covered by knowledge acquired after learning the Software Design KA that has an in-class time of 9% of the core and is acquired at the comprehension and application level and, in odd units, at the knowledge level. On the other hand, GSwE2009 has a KA called Software Design with an in-class time of from 9 to 11%, containing the knowledge required to perform this task, which is acquired mainly at the application and occasionally at the analysis level.

In both cases, recent graduates of the educational programmes based on the SE2004 or GSwE2009 cores are qualified to perform this task, although it is true that the recent graduates of GSwE2009-based programmes have more practical knowledge to do the job. In this case, this task has been marked with extra coverage in Table 6.

6.2.6. Arch-06. Designing and testing prototypes

There is no section of the SE2004 core that specifically focuses on prototyping. Now if we assume that prototype design and testing is based on similar principles to the design and testing of any other system the thinking would be as follows. There is the Software Design KA concerned with software design, and, therefore, the same discussion would apply as for the Arch-5 task. On the other hand, there is a Software Verification and Validation KA, containing the Testing Unit with an in-class time of 4% of the core, where the knowledge is acquired at application and comprehension level. Following the above reasoning, the SE2004 provides knowledge to support this task.

On the other hand, GSwE2009 proposes a Software Design KA where the same argument as for Arch-5 task applies. As far as testing is concerned, it proposes the Testing KA with an in-class time of from 4 to 6% of the core, where most of the knowledge is acquired at application level and some occasionally at analysis and comprehension level. Table 6 indicates, therefore, that GSwE2009 provides extra coverage for this task.

6.2.7. Arch-07. Defining detailed specifications

SE2004 has a KA called Software Modelling and Analysis in which there is a Specification and Documentation Unit with an in-class time of around 1% of the core, where the knowledge is acquired at comprehension and application level. Similarly, GSwE2009 contains the Requirements Engineering KA, which covers the Requirements Specification Unit with an in-class time of 1% of the core, approximately. In this case, however, all the knowledge is acquired at application level.

Recent graduates of a GSwE2009-based programme are in a better position to perform this task than SE2004 graduates because they have more practical knowledge. However, the low percentage of in-class time may not be sufficient for tackling this task in the real world. For this reason, the task has been specified in Table 6 as having partial coverage and coverage, respectively.

6.2.8. Arch-08. Creating maintenance and implementation plans

This task accounts for the creation of maintenance and implementation plans. The cores of both SE2004 and GSwE2009 contain aspects related to maintenance but not to implementation plans. The points related to maintenance are detailed in the following.

SE2004 has a knowledge area called Software Management, within which the Software Configuration Management Unit accounts for very general aspects of maintenance. The workload is insignificant, less than 1% of in-class time of the core where knowledge is acquired at knowledge level.

On the other hand, GSwE2009 includes a Software Maintenance KA with an in-class time of from 3 to 4% of the core, where skills are acquired to develop this task mostly at application level.

On the grounds of the above, recent graduates of a programme based on SE2004 will have very little knowledge of this task, whereas recent graduates of a GSwE2009-based programme will be able to satisfactorily perform the part of this activity referred to maintenance plans. For this reason, we have specified this task as having partial coverage and coverage, respectively, in Table 6.

6.2.9. Arch-09. Enhancing products

This task is related to product maintenance. Within SE2004, the Software Management KA includes the Software Configuration Management Unit, which accounts for very general aspects of maintenance, with an insignificant workload of less than 1% of the core in-class time, where knowledge is acquired at knowledge level.

On the other hand, GSwE2009 includes a Software Maintenance KA with an in-class time of from 3 to 4% of the core, where skills are acquired to perform this task mostly at application level.

Therefore, recent graduates of a degree programme based on SE2004 would not have knowledge enough to execute a maintenance plan. In the case of recent graduates of a GSwE2009-based programme, the knowledge required to deploy and execute a maintenance plan for a particular product are acquired at application level. Therefore, this task was labelled with partial coverage and coverage, respectively, in Table 6.

6.3. IT Business Consultancy profile

Finally, let us look at how SE2004 and GSwE2009 support the Business Consultancy profile tasks.

Table 9 summarizes the relationships between the Business Consultancy Career Space profile and SE2004 and GSWE2009 KAs.

6.3.1. BC-01. Defining business requirements for the IT solution

Although neither SE2004 nor GSwE2009 have any KA covering this task as such, it should be taken into account as part of both the functional and non-functional requirements that are gathered when implementing a software project. According to the current core of the two recommendations, however, there is no guarantee that recent graduates of these programmes will have knowledge enough to perform this task with respect to specific business requirements.

6.3.2. BC-02. Defining IT strategy for the business (which might be, for instance, the best ways to capitalize on the latest Internet of mobile phone technologies). Participating in business needs planning and strategy process

This task focuses on aspects that are beyond the scope of the core of both SE2004 and GSwE2009, as the development of IT strategies is broader than the development of software projects. Therefore, both recommendations are specified as providing no coverage for this task in Table 6.

6.3.3. BC-03. Identifying and defining opportunities to simplify, improve or redesign business processes using IT solutions

This task refers to business processes; therefore, neither the SE2004 nor the GSwE2009 core specifies knowledge for developing this type of activity. As above, this task was denoted with no coverage in Table 6.

6.3.4. BC-04. Analysing, planning, configuring and developing IT solutions

Like BC-02, this task focuses on aspects that are beyond the scope of the core of both SE2004 and GSwE2009, as the development of IT strategies is broader than the development of software projects. In both cases, therefore, this task has been denoted with no coverage.

6.3.5. BC-05. Overseeing and co-ordinating various aspects of the solution including information flow, data security, business recovery, system implementation, and change management

This task is related to the IT solution. So, although there are specific KAs related to Software Management in both SE2004 and GSwE2009 concerning the control, evaluation and running of software project change management, the performance of this task would involve additional knowledge about the IT domain. For this reason, this task has been marked with partial coverage in both cases in Table 6.

6.3.6. BC-06. Defining and ensuring implementation of standards and processing across the organization in support of the solutions

This task could be related to the application of software quality standards and processes in an organization. In this case, the way in which SE2004 and GSwE2009 would account for the knowledge for undertaking this task would be as follows. In the case of SE2004, the Software Quality KA accounts for an in-class time of 3% of the core, and knowledge is acquired at comprehension and knowledge level. On the other hand, there is also a Software Quality KA in GSwE2009 with an in-class time of 3–4% of the core where knowledge is acquired at application level.

Recent graduates of a programme based on GSwE2009 will have practical skills for performing this activity, whereas recent graduates of a SE2004-based programme will have only knowledge-level and no practical skills for performing the task. However, as the percentage of in-class time is limited in both cases, this task has been marked with partial coverage and coverage, respectively.

6.4. General knowledge coverage analysis

As expected, when industry needs to hire employees to perform tasks related to the Software and Applications Development Profile, we find that, generally speaking and save exceptions that we discuss later, programmes based on both SE2004 and GSwE2009 can support these tasks. On the other hand, again in general terms, we find that recent graduates of a programme based on GSwE2009 have more basic knowledge than recent graduates of a SE2004-based programme, mainly due to the depth and practicality with which GSwE2009 deals with the knowledge. This applies to tasks related to software requirements (DEV-12), software design (DEV-0) and the verification and validation process (DEV-02 and DEV-08). SE2004-based programmes provide core knowledge to cover these tasks but at the Bloom taxonomy levels of comprehension and application. GSwE2009 programmes extend such knowledge for most relevant SEEK units to the analysis level, thereby providing a more practical view.

Other tasks where GSwE2009-based programmes enrich SE2004 knowledge are related to planning (DEV-13), configuration management (DEV-09) and cooperation with the system architect (DEV-07). The first two tasks are covered in SE2004 at the comprehension and knowledge levels, whereas GSwE2009 deals with them at application level. In both cases, however, the proportion of recommended core knowledge is very low (1% and 3%, respectively). As we argued in Section 6.1, it is very debatable whether such knowledge is sufficient for tackling potential problems related to this topic in industry.

There are some exceptions to this general rule according to which GSwE2009 rounds out SE2004 knowledge. They relate to tasks concerning construction and programming (DEV-05 and DEV-06). As SE2004 programmes place the emphasis on knowledge related to software construction and implementation, recent graduates would have more in-depth knowledge than recruits from graduate programmes that focus on higher lifecycle activities, as discussed in Section 6.1. The percentage of in-class time spent on this type of knowledge in SE2004 and GSwE2009, that is, 28% and

3%, respectively, is a perfect reflection of this situation. GSwE2009 graduates would not be able to perform such tasks unless they have received a previous undergraduate education in SE or computer science. Thus, entry-level professionals that have completed a SE2004-based programme would be better qualified for this kind of programming tasks.

Exceptionally, there are also tasks for which the cores of neither SE2004 nor GSwE2009 provide enough knowledge. This applies to tasks related to system routines/modules, performance, memory, size analysis (DEV-03) and deployment (DEV-10). In the first case, the core of undergraduate studies based on SE2004 does not provide comprehensive knowledge for performing the task (as already mentioned, it only provides some very abstract knowledge about product quality assurance at knowledge and comprehension level with minimal in-class time of around 1%). On the other hand, this type of knowledge is not usually covered by either graduate programmes in SE or the GSwE2009 core. There is no guarantee that recent graduates from undergraduate or postgraduate programmes recruited for this profile would be skilled enough perform this type of tasks unless they have taken optional subjects covering this topic.

As regards deployment (DEV-10), recent graduates of a basic compulsory programme based on SE2004 will not necessarily have the knowledge to perform this task. As discussed in Section 6.1, SE2004 suggests optional knowledge about deployment. A recent graduate of a programme based on GSwE2009 will only have partial knowledge for performing this task, specifically knowledge related to acceptance testing and with a very low in-class time (close to 1%, as we have already seen). Neither SE2004 nor GSwE2009 graduates can be expected to perform this task properly unless they have taken optional subjects or are given special-purpose training by the company.

Regarding, the Software Architecture and Design Profile, most tasks in this profile are covered by SE2004 and GSwE2009. However, it would be preferable to employ recent graduates of GSwE2009-based programmes in so far as they have a greater level of core knowledge and skills than recent graduates of programmes based on SE2004. This applies to tasks related to the development of detailed requirements models (ARCH-03) and designing and testing solutions and prototypes (ARCH-05 and ARCH-06, respectively). As discussed in Section 6.2, the core knowledge provided by SE2004 covers ARCH-03, ARCH-05 and ARCH-06 at comprehension and knowledge level, whereas GSwE2009 stretches to application level, thereby providing a more practical view of such topics.

Other tasks related to building architectures (ARCH-02), detailed specifications (ARCH-07), maintenance and implementation plans (ARCH-08) and enhancing products (ARCH-09) are significantly enriched by GSwE2009-based programmes over SE2004 programmes. Although the knowledge for ARCH-02 and ARCH-07 is acquired at the comprehension and application levels in SE2004 and at application, and sometimes analysis, level in GSwE2009, the in-class time is very low in both standards. Tasks related to maintenance and implementation plans and enhancing products are partially covered by SE2004-based programmes. As discussed in Section 6.2, only very general aspects can be identified with a very low in-class time. GSwE2009 programmes, on the other hand, recommend a higher in-class time for ARCH-08 and ARCH-09 acquired at application level.

Neither the SE2004 nor the GSwE2009 cores deliver knowledge of some tasks related to defining market requirements (ARCH-01) and extending analysis models to solve system constraints (ARCH-04). As we saw in Section 6.2, this knowledge is not included in the core at the undergraduate level and is covered, secondarily, at graduate level with an almost negligible in-class time. In this case, although it is preferable to hire recent graduates of a programme based on GSwE2009, the knowledge acquired in the core would need to be supplemented either by optional subjects or further training.

Finally, as regards the recruitment process for performing tasks related to the IT Business Consultancy profile, note that, generally, the cores of neither SE2004 nor GSwE2009 provide knowledge for tackling the related tasks. Consequently, this profile is not very well accounted for in the cores of undergraduate and graduate programmes based on the standards examined here. Both SE2004 and GSwE2009 graduates will require further training (unless they have taken electives covering this knowledge).

The analysed cores include some knowledge about only two of the IT Business Consultancy tasks related to overseeing and coordinating various aspects of the solution (BC-05) and defining and ensuring the implementation of standards across the organization (BC-06). In the first place, both SE2004 and GSwE2009 provide some knowledge related to software management, but more IT management knowledge is needed to perform this task properly in both cases. In this respect, it would make almost no difference whether a recent graduate of a programme based on SE2004 or GSwE2009 were hired, as they receive more or less the same training. Only in the case of BC-06 will recent graduates of a programme based on the GSwE2009 core be more skilled at performing the task, as they acquire some knowledge at application level.

7. Conclusions and discussion

We have studied the core knowledge suggested by graduate and undergraduate SE educational recommendations and its relationship to representative tasks to be performed by a software practitioner according to industry. We set out to determine not whether graduates are capable of performing a task properly but whether or not they have received core technical knowledge to do the job. Experience is another key requirement for performing tasks, but this is only acquired with practice. Our aim is to analyse whether graduates have at least studied a programme whose core provides the basic knowledge to perform the respective tasks.

We have found that none of the three industry profiles is completely covered by either SE2004 or GSwE2009. The biggest gap found concerns tasks associated with the IT Business Consultancy profile. Knowledge required by such tasks is beyond the classical technical knowledge that we are accustomed to in most undergraduate and graduate SE programmes. In this respect, our results are consistent with other studies that claim there is a need to move from technical to the business issues (Gartner, 2005; Davey and Tatnall, 2008). Such knowledge would probably be better suited for a graduate than an undergraduate programme, but, in any case, its increasing relevance is unquestionable.

Missing soft skills related to core knowledge mainly in GSwE2009 are also noteworthy. Knowledge related to leadership, negotiation or giving presentations, identified for example by Lethbridge (2000) and Kitchenham et al. (2005) as relevant from an industry viewpoint, is also missing in this core. This affects tasks like Supporting Project Management or Co-operating with the System Architect and/or System Designer. Although this knowledge could be delivered outside of the core, specific programme definitions might, in view of its relevance, want to address the issue and somehow guarantee that the knowledge is included as part of the core.

In sum, we have identified some specific gaps in the core knowledge suggested by *de facto* SE education standards. These results could be used in different ways. On one hand, academics responsible for defining SE programmes could take them into account to develop curricula that train students to fit specific profiles. Industry can also use this information as a reference about the core knowledge of their potential employees, the tasks for which they are best suited and the additional knowledge that they might require. Finally, graduates could use this knowledge to identify which post it is best to apply for and the further training that they need depending on the job they would like to do.

The results of this study have to be contextualized by companies and how they develop software. In companies that outsource software construction, for example, software engineers should have specific capabilities for managing outsourcing. This would mean rounding out their training with respect to the SE2004 or GSwE2009 cores with specialized knowledge on this topic. Where the focus is on subcontractor companies the technical knowledge provided by SE2004 and GSwE2009 might well be sufficient.

The analysis conducted here was based on industry needs reported by the Career Space consortium, specifically focused on software engineering. However, the Gartner report, mentioned in Section 2, gives a forecast of areas of expertise in the ICT profession (Gartner, 2005). Although this report gives only a general description of these areas, it is worth mentioning because it supplements some aspects of the tasks proposed by Career Space. In actual fact, Gartner stresses tasks related to technology infrastructure and services, business intelligence, outsourcing and global skills. These are more global areas of expertise that cannot be related on a one-to-one basis with specific Career Space tasks. Thanks to the visionary approach of Career Space, however, services, business intelligence, the technical side of global skills and outsourcing would be covered by tasks related to the application of modern design methods and tools. On the other hand, Career Space does not explicitly deal with the management side of outsourcing and global skills, as management activities are very much focused on project management.

In any case, the cores of neither SE2004 nor GSwE2009 account for knowledge that can cover these professional practices. The Gartner report also stresses business-related aspects as part of professional profile requirements. As already mentioned for the Career Space IT Business Consultancy profile, the cores of neither SE2004 nor GSwE2009 cover these concerns.

Finally, note that it might be interesting to replicate this study for individual programmes with detailed subjects and specializations. This way, students and industry would know, on one hand, which profiles graduates of a particular specialization are best qualified for, as well as what is missing from a specialization to qualify graduates for a specific profile.

Another complementary study might be related to the replication of this analysis for computer science programmes. Notice that there are some tasks whose characteristics do not fit well in SE programmes (for example, Analysing system routines/modules (Dev-03)). However, it might warrant coverage by computer science programmes. Companies hiring professionals for these tasks would probably opt for computer science graduates instead of software engineering graduates.

References

Aasheim, C.L., Li, L., Williams, S., 2009. Knowledge and skills requirements for entry-level information technology workers: a comparison of industry and academia. Journal of Information Systems Education 20, 349–356.

Abraham, T., Beath, C., Bullen, C., Gallagher, K., Goles, T., Kaiser, K., Simon, J., 2006. IT workforce trends: implications for IS programs. Communications of the Association for Information Systems 17, 1147–1170.

Bailey, J.L., Stefaniak, G., 2001. Industry perceptions of the knowledge, skills and abilities needed by computer programmers. In: Proceedings of the 2001 ACM SIGCPR Conference on Computer Personnel Research, pp. 93–99.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R., 1956. Taxonomy of Educational Objectives: The Classification of Educational Goals; Handbook I: Cognitive Domain. Longmans, Green, New York.

Bureau of Labor Statistics, US Computer and Information Systems Managers. Department of Labor, Occupational Outlook Handbook, 2008–2009 edition. http://www.bls.gov/oco/ocos258.htm.

Bureau of Labor Statistics, US Computer Scientist and Database Administrators. Department of Labor, Occupational Outlook Handbook, 2008–2009 edition. http://www.bls.gov/oco/ocos042.htm.

Career Space, 2001. Curriculum Development Guidelines. New ICT Curricula for the 21st Century Designing Tomorrow's Education, http://ec.europa.eu/enterprise/sectors/ict/files/ictcurricula21century_en.pdf.

Davey, B., Tatnall, A., 2008. Where will professional software engineering education go next? IFIP International Federation for Information Processing. Learning to Live in the Knowledge Society, vol. 281. Springer, pp. 185–192.

Gartner, 2005. The IT Professional Outlook: Where Will We Go From Here? Gartner, Orlando, FL.

Gallavin, M., Truex, D., Kvasny, L., 2004. Changing patterns in IT skills sets 1988–2003: a content analysis of classified advertising. The Database for Advances in Information Systems 35, 64–87.

Generic ICT Skills Profiles: Future Skills for Tomorrow's World (Career Space), 2001. US Department of Education. Educational Resources Information Center. http://www.eric.ed.gov.

Kim, Y., Hsu, J., Stern, M., 2006. An update on the IS/IT skills gap. Journal of Information Systems Education 17, 395–402.

Kitchenham, B., Budgen, D., Brereton, P., Woodall, P., 2005. An investigation of software engineering curricula. Journal of Systems and Software 74, 325–335.

Koong, K.S., Liu, L.C., Liu, X., 2002. A study of the demand for information technology professionals in selected Internet job portals. Journal of Information Systems Education 13, 21–28.

Kovacs, P.J., Davis, G.A., 2008. Determining critical skills and knowledge requirements of IT professionals by analyzing keywords in job postings. Issues in Information Systems 9, 95–100.

Lee, C.K., Han, H.J., 2008. Analysis of skills requirement for entry-level programmer/analysis in fortune 500 companies. Journal of Information Systems Education 19, 17–27.

Lethbridge, T.C., 2000. What knowledge is important to a software professional? Computer 33, 44–50.

Lethbridge, T.C., Díaz-Herrera, J., LeBlanc, R.J., Thompson, J.B., 2007. Improving Software Practice Through Education: Challenges and Future Trends. ICSE 2007. Future of Software Engineering Track.

Loftus, C., Thomas, L., Zander, C., 2011. Can graduating students design: revisited. In: Proceedings of the 42nd SIGCSE Technical Symposium on Computer Science Education, pp. 105–110.

Longo, G., 2006. Attracting top IT students. Declining CIS enrolment: an examination of pre-college factors. Information Systems Education Journal 4, 3–13.

Mead, N.R., Allen, J.H., Ardis, M., Hilburn, T.B., Kornecki, A.J., Linger, R., McDonald, J., 2010. (SEI) Software Assurance Curriculum Project. Volume I: Master of Software Assurance Reference Curriculum, CMU/SEI-2010-TR-005.

Prabhakar, B., Litecky, C.R., Arnett, K., 2005. IT skills in a tough job market. Communications of the ACM 48, 91–94.

Software Engineering, 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series.

Software Engineering 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering.

Standard Occupational Classification (SOC), 2000. http://www.statistics.gov.uk/methods_quality/ns_sec/downloads/SOC2000.doc.

Surakka, S., 2007. What subjects and skills are important for software developers? Communications of the ACM 50, 73–78.

Wu, J.H., Chen, Y.C., Chang, J., 2007. Critical IS professional activities and skills/knowledge: a perspective of IS managers. Computers in Human Behavior 23, 2945–2965.