Present and Future of the Informatics Profession
UPGRADE is the European Online Magazine for the Information Technology Professional, published bimonthly at http://www.upgrade-cepis.org/

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, http://www.cepis.org/) by Novática (http://www.ati.es/novatica/) and Informatik/Informatique (http://www.svifsi.ch/revue/)

Chief Editors
François Louis Nicolet, Zurich <nicolet@acm.org>
Rafael Fernández Calvo, Madrid <rfcalvo@ati.es>

Editorial Board
Peter Morrogh, CEPIS President
Fernando Sanjuán de la Rocha and Rafael Fernández Calvo, ATI
Prof. Carl August Zehnder and François Louis Nicolet, SVI/FSI

English Editors: Michael Hird, Alasdair MacLeod

Cover page designed by Antonio Crespo Foix, © ATI 2001

Layout: Pascale Schürmann

E-mail addresses for editorial correspondence:
<nicolet@acm.org> and <rfcalvo@ati.es>
E-mail address for advertising correspondence:
<novatica@ati.es>

Copyright
© Novática and Informatik/Informatique. All rights reserved. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, write to the editors.

The opinions expressed by the authors are their exclusive responsibility.

Present and Future of the Informatics Profession
Editors: Rafael Fernández Calvo and François Louis Nicolet

Joint issue with NOVÁTICA and INFORMATIK/INFORMATIQUE

2 Presentation: Reflection about Ourselves
   – Rafael Fernández Calvo and François Louis Nicolet

4 IT and Professionalism: An Industry View  – Peter Morrogh
   CEPIs is working on defining a base level of knowledge that all professionals must have and higher levels of knowledge that are specific to particular jobs.

6 Who Are We?  – Peter J. Denning
   This article presents a critical analysis of the characteristics that makes that a certain type of activity is considered a profession.

10 Let’s Design Everything Again: Thoughts on Computing and Its Teaching  – Ricardo Baeza-Yates
   The aim of this article is to give a critical and constructive analysis of the current state of affairs and the implications this has on education.

   A divorce exists between what universities offer and what society is demanding. The new socioeconomic climate is dragging universities into the realm of the “market” causing them to become oriented to “customer satisfaction”.

25 The Changed Professional Model of the IT Project Manager: From the Design of Technical Solutions to Social Mediation  – Beate Kuhnt and Andreas Huber
   Acceleration is the magic word: IT projects, which in former times took five years, must be executed today in a year. How does this acceleration affect project management?

30 What Must a Training in Information Technology Impart?  – Hans-Peter Hoidn
   Two main characteristics of a successful IT professional stand out: firstly a knowledge of basic concepts, and secondly practical experience and social skills.

32 Computers will Enforce Professionalisation  – Torsten Rothenwaldt
   The specialist in bits and bytes is now considered the driving force behind innovations, and the dearth of skilled IT staff is seen as an obstacle to economic growth.

35 The Future of Software Engineering  – Karol Frühauf
   Three issues stimulate my thoughts: the nature of software, the people creating software, and the essential factors in the creation of products that contain software.

37 The Future of Informatics as a Scientific Discipline  – Pedro G. Gonnet
   Informatics finds little or no acknowledgment as a science outside the university. On a long term this will lead to disastrous consequences for informatics as science and application.

39 The Shortfall of Informatics Professionals and the Legislation Regulating the Informatics Profession in Spain  – Rafael Fernández Calvo
   This article describes the shortfall situation concerning Informatics professionals in Europe and looks at the legislation regulating the legal practice of the Informatics profession in Spain.

Coming issue (October):
“Ubiquitous Computing”
Reflection about Ourselves

Rafael Fernández Calvo and François Louis Nicolet, Guest Editors

The aim of our survey and tutorial monographs is to convey our readers additional knowledge in their own professional speciality and an insight into other fields of informatics, so they can benefit from other professionals’ experience and know-how. The papers we publish are in general about technical subjects. But this time, we propose to reflect about ourselves: Who are we? We asked people from very different “classes” within our discipline to give their personal view about us informaticians and our status in society. We have asked them to comment about the present and look into the future “… We have asked them to comment about the present, and look into the future, being fully aware that our profession is too wide and has too many facets to pretend that most of them, let alone all, can be covered in a single issue of a magazine like ours. We publish the views and opinions of professors, of professional societies’ board members, of practitioners, and of a student. We do not expect you to agree with the very personal views of the authors – indeed, we would appreciate if you felt provoked by some of the opinions expressed. And we welcome your feedback.

One of the characteristics of a profession is the existence of a professional association the raison d’être of which is to actively promote and encourage the formation of a genuine profession, built on a sound and objective base. In our profession, there is a need for objective assessment of professionalism, says Peter Morrogh, President of the European Professional Informatics Societies (CEPIS): Most organisations are totally dependent on IT in order to run their businesses and cannot afford to entrust their IT to people’s care who lack the necessary knowledge and skills. CEPIS, with many of its member societies across Europe, is working on defining a base level of knowledge that all professional must have and higher levels of knowledge that are specific to particular professional jobs in industry. It links together not only the foundation knowledge that an individual must have but also vendor certification schemes, and third level education programmes. It will link into a continuous professional development framework.

Peter J. Denning, a past President of the Association for Computing Machinery (ACM), likewise observes that everyone has become dependent on IT professionals. “Who Are We?”, the title of his paper could have been the cover title of the present issue. His view is the one of a university professor and former president of the largest professional association. He defines the criteria for a profession and analyses how well IT meets them. Denning’s paper is from Communications of the ACM that has given its kind permission for republishing it.

Education is a key issue in a discipline like Informatics where frequent dramatic inventions and developments, both at the research and practical levels, are the norm. We have included two articles that cover this outstanding subject: from a global point of view Ricardo Baeza Yates gives a personal view of our field, providing a critical and constructive analysis of the current state of affairs and the implications this has on education, while Rafael Camps Parè, taking Spain as a case study, sets out to describe the divorce that exists between what universities offer and what society is demanding, and describes how the new socioeconomic climate is dragging universities into the realm of the “market” causing them to become oriented to “customer satisfaction”.

Kuhnt and Huber note that many IT projects fail because there are too many project leaders whose training has been very mechanistic-artisanal. They postulate that project leaders need to be trained in the so-called soft competence. We hope that the authors imply that IT project leaders are trained informaticians, for, in practice, we meet too many leaders of IT projects which have but a very rudimentary or no informatics education at all. The skills claimed by the authors can be only complementary to a solid background in informatics.

Hans-Peter Hoidn is a practitioner with an academic background. His experience is that IT training must convey the basic concepts as well as reinforce skills by providing the opportunity to learn an appropriate craft. Theoretical training does not impart experience, and a training that is primarily practical in content does not adequately promote the understanding of concepts. A training in IT must impart social skills. Their lack is often responsible for the failure of projects.

Institutionally, IT is firmly established as a profession, but on an intellectual level, the present dynamics are having an anti-professionalising effect, because due to the shortage of skilled staff, fewer and fewer computer solutions are being developed by trained specialists. This is the observation of Torsten Roth- enwald, an industry informatician, in his provocative paper. He reflects on the situation of the industry IT professional.

Karol Frühauf postulates “Software is the future currency of the world” in order to reflect about the economic role of software, the required qualities of software, the required qualifications of software developers, and design principles. His view is that of the experienced practitioner with a philosophic vein.

Rafael Fernández Calvo writes about the shortfall of Informatics professionals in Europe and looks at the legislation regulating the legal practice of the Informatics profession in Spain.
by laws passed by the Autonomous Communities (regions) and its potential effects, openly negative in some cases, on this shortfall.

We also asked one who is preparing his professional future to convey his view about the future of our profession. Informatics student Pedro Gonnet formulates very straightforwardly his concerns about the position of computer scientists, the attitude of industry, and the future of computer science education in Universities.

The Editors

Rafael Fernández Calvo is a Law Graduate of the Universidad Complutense of Madrid, Graduate in Journalism of the Escuela Oficial de Periodismo (Official School of Journalism) of Madrid and Diploma in Management Skills from ESADE. After a long career in the world of Informatics as a specialist in Databases, Data Design and Strategic Consultancy for Information Systems in IBM, he is currently editor of Novática, the journal and magazine of ATI (Asociación de Técnicos de Informática – Association of Informatics Technicians), acting editor of this association’s website, co-editor of Upgrade and associate professor in the Faculty of Law/ICADE of the Universidad Pontificia Comillas, Madrid. He is a senior member of ATI, and was member of the Board of the association and the first president of its Madrid Chapter. <http://www.ati.es/GENTE/rfcalvoi.html>

François Louis Nicolet is editor of INFORMATIK/INFORMATIQUE, the professional magazine of the Swiss informatics societies, Co-Editor of UPGRADE, and vice president of CEPIS. He studied physics, then met his first computer, a CDC 6600, at CERN in 1966. Phases of his professional life have included operating systems, compiler development, process control and real time systems – from punched cards to Internet. His interests focused on software quality and reliability. He has been a founding member, and later chairman, of the Swiss Chapter of the ACM, and a founding member of the Swiss Informaticians Society (SI). <nicolet@acm.org>

The English Editors

Mike Andersson, David Cash, Tracey Darch, Nick Dunn, Laura Davies, Roger Harris, Michael Hird, Pat Moody, Brian Robson
**IT and Professionalism: An Industry View**

*Peter Morrogh*

Most organisations are totally dependent on IT in order to run their businesses. However, the basis for objective assessment of professionalism lacks many of the identifying features of other professions. CEPIS is working on defining a base level of knowledge that all professionals must have and higher levels of knowledge that are specific to particular jobs.

**Keywords:** assessment of professionalism, professional associations, continuous professional development, CEPIS, EPICS.

Most organisations are totally dependent on IT in order to run their businesses – whether it is in designing new products, serving customers, reporting performance, or whatever, IT provides both the opportunities and the constraints. If IT fails to deliver the service that the organisation and its customers demand, then the performance of the organisation can be impacted. Poor quality systems, particularly as they become web-enabled, pose a serious risk to the organisation. Never before has there been such a need for a genuine IT Profession.

However, the basis for using the term “IT Profession” is, at best, shaky and it lacks many of the identifying features of other professions, such as success with professional exams, a period of apprenticeship, a process for keeping continuously up-to-date, a mechanism for withdrawing the “professional” mark for those who do not maintain the necessary standards and so on. The truth of the matter is that many people working in the industry do not feel the need for objective assessment of professionalism when there are so many jobs available and too few individuals to fill them. To varying degrees, computer societies do their best to encourage professionalism, but are hardly able to enforce it; even if they could, the withdrawal of professional recognition would be unlikely to have an impact on an individual’s job opportunities: in most countries across Europe, less than 10% of potential “professionals” feel the need to be a member of their national IT association. Likewise, very few advertisements mention membership of a professional organisation as being essential, let alone desirable. So not only do most individual IT practitioners see no reason to be a member of a professional organisation, but the majority of employers have a similar point of view. This situation is dire and needs to change: the impact of IT in all walks of life is pervasive and fundamental; national associations and organisations like the Council of European Professional Informatics Societies (CEPIS) must actively promote and encourage the formation of a genuine profession, built on a sound and objective base.

The concept of professionalism very definitely exists, but it is far from clear what jobs in informatics warrant the “professional” tag. A broad range of different skills are required in order to provide a business with an efficient and effective IT service which meets an organisation’s internal and external requirements. The list of jobs is very varied and is becoming increasingly diverse: it covers design, infrastructure, software development, support, communications, software testing, security, project management and so on. In the worst case, failure of any one of these elements can cause a disruption to a business, with the potential for consequential loss of customers and revenue. The risks have increased with organisations making their systems available over the Internet and World Wide Web. Therefore, from the organisation’s point of view, IT is “professional” only in so far as it is effective and delivers an excellent service to the business. Unreliable software, security failures, delayed projects, cost over-runs and so on are the mark of an unprofessional IT service. The delivery of an excellent service involves many different kinds of skills and not all of them warrant the term “professional”, even though they are fundamental to the provision of that service.

Traditionally, software engineers, software designers, developers, database experts and managers have been inclined to restrict the term to their own community and few would deny that these people should be professionals. This is far too narrow a view: with the increasing complexity of technology solutions, the diversity of technical/business people who contribute towards an organisation’s IT systems and services is broadening. There are common features that identify professionals: these are a mixture of business, technical and personal/interpersonal education, knowledge, skills and experience. There is a tendency amongst practitioners to focus on technical skills; the other attributes are as important, or more important.

*Peter Morrogh* is IT Development and Quality Manager in the Insurance and Investments Division of Bank of Ireland, based in Dublin. He has a very broad range of experience as an IT manager and consultant, in Europe and the Middle East. He is the current President of the Council of European Professional Informatics Societies (CEPIS) and has served CEPIS in a number of other capacities. He is a past Chairman of the Irish Computer Society (1989 to 1997). He was also one of the founding Board members of the European Computer Driving Licence Foundation ECDL). <pmorrogh@indigo.ie>
This thesis is supported by a small survey (60 participating organisations, consisting of user organisations, software companies and IT services organisations) carried out by the Irish Computer Society in late 1994, organisations were asked to allocate 10 points across 3 headings (business, technical and personal/interpersonal) for 3 generic job descriptions. The results were:

<table>
<thead>
<tr>
<th></th>
<th>Business</th>
<th>Technical</th>
<th>Personal/Interpersonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Leader</td>
<td>4.0</td>
<td>2.9</td>
<td>3.1</td>
</tr>
<tr>
<td>Systems Analyst</td>
<td>3.9</td>
<td>3.6</td>
<td>2.5</td>
</tr>
<tr>
<td>Programmer</td>
<td>2.2</td>
<td>5.7</td>
<td>2.1</td>
</tr>
</tbody>
</table>

The job categories are restricted (and now look a bit dated). But, the message is clear: technical skills are of varying importance and are only useful in combination with other skills. Therefore, to determine professionalism solely on the basis of technical skills, gives a very dangerous slant to the term “IT professional”. Unfortunately this approach is supported by a number of world-wide vendor certification schemes, which tag on the word “professional” to their certificates. Undoubtedly such qualifications are excellent at a technical level; however, they do not necessarily warrant such recognition.

The average industrial IT manager (including those who work in software houses and IT services organisations) is inadequately equipped to determine whether somebody is, or is not, a professional, given the enormous diversity of activities that the IT function is expected to perform. Tools such as the current version of the British Computer Society’s Industry Structure Model (ISM) have made an important contribution to providing an industry framework. The ISM embraces not only the IT skills that the individual requires but also the experience, responsibilities and softer personal/interpersonal characteristics that are needed.

CEPIS, with many of its member societies across Europe, is working on defining a base level of knowledge that all professionals must have and higher levels of knowledge that are specific to particular professional jobs in industry. The working title for this certification scheme is “EPICS” (European Professional Informatics Certification Services). This scheme is interesting because it links together not only the foundation knowledge that an individual must have but also vendor certification schemes, third level education programmes and so on, which are essential for an individual to become a specialist. It is planned to start running pilots late in 2001 or early 2002. This will be a significant step towards improving IT professionalism and enabling a new route into the IT profession. It is hoped that EPICS, in due course, will link into a continuous professional development framework. CEPIS and its member societies are in an unique position to undertake the EPICS work because of their broad experience with the European Computer Driving Licence, which has been an outstanding success in Europe (over one million participants at the end of 2000) and, increasingly, internationally.

CEPIS has recognised the importance of continuous professional development for many years. In 1996/1997, CEPIS developed the European Informatics Continuous Learning (EICL) programme, which has contributed towards the British Computer Society’s Continuous Professional Development (CPD) scheme (note the correct use of the word “professional” by the BCS). The EICL/BCS CPD programmes both recognise that professional development is not just technical but also business and personal/interpersonal. Organisations will increasingly adopt schemes such as the EICL or CPD in order to encourage professional practice and keep participants up-to-date, as part of their on-going career development. There is a key role for computer societies in this area; their independence and not-for-profit status, will gain welcome recognition from the majority of employers, who at present fight with a mish-mash of qualifications, courses and so on, with no overall direction or structure.

There are a few skills that every IT manager looks for in his/her employees who have basic IT aptitude: the ability to think logically and creatively, to take ownership for work and to able to resolve problems. The traditional route into IT involves the study of Computer Science, or some similar degree programme; there is an equally relevant route into the industry for those who have a first degree in a non-IT discipline (such as Mathematics, Physics, Engineering, Arts etc.), followed by an IT post graduate degree or diploma. Individuals with a broad education can often have a more innovative approach to solving problems and coming up with ingenious but robust solutions. Third level educational establishments play an important role for encouraging creative thinking amongst new entrants into the IT profession; they are also an excellent resource for developing the on-going education needs of existing IT personnel.

It is difficult for an individual to be a professional unless the organisation that he or she works for adheres to professional practices and standards. Therefore, it is in the direct interest of organisations to encourage employees to develop their knowledge and skills, within a professional framework. As the IT industry matures, there must be greater partnerships between those who employ IT professionals and computer societies. Computer societies and organisations like CEPIS must seize the role that is theirs to take. Projects like EPICS are a step in the right direction in the long road to creating a genuine, recognised IT profession.
Who Are We?

Peter J. Denning

This article presents a critical analysis of the characteristics that make a certain type of activity considered a profession. These characteristics are compared with the present situation in the field of the Information Technologies to examine whether it meets the criteria for profession. The paper concludes with a summary of the forces forming an IT profession.

Keywords: IT professional, computer scientist, professional standards, professional societies, certification. Principles, Practices, Responsibility.

To most of the hundred millions of computer users around the world, the inner workings of a computer are an utter mystery. Opening the box holds as much attraction as lifting the hood of a car. Users expect information technology professionals to help them with their needs for designing, locating, retrieving, using, configuring, programming, maintaining, and understanding computers, networks, applications, and digital objects. They expect academic computer science to educate and train computing professionals, to be familiar with the changing technologies, and to maintain research programs that contribute to these ends. Students aspiring to be professionals look to faculty for a comprehensive, up-to-date view of a world with many fragments, for making sense of rapidly changing technologies, for assistance in framing and answering important questions, and for training in effective professional practices.

In short, everyone has become dependent on IT professionals as much as on the information technologies themselves.

Who are the IT professionals? They are a much larger and more diverse group than computer scientists and engineers. They have organized themselves into professional groups in three categories covering at least 40 specialities (see the accompanying table). The first category comprises the major technical areas of IT and spans the intellectual core of the field. The second category comprises other well established fields that are intensive users of IT; although their practitioners focus on issues in their own fields, they draw heavily on IT and often make novel IT contributions. The third category comprises areas of skill and practice necessary to support the IT infrastructures that everyone uses.

But wait! Are we entitled to use the term “profession” in connection with IT? What are the hallmarks of a profession? Do we meet them?

What Makes a Profession?

Today, most people understand computer science as a discipline that studies the phenomena surrounding computers. These phenomena include design of computers and computational processes, representations of information objects and their transformations, hardware, software, efficiency, and machine intelligence. In Europe, the discipline is called “informatics” and in the U.S. “the discipline of computing.” The computing profession is understood as the set of people who make their livelihood by working with computing technologies.

But “making a livelihood” is a narrow view of profession. If you examine other professions, you see four hallmarks:

• A durable domain of human concerns.
• A codified body of principles (conceptual knowledge).
• A codified body of practices (embodied knowledge including competence).
• Standards for competence, ethics, and practice.

A profession includes institutions for preserving the knowledge and the practices, enforcing the standards, and educating professionals. Health care, law, and libraries are three prominent examples that illustrate these principles.

Health care. Health is a permanent concern of all human beings. Breakdowns in health are inevitable because of disease, accident, or aging. Health-care professionals take care of people’s concerns in health. Hospitals, HMOs, insurance companies, government health programs, the national medical association, the medical colleges, and medical schools are the principal institutions of this profession. Their curricula and certification programs codify their conceptual and professional knowledge. Doctors must be licensed to practice medicine and can obtain certificates testifying to higher levels of competence in various specialities. Doctors who violate professional standards are subject to reprimand or censure by the national medical associations, malpractice lawsuits, and loss of license.

Peter Denning is a past President of ACM, the Director of the ACM IT Profession Initiative and recipient of twelve awards for invention, teaching, and professional service. <http://www.cs.gmu.edu/faculty/denning.html>

* This article is republished from Communications of the ACM, February 2001/Vol. 44, No. 2, by kind permission of the ACM.
Law. The rule of law is a permanent concern of most human beings. Most people live in societies with governments, constitutions, legislatures, and laws. Implementing agreements and carrying out actions without violating laws or incurring penalties is an ongoing concern for them. Breakdowns are inevitable because people do break laws and because many business practices are governed by contracts. Two allied professions help people deal with their concerns and recurrent breakdowns about laws: the legal profession (lawyers, judges) and the law enforcement profession (police, other law enforcement agents). Law schools, police academies, legislatures, courts, and the national legal and police associations are the principal institutions of these professions. Lawyers must pass a bar examination and be licensed to practice law. Lawyers who violate professional standards are subject to reprimand or censure by the legal association, malpractice suits, and loss of license. Similarly, police are trained rigorously and are subject to sanctions.

Libraries. The preservation and sharing of recorded human knowledge is a durable concern of many human beings. Progress in technology, law, commerce, politics, literature, and many other aspects of civilization depends on access to knowledge created by our ancestors. Civilizations can be interrupted or lost when they lose access to their own historical documents and records. The profession of library science helps people deal with these concerns by preserving documents, making them available publicly, cataloguing and organizing them, and preserving them. Curricula in library science preserve and transmit bodies of conceptual and professional knowledge about libraries. Libraries, schools of library science, and library associations are the principal institutions of this profession. Librarians must earn certain credentials to practice the profession and are subject to reprimand or censure by their professional associations.

How IT Stacks Up
How well does IT meet the four criteria for profession? The durability criterion is dearly met: computation and effective communication are ongoing concerns and sources of breakdowns for all human beings. Ours is a world of information and numbers, many processed by machines and transmitted by networks. The telephone and fax are ubiquitous, the Internet soon will be, and databases are springing up like weeds everywhere in the Internet—all technologies that extend the distance and time over which people can successfully coordinate actions. Nearly everyone in every developed country is affected by digital telecommunications; leaders in underdeveloped countries are aggressively installing informational infrastructures to accelerate their countries’ entries into world markets. In the same way, computation is an integral part of the daily practices of finance, engineering, design, science, and technology. Word processing, accounting, databases, design automation, and report-writing software impact every other profession. The digital world offers many new kinds of breakdowns, ranging from failures of computers and communications, to software bugs, to the challenge to install software that improves an organization’s productivity. In other words, the concerns are not phenomena that surround computers. It is the other way around. The computers surround the concerns.

The body of principles criterion is clearly met. Our conceptual knowledge is codified in the curricula of our degree and training programs and is often proposed or endorsed by professional societies.

The body of practices criterion is not met—yet. Few university programs define criteria for different levels of professional competence and test people. Professional societies do not do this. The Institute for Certification of Computer Professionals (ICCP) does it in a narrow area, but is not widely known or used. The growing interest in state licensing of software engineers, however, is prompting the professional societies to examine the certification of skills of IT professionals. And the growing interest in forming IT colleges is prompting academic leaders to include professional knowledge in the curriculum.

The professional responsibility criterion (ethics, standard practice) is partially met. The professional societies (mainly ACM and IEEE) have codes of ethics but do not enforce them. We have yet to develop criteria of competence and to ask our colleges and universities to certify their graduates. There are all too many discontent users, a signal that we are not listening carefully to them and serving them.

So IT meets several of the criteria and is moving toward meeting the remaining criteria in the next decade.

In thinking about the IT profession, we need to distinguish crafts, trades, disciplines, and profession [Holmes 00]. A craft is a set of practices shared by a community of practitioners but does not enjoy a recognized social status. A trade is an organized group of practitioners (some may be craftsmen) with restrictions imposed by society in return for freedom to practice...
the trade for the benefit of society. A discipline is a well-defined field of study and practice. A profession may include many disciplines, several trades, and many crafts. It embodies a core value of listening to its clients and for being socially responsible.

The U.S. Department of Education, acting under a Congressional mandate, has defined a profession as a set of people who have at least two years of post baccalaureate education and whose field is on an approved list. This definition is much narrower than the one offered earlier.

The Identity of an IT Profession

What do people think you care about? What actions do you take? Are your words and actions aligned? What standards do you adhere to? What do you offer them? How do you interact with them? What breakdowns do you resolve? Create? The answers to these questions manifest an organization’s identity in the world.

The current identity of our field is a mixed bag. We are seen as passionate innovators and prolific inventors. We are seen as nerdy—single-mindedly focused on IT and inept with social relationships. We are seen as technology-centred, not human- or user-centred. We are seen as oblivious to the social, political, and business consequences of our tools and services. We are seen as avoiding responsibility for malfunctions of our tools and breakdowns in our services. We are seen as difficult to communicate with.

The challenge facing professional societies is to promote a new identity characteristic of a profession. We must articulate standards for professional members and organizations. We can promote the adoption of those standards by enlisting industry leaders to do so in their organizations and by giving visibility to organizations, companies, schools, and government that exemplify those standards.

Through cooperative action, professional societies can address large issues facing the IT field that not one of the specialties addresses adequately on its own. Principal among these are:

- Developing, codifying, and teaching methods for designing and implementing safe and reliable software systems, especially the large systems in critical applications. This is seen by many as the central mission of our field.
- Reforming IT education to bring more coherence, endowing it with a common core, and incorporating professional practices alongside conceptual principles.
- Providing programs for the ongoing (lifelong) education of IT professionals.
- Providing standards for professional competence at multiple levels in each specialty and certifying individuals who meet those standards.
- Learning to connect with and be responsive to customers, clients, and users.

The Role of Computing

It is an irony that the discipline that gave birth to the IT profession is not the driving force in the profession. Computer scientists are the inventors and visionaries, but the field is being driven by the large numbers of pragmatists who are the users of the field and include many powerful business, civic, government, and industry leaders. Computer scientists need to come to grips with the fact they are no longer in control of the field. They do not call the shots. Their research is not the driving force behind most IT innovations. They are one of many professional groups in the field. What role can they play?

I believe the natural role for computer scientists, consistent with their history as the progenitors, is the custodian of the intellectual and scientific core of the field. This is an important role that must be filled by someone if the IT profession is to achieve coherence. But it will not come automatically. It will come if computer scientists learn to embrace commercial applications, interactions with other fields, and the concerns of their customers. This may be a chasm too wide for many computer scientists to cross [Denning 98].

In recent years there have been many tensions between computer scientists and software engineers. Software engineers claim to be an engineering discipline and want out of computer science departments; for their part, computer science departments do not want to lose an important segment of their field. Many of these tensions arise, in my view, from a failure to distinguish the body of principles from the body of practices. Software engineering and computer science share the same conceptual principles, but they have significantly different professional approaches and practices. They can share curricula in the principles, but they need to go separate ways with professional practices. With the interpretation of the IT profession offered here, both are part of the profession but there is no requirement that one be part of the other.

I have been troubled in recent years by the skirmishing between software engineers and computer scientists, by the insularity of many computer scientists, and by the question of coping (in education) with the large demand from pragmatists for help. Somehow we have to adapt, take leadership, but give up our traditional feeling of control over the shape of the discipline. My conclusion is that we need to think in terms of profession rather than discipline, for there appear to be many disciplines that want to be part of the profession. Once we come together as a profession, we will be better able to accomplish the goals that no one of us can do alone.

A Profession of Information Technology

Here is a summary of the forces forming an IT profession:

- Most of those who use computers and communications do so through hardware, software, and networks whose inner workings are mysteries to them.
- People in business and their clients, people at home, people in science and technology, and people depending on large software systems have concerns about the design and operation of reliable hardware, software, and network systems to help them do their work.
- These people seek professional help in taking care of their concerns. They expect computing professionals to be responsive, competent, ethical, and able to anticipate future breakdowns.
- The Profession of IT is coming into existence to provide that help. It currently has a negative identity, but professional
societies can define and promote a strong professional identity.

- The education of IT professionals must account for practices as well as descriptive knowledge. It must include training as well as general education. It may not reside in any single university department, being distributed among computer science, software engineering, computational science, computer engineering, and related departments such as astronomy, physics, chemistry, biology, management science, linguistics, or psychology—each of which contributes important specialities to the profession.

- Individual IT professionals should embrace boundaries between their specialities and others in the profession. As a whole, the IT profession must embrace its boundaries with other fields to assure a constant stream of life-giving innovations.

Through its research, the Profession of IT must anticipate future breakdowns that others will encounter. A close interaction between computer researchers and others is essential so that the questions under investigation remain connected to real concerns, both short and long term. Otherwise computing research can drift into irrelevance and cease to earn public support.

References
[Denning 98]

[Holmes 00]
Holmes, N. Fashioning a foundation for the computing profession. IEEE Computer (July 2000), 97–98.

©2001 ACM 0002-0782/01/0200 $5.00
Let’s Design Everything Again: Thoughts on Computing and Its Teaching

Ricardo Baeza-Yates

“Man is the only animal to trip over the same stone twice”
“El hombre es el único animal capaz de tropezar dos veces en la misma piedra”

Spanish popular saying.

“Everything is highly intertwined”

Ted Nelson, inventor of Xanadú.

The aim of this article is to give a personal view of our field; a critical and constructive analysis of the current state of affairs and the implications this has on education. Although this view is rooted in a local context, most of the issues considered are also of relevance in a global context.

Keywords: Training in Computing, Software Engineering, Databases, Interfaces, Public Software

1 Introduction and Motivation

The first problem we have with our field is what name to give it. Are we talking about Computer Science and/or Computer Engineering or Informatics? Is the correct term computer, computation, computational or informatics? Where do information systems fit in or should they be considered as a separate area? I still do not have any clear answers. Perhaps the problem is intrinsic. As the joke goes: Computer Science has two problems: Computer and Science. Have you ever heard of the science of washing machines or any other machine? Do mathematics or physics need to say they are sciences? In short it’s a problem of coming of age, of maturity, and consequently of insecurity. In any case it is clear that what we do is rooted both in engineering and basic science. Before we go on it is necessary to make something clear. Many of the things I say here are self-evident or common sense. Nevertheless, despite being obvious, many of them have not been said by anybody else and that’s why I am saying them here. Is it that they are very obvious or that they only become obvious once you know them?

The ultimate aim of any software is to transmit some knowledge to the mind of the person using it and vice versa. The biggest bottleneck occurs at interfaces, at the final point of communication with the user, and the problem lies not only in bandwidth but also in the very way information is represented (Figure 1). As Dijkstra said recently, we have still not been able to rise the challenge of making large software systems less complex.

How to eliminate this bottleneck is the main aim of this article, which may seem like a hotchpotch of apparently unconnected arguments. However, we often forget to analyse our universe as a whole, from the point of view of an outside observer. For anything we wish to study, form and content should be given their due degree of importance and consideration. I will start by presenting an analysis of the relationship between technology and culture, and an analysis of computing.

Fig. 1: Communication of Information.

Ricardo Baeza-Yates is a Doctor (Ph.D.) in Computer Science (University of Waterloo, Canada), Magister in Electrical Engineering and Computer Sciences from the University of Chile and Electrical Civil Engineer by the same university. He is currently a Tenured Professor in the Department of Computer Science of the University of Chile and his fields of research are algorithms, information retrieval and visualization. He is the author of several books. He has twice been president of the Chilean CS Society (SCCC). Currently, among other posts, he is president of the CLEI (Centro Latinoamericano de Estudios en Informática) and is the international coordinator for the subprogram of IT and applied electronics of CYTED (Iberoamerican Cooperation Programme on Science and Technology). During the year 2000 he set up a search engine for the Chilean Web <http://www.todocl.cl>. <rbaeza@dcc.uchile.cl>
itself and in the context of our profession. I will go on to talk about some of the implications on education of this analysis and I will put forward some basic ideas regarding what to teach and how to teach it. In short, my message is that we should not forget the many kinds and many levels of relationships that exist, that we should constantly test the hypotheses we make and that we should really redesign and not merely reengineer.

2 Technology and Society

We are moving in a highly technological world so it is important to understand the way technology and our society interact. The relationship between technology and culture is one of love and hate, of successes and failures, of visionaries and monopolies.

How long does technology need to reach all levels of society? In many cases not many years, from a historian’s point of view. For example, printing took a hundred years to reach all of Europe. However, for someone living at that time this was considerably longer than an average lifespan. The telephone or commercial aviation took more than 30 years to have any impact on a significant percentage of the population. The fax was invented last century but has only had any real impact on society in the last couple of decades and it is still not present in most homes. To quote Norman: “Today we often hear that the pace of change has speeded up, that changes happen in “Internet time”, in months or weeks, not decades or years. False.” [Norman 98]. The Internet has been with us for more than 30 years and it is still not in every home, even in developed countries.

History is packed with examples of innovative or high quality technologies which flopped. Here a few: Edison invented the phonograph in 1877 and in spite of this his company failed; nobody has heard of the first car company in the United States (Duryea); the Macintosh operating system was much superior to DOS, but it lost the commercial battle; Sony’s Beta technology was better than VHS; etc.

One of the reasons for this failure is an inability to understand what the customer really wants. Logic does not always defeat the whims of the market. In the case of Edison, the problem was their choice of artists for their records. The public wanted to hear the best known singers. It didn’t matter if there were others equally as good or better, it was the name that counted. In the case of a product, again quoting Norman: “…it only matters that what is being offered is good enough for the purpose. Moreover, if you lead the marketplace in sales, it is permissible to use a nonstandard infrastructure. After all, if you have the majority of customers, then what you do becomes the standard. Your competitors have little choice but to follow. If you are not the leader, then having nonstandard infrastructure is a bad idea. Ultimately, it leads to extinction” [Norman 98]. For software there are many examples of this kind.

2.1 Accidental Products, Damaged Products

Sometimes bad products are sold as part of a marketing strategy to break new ground and get a name known. Speed is the important thing, not whether a product actually works or not. For this reason haste is the most important impediment to quality. Often technological quality, a good design, user-friendliness (or the contrary), are equally unimportant. To quote Norman: “What kind of world is this, anyway, where horrible products don’t matter?” It’s our world and the only kind we have.

“Buy! The only 100% compatible 64 bit RISC computer, Posix compatible operating system and with total connectivity, even ATM. The solution for this world of open systems. Plus the software that you need free: object oriented development tools with graphic intelligent interface, transactional database and SQL server with support for 99 known or future formats and 64 more utilities.” Beware! As with other products of this consumer market of ours, the reality we find is different from what the advertising claims (although there are always exceptions to prove the rule).

Most of the most popular and influential technologies in computing nowadays were never meant to be used as they are used today, nor were they meant to dominate the market the way they do. The success of MS-DOS/Windows, Unix/Linux, several programming languages and the World Wide Web are proof of this. By this I don’t mean they are good or bad, merely that they were not designed for what they have become today. Those of you who know the history of DOS and CP/M will know what I am talking about, or how prototypes like X-Windows or Mosaic have changed the course of computing history. These and other cases like them are clear examples of accidental products and systems that have ended up dominating our world while at the same time making us less sure of our ability to shape the future of technology.

According to Karl Reed this should be a task for the professionals of computing, who should not only report on new advances or problems but also steer the development and use of information technologies [Reed 98]. According to Reed this has happened due to an aversion to planning. I prefer to think that what is happening is that our planning is not successful either because of the environment I am about to describe, or because of a lack of time.

Would you accept a car with an engine you had to switch off and start up again in order to make it work (people tell this as a joke but it isn’t really so funny) or a television that gave you an electric shock every now and then? Of course you wouldn’t, but that’s what we accept in software. In March 1999 Microsoft recognized in a private meeting with its distributors that five thousand errors in Windows’95 had been corrected in Windows’98 (but they didn’t say how many new ones had been added!). That is to say that millions of defective software copies were sold at an irreparable cost to their customers. In short, what we need are serious, high quality software companies if we want to get back on course again. This also means changing marketing policies and not accepting with half finished products.

Sadly, better solutions take time and many companies are reluctant to invest in development and/or research if technology changes every year. That is the current paradox, and while it is understandable it is also damaging. Today, instead of thinking, we are using familiar solutions that were no use 20 years ago. It’s true that it’s not a good idea to re-invent the wheel, but
Neither is it a good idea to have no time to invent anything. There will come a time when it will be pointless to invest in new technological advances if we cannot use them properly. When this time comes we will have to go back to the drawing board and think. Yes, think; something that the pace of the modern world let’s us do very little of, to the point where we are forgetting how to do it.

2.2 Implications
The use of technology is conditioned to a great extent by historical accidents and cultural variables as much as by the nature of the technology itself. Social, cultural and organizational aspects of technology are much more complex than mere technical aspects. After a technology has been established, it becomes entrenched and it is very difficult to make any changes. We should try to bear this in mind for the future, since in our field there are many examples which we will be looking at later on, the most obvious one being the Windows operating system.

To close, let us remember that technological advances are of no use unless society advances alongside them. In fact some studies show that productivity has not actually increased, in spite of the degree of computerization the world has experienced [Brynjolfsson/Hitt 98], [Dewan/Kraemer 98].

3 Our Technological Environment
Technology advances so quickly that it leaves no time for thinking or using it to design efficient solutions. I would now like to show how many of the solutions we use (and consequently their designs) are based on premises that are no longer valid; and how solutions which existed in the past are now coming back into their own. However, in the long run the technological race is self-defeating. But let us start with a little history.

3.1 A Little History
Things have been rediscovered so many times. You would think that Windows had discovered graphic interfaces if you didn’t know the history of Xerox Parc and then Apple. Others believe that RISC technology was invented by IBM with its line of RS-6000 equipment, unaware that it was developed in the mid 70s.

Let us make a brief analysis of the development of computing in recent years. Many technologies have advanced exponentially. This is the case of Moore’s famous law that states that the capacity of microprocessors doubles every 18 months. This prediction, made in 1965, is still proving to be true [Hamilton 99]. The same can be said of the memory chip capacity per dollar which has increased by a factor of 134 million in the last 40 years. Recently a similar rate of growth can be seen with the Internet. The number of computers connected doubles every 15 months. This cannot continue indefinitely since today more than 20% of the world’s computers are already connected and it would mean there would be as many computers as people by the year 2010. The growth of the Web is even more impressive. Since 1993 the number of servers doubles every 3 months, and today the figure stands at over 30 million. Similarly the United State’s Internet capacity increased by 1,000 times in the 80s and possibly by that much in the 90s. In spite of this, traffic on the net is growing at an even faster rate.

If we compare a present day personal computer with a typical one 20 years ago, we can see that the storage capacity has increased more than 1,000 fold and the processing capacity by at least 150 times. These drastic differences in growth rates lead to problems. For example, improvements in speed on the networks (several Gb per second is being forecast) are difficult to exploit since processors are not as fast. Other technologies have not developed at the same pace, such as disk transfer rates, which have increased by much less, and are now as input/output one of computing’s bottlenecks.

As far as we users are concerned we haven’t even doubled our capacity and yet sometimes I am amazed at how easy it is to get used to something bigger (as one of the interpretations of Murphy’s law states: however big your disk is it will always be nearly full). The same can be said of software, which hasn’t undergone any spectacular advances either; you could even go as far as to say that methods haven’t changed much in the last 10 years. While it is true that many computational resources are cheap, the solution does not lie in using the design that we already had, but without optimising it and asking the user to buy a computer twice as big and twice as fast.

The most important features of present day computing are largely dictated by the impact of the Internet. Among them we can mention interactivity, distributed information and processing, digitalization and the use of multiple media, use of shared resources and collaborative systems, standardization and open systems. It is difficult to make predictions; many have got it wrong in the past. The most famous examples are from IBM’s founder, Thomas Watson, in 1943: “I think there is a market for five computers”; and from Digital’s founder, Kenneth Olsen, in 1977: “There is no reason why a person would want to have a computer in their home”. Among short term speculations are: the mass adoption of optical fibre, the development of wireless networks, the convergence of PCs with Unix workstations, the greater use of collaborative tools and of course the total massification of computers and the Internet.

3.2. Operating Systems and Networks
Most premises upon which traditional operating systems were founded are no longer valid. In the past hardware resources (CPU, memory, disk) were very expensive so their use was limited. This led to many solutions being unnecessarily complicated, in order to reduce their cost or the impact they had on shared reduces.

These premises changed in the 80s and while costs went down, the speed of processors and size of memories increased by more than 100 times. Why then aren’t operating systems at least 100 times faster? To adapt existing solutions at first improvements were used in the interfaces (for example the cache memory). However the complexity of the solution itself imposed a maximum limit to these improvements. The solution is to simplify the solution. Enter the paradigm “simplicity equals speed”. One of the corollaries of this paradigm has been the shift from CISC processors to RISC ones.
Also history is forgotten. Windows 1.0 was never sold, Windows 2.0 was a failure and only Windows 3.1 was a success, being no more than a good patch on DOS. Windows NT needs a minimum of 16Mb and 32Mb is recommended. What happened to the 64K that DOS needed? Is memory so cheap that we can afford to forget about being efficient? Are processors so fast now that we can forget about good data structures and algorithms? Defenders of Windows NT will say that it is much more than DOS, that it includes a system of windows, network connectivity, multiprocessing, etc. All right, but for example Linux with X-Windows works with 4Mb and better with 8Mb. Why then does Windows NT need so many resources? Clearly there is a design problem somewhere. The rise in mobile computing may help to improve designs in this particular area, since with laptops we cannot afford the luxury of having a lot of resources or of using a lot of energy (battery).

A similar phenomenon has occurred in networks. In the past they were expensive and slow. Now they are cheap and fast. Most current technologies have had to adapt to the changes, although there is still a lot more to be done. For instance, ATM was designed in the 60s and is now coming back onto the scene, because it is simple and fast, reaching speeds of 155 Mb/s. However that is still a long way from speeds of several Gb/s that can be reached with optical fibre.

Another example is X-Windows, the most popular Unix windows system, which is transparent to the network protocol used. That is to say, it is a distributed windows system. The communication protocol used by X-Windows assumes that the network is fast and the screen graphics are slow. However nowadays that is not true, because while networks are fast, they are congested and shared by many users. And the speed of screen graphics has also increased.

3.3. The Art of Programming

Programming is perhaps the heart of Computer Science. It is the world of algorithms and data structures and programming paradigms. Throughout its evolution programming has been more an art than a science or engineering. It is not for nothing that Knuth’s famous trilogy on algorithms and data structures is called The Art of Computer Programming.

For many people programming is not an entirely respectable job; that’s what programmers are for. But we should make a distinction between people who are able to design the solution to a problem and turn it into a program and those who are only able to turn the solution into a program. A real programmer, as Yourdon would say, is someone who can carry out the whole process, from analysis to implementation.

Programming keeps you in training for solving problems whether they are big or small. Programming should be satisfying. It should in no way be demeaning for an engineer, or whatever we think we are, to program. Quite the contrary; it is we engineers who will often make the best programs because only we fully understand our own solutions. Another important point is that good code is not the least comprehensible or the most extravagant code, but the one which is the clearest, most efficient and best documented. Many people also equate being a keen programmer with being a hacker. As with any addiction, extremes are not wholesome. Neither should we see hackers as evil programmers. There are good hackers and bad hackers, and the former are indispensable.

3.4. Software Engineering?

In 1999, an important executive of a major US computer company told me: “We could afford to do it well because we had the resources and we wanted to break into a new market”. Of course at a technical level we would always like to do things well, but the market is telling us otherwise. There is no time, there are no resources, it’s now or never. The result is badly designed and poorly tested products. These days the only company in the market that could afford to do things well is Microsoft. But they don’t seem to want to.

Perhaps the best place to start is with the famous year 2000 glitch, or the millennium bug (although really this century started in the year 2001). Whichever way you look at it, this was a ridiculous problem which had a massive impact. Should this embarrass us? I don’t think so.

Was it a mistake to consider only two digits instead of four for the date? Everybody knows that the main reason was to use less memory, a resource which 20 years ago was much more expensive than it is now. I think it was neither a mistake nor was it good design. The real reason is that none of the designers thought that their software would still be in use after more than 20 years. Not even today do we think that, plagued as we are with annual changes of hardware. It is true that in some instances programs have developed without the original design having changed, but this is not the normal case. Why do we go on using that software? Because of the bad software development habits we mention below.

Computing changes, but that does not mean that it improves. Many companies may prefer not to change software which we know works or which we know where it doesn’t work. This software survives successive changes of hardware until in many cases it loses its original source code as a result. Other companies have tried to change it, but their projects have failed due to not using the right methodologies and/or tools. There again, today we can see the other extreme. There is an excessive use of resources and the design is of secondary importance. For example, Windows’98 has more than twice the number of lines of code than the latest version of Solaris and occupies much more memory when running. The reader can draw his or her own conclusions as to which operating system is better designed, leaving aside the fact that the more lines of code there are the more chance there is that there will be errors. Just because memory today is cheaper doesn’t mean we should overuse it.

Why does this happen? Let’s draw a parallel with civil engineering. Can you imagine a bridge being built that falls down five times while under construction due to design faults? Unthinkable. Worse still, can you imagine that, at the very moment of opening that same bridge with 100 people on it, you find a fatal error in its design? Impossible. However, everyone in programming uses trial and error techniques. Now consider the number of designers. A house is designed by one to three architects. What would happen if there were dozens? And when
a house is being built you don’t make major changes to its design. How many times do software implementers change the design? Plenty, partly because often they are the same people and having two roles without separating them clearly is always a problem. We used to talk a lot about reusability, but it is only now, with class libraries and design patterns, that this word has any real meaning. In the past it was difficult to make use of what had been done by other people for countless reasons; code not available, different language or environment, lack of documentation, etc. Modularity and component independence is vital if we want to integrate different products and technologies. We can also talk about quality. If we add reuse and using the right control tools, in the future we may be talking about real software engineering [IEEE 98]. Although I am aware that others might shoot me down here, I would say that software engineering is actually software handcraft. TeX is perhaps the best example, since in the beginning it was the work of an excellent craftsman, Don Knuth, and for the last 10 years not one error has been found in its code (and the final cost of each single error grows exponentially). Unless we change our way of thinking and stop relying on always being able to test, and that if there are errors it doesn’t matter, programming will continue to be an art in which few will be masters and most will always be apprentices. This change will need to be radical, since even the biggest software companies are still not in a position to say that their product has no error. The following examples from Windows illustrate this point.

Windows’95 contained nearly 15 million lines of code. Applying Caper Jones’ estimates [Jones 96], a code of this size has a potential number of errors of nearly 3 million, which gives an idea of how many tests need doing. To get this figure down to five thousand requires at least 18 repeat tests [Lewis 98a]. Although software companies should perhaps perform more tests this would raise the cost and delay the products release on the market.

Sadly history shows that bringing out new versions quickly often means a more successful product. This happens because customers do not base their choice on quality, though this is less true of critical products such as a Web server. Here quality is more important which is why the Apache server wins over a Unix type operating system, although it is public domain software. Many companies say that they don’t use public domain software because it doesn’t have support. But most PC products, especially Windows, don’t have support either. Windows NT has around 25 million lines of code, which means more tests should be carried out to ensure required levels of reliability. Moreover Windows NT is supposed to be certified at security level C2 for use on Internet. However a study carried out by Shake Communications Pty. Ltd. revealed 104 problems, some of them very serious ones, which make it vulnerable to hackers [Lewis 98b].

In the case of software, suppositions similar to those regarding operating systems were made: expensive and limited resources. Now resources are cheap and plentiful. But it is also bad to misuse resources by writing software needing large amounts of memory or a lot of available space on the disk. This is acceptable only when it is really necessary, and on most occasions this is not the case. This is another side effect of not having enough time to design software and of producing it in order to bring out new releases as quickly as possible, because that’s what the market is demanding. This abuse of technology has a harmful effect. For instance, if we want to do something faster, the most common solution is to buy a faster computer. However it is cheaper and possibly faster to use a better solution (better software, better parameter adjustment, better network configuration, etc.).

3.5. Artificial Intelligence?

Artificial Intelligence is one of the areas of computing which promised most and has progressed the least. Whether it be in games like chess or processing of natural speech, results have shown that good heuristics, black boxes or neural networks are only partially effective. But we are still a long way from the Turing Test. Let me use chess to help me put across my ideas. In May 1997, Gary Kasparov, then world chess champion, was beaten by Deep Blue from IBM (Big Blue), the champion of chess programs. Has the machine triumphed? By analysing this pseudo-victory of artificial intelligence over man perhaps we can put an end to the abuse of terms like expert or intelligent systems. Isn’t a good algorithm intelligent? Is brute force intelligent?

At the beginning of the 50s it was predicted that in 20 years there would be programs capable of defeating the world chess champion. More than twice that time was needed for that to happen. So does this make computer programs intelligent then? No, Deep Blue doesn’t think like a person (neither does it think, but let’s just say it does something similar, for comparison’s sake). Kasparov knows what lines of play to analyse and he studies just a few moves in depth. On the other hand, Deep Blue analyses millions of moves and appraises a large number of positions, but it can do it faster. The fundamental difference lies in intuition, creativity and long term strategy. If Deep Blue had the ability to assess positions like Kasparov does, it would be invincible. However, Deep Blue evaluates a position on the basis of heuristics. That is, rules that work most of the time but other times don’t.

The more complex the game is and the longer term the objective is, the more difficult it is to analyse any given position. For example, for some time the best checkers program has been better than any human. Why? Because there are far fewer possible positions in checkers and the rules are much simpler, which allows it to assess every possible move. On the other hand, in the oriental game of Go it is necessary to control the board little by little, without knowing until the end if many of the pieces are still alive or not. This makes it more difficult to analyse, because long term strategy is required. In this case intuition and experience are much more important than memory (as in the game of bridge) or an ability to make rapid calculations (as in checkers).

The first misinterpretation we can make of Deep Blue’s win is that it may seem as if computer has defeated man. What has actually happened is that a group of experts in computing and chess have programmed a high powered computer and have succeeded in defeating the world champion. That is to say, a
group of people who have worked over a long period, concentrating especially on working out how to defeat the champion, have had more success than the intelligence and memory of one man working alone. I don’t consider it such a big thing that a program can beat one person, since it’s an unfair fight. Deep Blue has a large number of processors, it knows more than a million games by heart and can analyse 200 million positions a second. It would be an interesting experiment to see if with less time per match its calculating capacity might be less important. Could Deep Blue defeat a group of grand masters? I doubt it.

There are also factors which have nothing to do with intelligence that affect a chess player’s concentration. According to some chess players, Kasparov had a great deal of respect for Deep Blue. Others say that he took his role as defender of mankind very seriously and that his defeat would be a milestone in history. And of course Kasparov is a human being, with emotions, who needs to eat, drink and sleep, and who feels the pressure of knowing that he cannot exert any psychological pressure of his own on his opponent. An opponent who neither makes mistakes nor gets tired. If we look back, one of the reasons for all the successful defences of his title was Kasparov’s greater psychological strength.

Man defeats himself every day. Kasparov was defeated in public. That’s all. When a computer can read a book, understand it and explain it, that will be something to shout about. Deep Blue is an example of software engineering, of a good program in a world where not many are to be found. A program that has been improved over many years, that uses knowledge from many sources and that has had time to evolve. If we made use of technology as Deep Blue does, we would be in a better world for sure.

3.6. Interfaces with Common Sense

Because of the limitations of the original Macintosh which, in order to keep its cost down, couldn’t run two applications simultaneously, (very different from its powerful predecessors: Altos and Lisa), Macintosh’s desktop metaphor was not centred on documents. The user was therefore required to choose an application and then choose a document, rather than selecting the document first and then the application to use it with. Though on the surface this looks like the same thing, it meant a radical difference in the development of interfaces. Only for the last few years has it been possible to select a document and run a predefined application or choose one from a menu. To quote Bruce Tognazzini, one of Macintosh’s designers: “We have come to accept that the way to create or edit a document is to open that document inside an application, or tool. This is equivalent to having to slide your entire house inside a hammer before you can hang a picture on the wall or hiving to put your teeth inside your toothbrush before you can brush them” (from the essay Nehru Jacket Computers in [Tognazzini 96]).

Let’s take a look at present day interfaces. The information we store is based on a hierarchy of files and directories in which we navigate from father to son and vice versa. That is to say in just one dimension. Not only that, but we have to remember where each file we create is and what name we gave it (not to mention limitations of length, symbols, or not being able to use identical names). Also, while the screen is a bi-dimensional space, the interface rarely makes use of this fact neither does it learn how we use it nor what order we do things in. For example, I might be moving a file right across the screen to put it in the waste bin and at the last moment my finger slips. Result: two icons end up one on top of the other. The interface might have assumed that what I was trying to do was to get rid of the file! In my opinion, part of the success of browsers and the model imposed by HTML lies in the fact that, as well as being a very simple interface, it has a single level link structure. New paradigms of visual representation of knowledge are already appearing [Greenberg 99].

The computer technology we use should be transparent for the user. In fact how many newbie users only use one directory to put all the files they use in? The user doesn’t need to know that there are directories or files. Besides not everything can be classified into directories and files. A file should be able to belong to two or more different classifications which should be able to change as time passes. How we understand things depends on our place in time and space. Our surroundings are not static, but the computer unnecessarily forces us to keep our documents immovable in space and time.

Let’s give this some serious thought. The computer should – and can – name and group together files and retrieve them using their content or the values of an attribute. For example, you might say: show all the letters I was editing yesterday, get the first lines of each letter, then choose the line I need. Another baseless premise is that we need a common interface for everybody. People are different, they think and work in different ways. Why not have interfaces which adapt to each user, which can be personalized and which learn the way and the order we do things in? To pave the way for the implementation of new interfaces, we should scrap the past and replace file systems with data organized in a more flexible and powerful way [Baeza-Yates et al. 99]. This leads us to our next subject.

3.7. Databases

One of the biggest problems of current databases is the large number of different models, although the relational model is the predominant one. However new applications need data which is not so structured and rigid: multimedia, hierarchical objects, etc. While suitable models do exist for these types of data, there are no tools which allow us to integrate well two or more models. In fact attempts to incorporate these extensions to the relational models have not been very successful.

If we forget past hypotheses, we may be looking at more powerful and flexible models. An example is the case of objects with dynamic attributes [Baeza-Yates et al. 99]. In this model the objects have a dynamic number of attributes, the values of which have type and are also dynamic. This model can be considered as an extension of the classless object model. However it is also a powerful query language that can handle object sets that satisfy arbitrary attribute conditions, including their non existence or if they have an undefined value.

There are many arguments in favour of this model: simplicity, flexibility and uniformity; the elimination of suppositions
4 Our Professional Environment

In addition to those problems directly caused by questions of a technological nature, there are also problems related to the market and the professional environment. For example, professional overspecialization, the lack of good software project managers or the scant interaction between theory and practice in software development [Glass 99]. We will be taking a closer look at some of these aspects in the following sections.

4.1 Complaints from Industry

The most commonly heard complaints from industry are that much of what is taught is of no use. Industry is saying that what is needed is for students to get more practical knowledge and skills, so they can apply what they have learnt to a business context and thus make their entry into the real world of industry so much easier. That what is needed are software engineers, not scientists [IEEE 97]. The first thing I would like to say is that all that is true, but it all depends on your point of view. Commercial objectives are short term while university objectives are long term. Other more specific complaints include the lack of industrial patents developed in universities and the absence of entrepreneurial innovation.

What a company wants is a young person with experience. The lack of practical knowledge is difficult to remedy in a system in which the technology changes so quickly. That is why concepts are so important, since they give the ability to adapt and learn. It is true that often companies lose the investment they make in training, but that is normal in a highly competitive market. Specialization (for instance, specific tools) and continuous training are the employer’s responsibility, not the university’s. However one of the main problems is that by investing in training the employer may lose the employee when he or she gets a better paid job on account of being better trained. This often happens as a result of the employer not giving enough importance to the investment they have made.

Finally, we come to the commercial aspect. I believe this problem goes beyond computer engineers; it is a question of the interaction between technology and society. We cannot have know-alls who are also good sales people with an understanding of business. These skills are often innate and cannot be taught (I often feel like I am trying to teach common sense, and the results are not too heartening). There are already shortcomings in the technical curricula given the present volume of different subjects there are in computing which cannot all be satisfactorily covered.

4.2 University-company Relations

Joint research by universities and companies has always been bogged down by various factors. These include the slow administrative apparatus of universities and the companies view, often justifiable, that universities are incapable of achieving short term goals. We need to develop an infrastructure for applied research and to increase technological transfer, which is what many developing countries really need to export software and maintain growth in this area.

Another way to provide an incentive for applied research would be to create research projects in which it would be obligatory to have some industrial counterpart. These projects need not only be for applied research but can also be for basic research, though with lower budgets and smaller working groups. This would allow specific problems to be tackled and would encourage support from companies since the risks would be smaller.

We should also be bringing universities and companies closer together in a way that is beneficial for both parties. Technological transfer, exchange programs and the like are ideas that have been mooted thousands of times already. The fact that universities register no patents is criticized. The same criticism could be levelled at Chilean software companies. First of all it is very time consuming. Secondly it is not cheap (at least 10,000 USD). Thirdly, in the course of the process the result cannot be published (which flies in the face of the current system of academic assessment based on publications, although this is changing in Europe). And point number four, ideas should not be patentable (for example, an algorithm).

4.3 Monoposoft vs. Open Source

The Open Source movement (that is, free source code) is gaining momentum every day and is beginning to attract media attention as a result. The classic example is Linux: Would its creator ever have imagined that it would be used now by millions of people? Meanwhile, Microsoft is fighting with the US federal government and their software is a source of money and jokes [Lewis 98a], [Lewis 98b]. Sadly many of these jokes should make us cry rather than laugh. But these jokes conceal important truths and lead some people to fight against windmills as romantic Don Quixotes.

How can it be that not only is there such a thing as free software, but its source code is public too? This doesn’t make any sense in a capitalist market, where it would be hard to imagine asking thousands of programmers to work for nothing. My personal opinion is that Open Source only exists because Microsoft exists. If we have to choose between cheap software and Microsoft software, for many different reasons we are bound to choose the latter. But if the alternative is free we would be willing to take a risk and try that software. Also, though it may seem to be a contradiction, freeware may be better than commercial software. If someone finds an error and reports it, in a matter of minutes you will be able to go to an Internet news group and get a correction for the problem. And if not, a lot of programmers will take a look at the code and one of them will spot where the problem is. This inefficient mechanism is nevertheless highly effective.
Another advantage is that the process is scalable: as the code grows in size more people can get involved in its development. A few years ago a Microsoft internal document that talked about the danger that Open Source represented for the company was leaked onto the Internet (see this and other related subjects in [Sanders 98], [IEEE 99], [Lewis 99a], [Lewis 99b]).

Microsoft is a de facto monopoly. Every two or three years, millions of users have to upgrade their copy of Windows. They don’t get complete compatibility with older versions but they do pay prices that keep abreast of the times: a captive market must be exploited. It’s like having to regularly move house but not always to a better one. In 1999 Bill Gates published his 12 rules for the effective use of Internet in companies [Gates 99], which show that he was utterly converted to the world of electronic mail. He has also learned the advantages of free software, particularly when it also allows him to blow away the competition: Explorer. From an economic point of view software development in Microsoft is not the most effective (in fact it is the users who find most of the problems, and they can’t always get direct help to solve them), but it is definitely the most efficient. Microsoft is perhaps the only company that is capable of stopping this snowball. It could even afford to take 5 years to develop a real operating system and applications with much better interfaces, as described before. But this is not going to happen, because it would mean earning less. Depending on the result of the anti-trust case, in which Microsoft has recently had a favourable but not definitive sentence, and on the advance of public code, this millennium will be the information age or the Microsoft age.

5 Final Comments

I would like to begin by quoting Peter Freeman [Freeman 97]: “If we compromise the core of computing science, we risk losing long-term foundational skills. If we fail to take into account the concerns of the computing professional, we risk becoming obsolete. The key is to achieve the right balance – but there is more than one way to get there”.

We have to concern ourselves with both form and content. If we can create good professionals they will become agents of change [Garlan et al. 97]. That should be one of the main aims of university and I feel that it has also been a major personal driving force for what I do.

Most of what we learn in our lives is of little use, it’s mostly technical knowledge. The important thing is the training associated with that learning process, the development of logical and analytical capabilities, the ability to abstract, conceptualize and solve problems. The objective is not knowledge per se, it is personal development. It is to learn and learn constantly. I believe that there is a better way to do this, by integrating knowledge and new tools in innovative courses in which the student has a greater understanding of the final goal. The main aims should be flexibility, adaptability, to put emphasis on concepts and to facilitate continuous learning.

All sciences have evolved in a real world context, not just in isolation, the origin of calculus being perhaps the most classic example of this. In the past there were people who knew most of what there was to know in terms of scientific knowledge, but nowadays this is very difficult, which forces us into group or multidisciplinary work. These two facts should help us to consider new ways of teaching. Two different lines of action present themselves. Either designing different professionals, based on the three elements involved: people, processes, and technology [IEEE 97], for example, an information architect [Baeza-Yates/Nussbaum 99]; or drastically changing the way we teach by integrating all the suggested classic contents [ACM] into one single problem solving based course [Baeza-Yates 00].

Although these proposals are of a preliminary nature, I think that they are a first step towards designing better, more complete and coherent curricula, and teaching them in a different way, by motivating students and giving them clear explanations of why they are learning each subject and how these subjects are related to the world they live in. In short, integrating everything, in a certain manner returning to the Renaissance, to encyclopedic and enlightened thought. It is also clear that we have to encourage critical thinking and lay great emphasis on matters of design.

This article is at the same time an essay on the many problems surrounding our field and a quiet appeal for sense. In our private lives as in our professional lives, we accept so many things as true, as basic hypotheses that we never question. Similarly, the ideas expressed here should be taken only as one more point of view to be considered. However, I do hope that these lines appeal to your common sense, that sense which is so important and at the same time in such short supply, and in passing create a little awareness of the multiple problems surrounding our field and our daily task. This is a constructive criticism and is in no way intended to be divisive [Glass 99].

Acknowledgements and Notes

I am grateful for comments and encouragement from Omar Alonso, Juan Álvarez, Karin Becker, Tania Bedrax-Waiss, Carlos Castillo, Helena Fernández, Terry Jones, Miguel Nussbaum, Greg Rawlins and Jorge Vidart. This article is an updated and abridged version of a paper that dates from 1999, whose full content, in Spanish, is available at <http://www.baeza.cl/manifest/manifest.html>.

References

[ACM]


[Baeza-Yates 00]


[Baeza-Yates et al. 99]


[Baeza-Yates/Nussbaum 99]

[Brynjolfsson/Hitt 98]

[Denning 97]

[Dewan/Kraemer 98]

[Freeman 97]

[Garlan et al. 97]

[Gates 99]

[Glass 99]

[Greenberg 99]

[Hamilton 99]

[IEEE 98]

[IEEE 99]

[Jones 96]

[Lewis 98a]

[Lewis 98b]

[Lewis 99a]

[Lewis 99b]

[Norman 98]

[Reed 98]

[Sanders 98]

[Tognazzini 96]
Bruce Tognazzini: Tog on Software Design, Addison Wesley, 1996

*English translation by Steve Turpin*
Can Universities Give what Society Is Asking of Them?

Rafael Camps Paré

This article, that takes Spain as a case study, sets out to describe the divorce that exists between what universities offer and what society is demanding, and describes how the new socioeconomic climate is dragging universities into the realm of the “market” causing them to become oriented to “customer satisfaction”. After showing how it is by no means clear that the non-university occupational training will be a success, the article proposes that universities should readjust their focus more towards providing the Informatics professional profiles that society needs.

Keywords: Informatics Engineering Courses, Occupational Training, University Offering, Society Demand, Market Orientation.

1 Criticisms and Self-criticisms

What do Informatics engineering graduates, and the companies and institutions that employ them, think about the quality and suitability of their qualifications? Is society satisfied with the Informatics graduates that universities produce?

In Spain we scarcely have any studies or surveys to give us reliable answers to these questions. The only works I know of are F. Sanchís’s survey [Sanchís et al. 97] of EU/UPM (Escuela Universitaria de Informática / Universidad Politécnica de Madrid) graduates, and a trial carried out in 1997 (but never published) for a future survey of FIB/UPC (Facultad de Informática de Barcelona / Universidad Politécnica de Catalunya) graduates. Both works refer to graduates from former education plans. Consequently their results can only hint at the current situation, but they are all we have to go on. The most noteworthy conclusions they have in common are:

• A general call for a less theoretical and more practical training
• Most graduates consulted work in development of business applications, generally as programmers
• Very few gave a positive appraisal of the suitability of their studies to the labour market
• A large percentage of them think that their degree is under-valued or not valued at all in their professional environment
• What they feel to be most lacking are subjects relevant to what is going on in companies.

At present we don’t have objective enough data on what companies think: we will need to resort to a personal view. My own view, arrived at after thirty three years of professional experience in companies, thirteen years as a university educator and several round tables and debates on the subject, is summarized in the following paragraphs.

Companies specializing in Informatics tools (for example major hardware and software suppliers) and the large user companies, judge the training received by Informatics engineers to be of a sufficiently high quality, because, although the graduates may not know about the tools and practices in common use, they learn quickly and adapt to new technologies with a greater ease than the traditional Informatics professionals who only had experience behind them, but little training.

But these companies also see that this facility for learning is not exclusive to Informatics graduates but is shared by, say, telecommunications or electronic engineering graduates. They find that it is precisely the non-Informatics people who are most willing to adapt (with no sense of superiority) to the company’s professional culture. The biggest defect these large companies see in recent Informatics graduates (in fact in graduates of any technical degree) is the enormous lack of ability regarding oral and written expression.

However for some years now most jobs for Informatics professionals have not been in these large technology user or supplier companies, but rather in service or outsourcing companies, and in the huge number of small and medium size companies with little Informatics culture. All these companies are looking for instant results and ask for knowledge of systems and tools that Informatics graduates tend not to have. Bear in mind that more than 95% of the knowledge required in the Informatics job offers that appear in the press is completely unknown territory for the Informatics graduates from most universities.

The lack of convergence between what is taught in universities and what society is asking occurs to a greater or lesser extent in all courses and in all countries. In western societies, criticism of the culture currently imparted in universities is...
gaining momentum, and they are being asked to bring the culture in line with the demands of the outside world.

A few years ago it began to be apparent that the level of investment in R&D in universities was not keeping up with technological innovation. Also, every day there is less support for the idea that the academic community should be the judge of the quality and relevance of the knowledge it produces. All this is putting the current university model into question. There is a growing feeling that if universities follow this course they could isolate themselves from present day society. According to J.M. Bricall, president of the European Conference of University Rectors, “the problems of financing public universities is symptomatic of the rift between [universities] and the society that upholds them” [Bricall 97].

In the Informatics field, society’s dissatisfaction is particularly strong. The circumstances surrounding Informatics’s birth and early years produced a rift between the academic world and the business world far greater than that which exists in those disciplines which reached maturity many years ago.

In Informatics, whether we like it or not, we follow in the wake of the USA. They exert the greatest influence on us and have the most experience in Informatics teaching. That is why, before discussing the needs of the Spanish Informatics environment, we should see how things are in the matter of “society satisfaction” in the USA. There should be some useful conclusions to be drawn which are applicable to our situation.

The controversy concerning the need to bring university Informatics teaching in line with society’s needs is already some thirty years old, but during this last decade criticisms have been getting stronger and louder. As Robert L. Glass has said, in Communications of ACM, “the communication chasm that exists […] between the university world and the industrial world, is getting bigger not smaller” [Glass 97].

As long ago as 1992 the US National Research Committee stated “…given the nation’s pressing economic and social needs and the changing environment for both companies and universities, the committee believes that the CS&E (Computer Science & Engineering) academic world must widen its perspective if it is not to become less and less relevant to the Informatics profession.”

Peter J. Denning, the great guru of university teaching of Computer Science, author in 1989 of the highly renowned work “Computing as a Discipline”, the inspiration behind Curriculum ACM/IEEE-91 and former president of the ACM, has been gradually moving his standpoint closer to the professional world, and in 1992, in a long and interesting article entitled “Educating a new engineer” [Denning 92] was already saying things like this:

“University education is experiencing a huge failure. There is a growing number of students, companies, educators, company executives, management specialists, civil servants and tax payers who are voicing their dissatisfaction with the teaching and research of the majority of our universities. […] Much of the research going on in universities bears no relation to the interests of business people or the interests of the people in general […]. The educators use links with industry to get more funding for their research projects, but are not open to discussing their study plans”.

And in 1999 the same professor Denning at the presentation of the ACM award “Outstanding Contributions to Computer Science Education” called upon university Informatics teachers to “cross the chasm that separates the academic world from the world of professionals, unless we want to find ourselves sidelined” [Denning 99].

From the enormous amount of criticism and self criticism expressed in these last few years in the USA I will just quote one typical example. The teachers of the Dept. of Computer Science of the University of Virginia, in a work [Knight et al. 94] in which they were explaining how they were making a complete change in their study plans, said: “The skills that an Informatics professional needs nowadays is the antithesis of what we are teaching”. And what they were teaching was what is being taught in nearly all universities, what is recommended in ACM’s curricula.

We have just seen the type of criticism that is being levelled at Computer Science and engineering teaching (and research) at universities in the USA. Given the influence that Computer Science departments from USA have on universities all over the world, it is not surprising that very similar criticisms can be heard everywhere. Including, of course, in Spain. It should not be forgotten that here, in the Informatics field, …

a) study plans tend to be inspired by those in the USA
b) universities are, on the whole, more removed from the business world
c) unlike the USA, we are not an Informatics tools producer; we are very much a user
d) university departments find themselves obliged to carry out their research along the same lines as the international academic community if they want to maintain the academic prestige demanded of them. But in the Informatics world and especially in countries like ours, following this path makes it difficult to get collaboration and backing from the business sector.
e) innovation in Informatics comes to companies only via suppliers (generally multinational companies).

Some Spanish university departments are aware of the problem. For example the LSI/UPC department (Dept. of Informatics Languages and Systems, Universitat Politècnica de Catalunya) in its strategic plan, remarks that its research activity and its postgraduate teaching have “little to do with industry” and fears “a reduction in funding if there is an increase in the trend towards applicability seen in the National R&D Plan” and in the 4th Spanish R & D Framework Program.

The professor F. Sáez Vacas said in a highly interesting work [Sáez Vacas 92]: “We tend to follow uncritically whatever is laid down by computing curricula drawn up by prestigious American associations. This is a mistake. The problem could be defined as the need to strengthen flows and links between the Informatics-Science subculture and the Informatics-Use subculture. I think that such a need is generic and universal, although it is specially pressing in countries whose unavoidable technological modernization has to combine with an optimum use of scant economic resources. In contrast to this idea,
one can observe that a major part of the teaching and research carried out in our country is following a different course, imitating more developed countries with regard to subjects and focuses. If we just analyse the social results a little, we can see they are little short of disastrous [...]. Such ways of working normally have one guiding criterion, a certain slavish following of the saying publish or perish, which American students are already parodying as faculty publish, students perish.”

2 A New Socio-economic Environment

The current socio-economic environment places Spanish universities in an extremely difficult situation that could be outlined in the following terms:

• In recent years there has been a considerable increase in the number of universities (both public and private). For instance, in the academic year 1997–98, 15% more places were offered in Catalonia than were applied for. In the immediate future, information technologies (basically the World Wide Web) will globalize the market so that the user will be able to choose from a very wide offer of education.

• The drop in birth rate is beginning to make itself felt now in the demand for places. In Catalonia in the last 6 years university place applications have fallen by 20%.

• The reduction in public funding for the universities does not appear to be a temporary measure. Experts assure us that self funding will become more and more necessary [Bricall 97]. As we will see later, there is an additional risk for the future of some Informatics Technical Engineering courses in the form of increased competition in the “middle level” occupational training market, stemming from the starting up of the High Level Technician qualification, TS (Técnico Superior) the new non-university courses.

Therefore, as J.M. Bricall repeatedly explains, universities are going to have to move in a highly competitive market. The management of some Spanish universities have lately been showing a special concern about graduates’ professional skills, about collaboration with the business world and about service to society. For example, UPC’s strategic plan talks repeatedly about responding to the needs and demands of society.

In the new socio-economic framework, this kind of statements have connotations or interpretations of a harsh neo-liberal reality, of a cut-throat market logic, hitherto absolutely unprecedented in the university world. You might now hear, for example, a university manager making the following statement, “We have to capture students by differentiating ourselves from the competition and offering them an attractive product. Otherwise we could find ourselves without students”.

It should be noted that Spain is not an isolated case. With some specific variations, all the countries around us are facing a similar problem. And in the case of Informatics the problem is aggravated by the rift with the professional world.

3 A Little Bit of History: Why a Bachelor’s Degree in Informatics?

Here in Spain, in the second half of the 70s, the main drivers and creators of Informatics studies set about creating some courses with an Engineering orientation. Other groups, Physicists and Mathematicians, wanted to create Informatics oriented specialities within their Faculties. But the social/university inertia, conflicts of interests between corporate pressure groups, and the political situation, made diversification of Informatics studies impossible in the universities. In 1976 one single university degree was created, with the name Bachelor’s degree in Informatics (Licenciatura en Informática), to be taught in Faculties.

In truth the adoption of the term “Bachelor’s degree” was imposed by the political circumstances of the day. Legally a new Bachelor’s degree could be created by Decree, but a new Engineering degree (awarded by Higher Engineering Schools) had to be created by Law. The difficult circumstances at that time (General Franco had just died and the Spanish Transition period was beginning) made it necessary to take the Decree option. This in turn made it necessary to take the unprecedented step of creating Faculties within the Polytechnic Universities (for example the Informatics Faculties of the Madrid and Catalonia Polytechnics). However the idea of the promoters and the faculty was to train “engineers”, even though circumstances obliged them to be called Bachelor’s graduates.

Here in Spain the first study plans were very Computer Science, a little Computer Engineering and very little Business Management Information Systems. With time the semantic burden of the term “Bachelor degree”, acquired over years of history, proved too great an influence on some people.

Therefore in 1976 it was impossible to have a “distribution” of studies related to Informatics. In my view it should have been done in the following way [Camps Paré 76] (my opinion has barely altered since then):

a) Introduction of basic concepts of use, application and implications of information technology for each field in all courses (medicine, hotel management, sociology, fine arts, etc.). In the particular case of Business Schools, creation of Information Systems type specialities oriented towards business.

b) Creation of Information Systems type specialities in engineering schools. Especially in middle level Schools of Technical Engineering.

c) Creation of Computer Engineering type specialities, both in Hardware and Software, in higher or middle level Schools of Engineering.

f) Creation of Computer Science type specialities in Faculties of Science.

e) Higher Level Occupational Training (Formación Profesional Superior in the Spanish official terminology), outside of the university environment, in at least the following two specialities: Business Applications Development and Computer Systems Administration.

So, twenty three years ago I was warning that most of the country’s professionals need training of high level occupational type and “If we don’t make occupational training work and be valued by companies, young people will go on masse to the faculties […]. The country only needs a small number of graduates from those new faculties”.

© Novática and Informatik/Informatique
4 High Level Occupational Training

Since the middle of the 70s Spanish society has been waiting for a Higher Level Occupational Training (Formación Profesional Superior) in Informatics to appear. But it took twenty years for the qualification of Higher Level Technician (Técnico Superior, TS) to be created. Finally, in 1996 and 1997 (in Catalonia) the study plans were published for the two Informatics specialities at TS level, which are: “Business Applications Development” and “Computer Systems Administration”.

These are enormously detailed study plans, with a mix of concepts, tools and practical work that seems to be in line with what most Spanish companies are looking for and need. The length of these courses is 2,000 hours (of which 350 are work experience). It is interesting to note that the length of the Informatics Technical Engineering courses is not very different (In EUPVG – Escola Universitaria Politècnica de Vilanova i la Geltrú, a Catalan middle level School of Engineering, it is 2,100 hours).

Let’s take a look at the major study area of the speciality “Higher Level Technician (TS)” in “Business Applications Development”: Operating systems, LANs, Analysis and design of databases, Analysis and design of applications, Structured and modular programming, Advanced programming, Database management systems, 4th generation environments and CASE tools, Design and implementation of presentation services, Workplace relations, Professional orientation and coaching, Synthesis (project development). 170 hours are left to the centre’s discretion.

The teaching staff for these courses in the public sector are technical teachers from professional training and secondary education teachers. But some private centres have begun to contract Informatics Technical Engineers, graduates with some experience in companies for this work.

5 What do we want from our universities, what do they give and what can they give?

Most of the companies are still unaware of the existence of “Higher Level Technician (TS)” or else they know that they hardly exist. That is why they look for the “Informatics professional” profile they need among Informatics Technical Engineers and on discovering that these engineers don’t have the profile they are after, they level their complaints at the universities.

If the day ever comes when TS courses are taught properly, with the right resources (teaching staff and material) and a sufficient number of centres, then without a doubt in a few years the Informatics Technical Engineering degrees might lose a large percentage of their students. But right now the public system of occupational training (TS courses), does not have, at least for the moment, the right resources. Only some private centres, very few, are able to give that training with an acceptable level of quality. Many centres are only equipped to teach the use of some tools. Few are able to teach concepts and criteria. The teaching of tools, in spite of what many with little Informatics culture unfortunately believe, is not enough to get Informatics technicians of good quality.

Meanwhile, what service companies, some programs that style themselves “master”, etc. tend to be teaching in Spain, usually consists of training oriented solely towards the immediate use of specific tools. The lack of a broader vision, the total lack of basic concepts, make this education completely short-sighted. The idea behind the TS plan is for students to also get a certain basic conceptual vision, but just how effectively this plan can be carried out (bearing in mind such factors as availability of competent and trained teachers, hundreds of centres of widely varying quality) remains to be seen.

Could the universities supply the thousands of Informatics professionals with a TS type profile of good quality which Spanish society has been asking the education system for during the last twenty years and has not been getting? If we were to significantly increase the “professional orientation” of Informatics Technical Engineering courses, without losing their conceptual and analytical base, then we would doubtlessly produce Engineers who would be valued by companies and would have a significantly higher quality profile than the TS (Higher Level Technicians). It is in this area, the training of the technicians of good quality that society needs, that universities have their great opportunity.

Many universities are in a position to give the right mix of academic rigor, abstraction, design and real world practicality. The conceptual strength of their Informatics departments could give the training in concepts needed for the professional to adapt (understand, appraise and decide) in the fast changing world of Informatics, Information Systems and companies. Universities should also give the right purpose-oriented training for the students’ entry into the labour market. And also they could supply a high degree of creativity, critical thought and flexibility.

Faced with the evidence that most of their graduates are working as programmers, the compilers of the Escuela Universitaria de Informática / Universidad Politécnica de Madrid (EUI/UPM) survey quoted in point 1 [Sanchís et al. 97] ask themselves “what is EUI/UPM’s function, to train programmers or to train broader based professionals?”. When it has been said that lately all newly graduates in Informatics find work it is not mentioned that this work tends to be as a programmer and not a very highly skilled programmer at that, nor that the boom is basically a passing one (taking the opportunity afforded by the Millennium Bug and the coming of the Euro to rebuild applications).

Also to be considered is that many other non-Informatics graduates are taken on for these same jobs. Spanish society avails itself of both short and long cycle degrees to cover its need for professionals that, in principle, do not need a “university” education. And there is every sign that it will continue to do so in the future.

It was twenty-two years ago that Informatics studies were set up in Spain. Today, as then, what most companies need and expect from the education system are professionals with a profile of the kind envisaged for the TS. But as these don’t exist (or scarcely do) they contract university graduates in Informatics or other specialities. And they don’t contract people with low level Occupational Training (FP2) because they have a tremen-
dously low level of training. University graduates (in Informatics, Telecom, Math, Industrial Engineering) tend not to have any knowledge of the specific Informatics techniques and tools used by companies, but have a base that allows them to learn ad adapt with ease, which is not the case of the FP2 graduates. Not only that but the FP2 people don’t usually have any great knowledge of tools either. The extremely low salary level for recent graduates is also a factor that influences companies’ choice.

When the Informatics Faculties were designed (1972–1976) some people considered that their Computer Science & Engineering type of orientation was the right one to have because they thought that:

a) Though we had missed the hardware production boat, we could still become a power in software;
b) Old middle level technical degrees were being done away with as they were being absorbed into the universities, but the training of the vast majority of the very large number of professionals which Informatics user companies would need, would not fall to the universities, whose traditional mission was to train “the educational elite”, but to Higher Level Occupational Training (TS), which was then beginning to be planned.

But time has shown us that:

a) Due to the inadequate structuring of the Informatics sector, we have been unable to be more than mere consumers of tools produced in other countries;
b) We have been unable to produce Higher Level Occupational Training to meet companies’ needs. Those who want to learn the “Informatics profession” and the companies looking for “Informatics professionals” will continue to turn to universities;
c) Universities have become “democratized-massified”. Countries like Spain, in which 40% of young people go to university, cannot afford the luxury of not giving professional Informatics training there.

Twenty-three years ago now, in the presentation of the proposal for the first curriculum for FIB/UPC (Facultad de Informática de Barcelona / Universitat Politècnica de Catalunya), R. Puigjaner and M. Vergés said [Puigjaner/ Vergés 75]: “[The creation of] Informatics teaching at a Occupational Training level, strikes us as highly interesting, since the greatest proportion of jobs on offer in the Informatics sector fall within that level …]. It would be a grave mistake to push the corresponding nucleus of students into university education”.

And in 1980, M. Martí and R. Puigjaner [Martí Recober/Puigjaner Trepat 80], the first two deans of FIB/UPC, insisted on an “unresolved and urgent” issue, which was “to start up studies quickly at the level of middle level Schools of Technical Engineering or high level Occupational Training”. Note the use of the conjunction “or”, which seems to a certain extent to be putting the two qualifications at the same level. The planners themselves had hopes that the Informatics Technical Engineering courses would be of a decidedly practical/professional nature and had doubts about the starting up of the Higher Level Occupational Training courses.

Therefore it should not surprise us that society turns to the technical engineering Degrees when looking for occupational training, nor that it complains when it cannot find it. And we university teachers cannot hide away in an ivory tower and say that it’s not our problem. It is our problem too. And it’s also an opportunity for the future which we should be careful not to miss.

6 Short Cycle and Long Cycle Engineering Courses

The old middle level technical courses, the old “Industrial Technicians” (electrical, mechanical, chemical, etc.) were absorbed into the university creating the present day “Technical Engineers”. The need to make a clear difference between Higher Engineering and Technical Engineering courses, and the simplification sought with the system of cycles, meant that the length of Technical Engineering courses was progressively reduced, from 5,400 hours (6 years) that they had in the plan 57, to the current 2,100/2,250. And as each centre has the possibility of reducing the lecture hours of a credit by a further 30%, it could even be cut to 1,550 hours (less than for the TS courses).

Those Technical Engineering degrees with tradition from the time of the “Industrial Technicians” have managed, in spite of the enormous reduction in hours, to keep up their profile, their identity or their own professional objectives. The universities did not want to “demean themselves” with that “low level” profile: they felt it would be filled by the TS, and that to a certain extent the Technical Engineering courses would gradually disappear. It is no surprise then, that some Informatics Technical Engineering degrees have basically the same contents as the first cycle of long cycle Engineering degrees.

In practically all western countries there are two different types of technological courses, some focused on R&D and others with a more practical and applied orientation. The former are longer than the latter whose duration tends to be between four and four and a half years. But the fundamental difference between the two types of courses does not lie in their length but in the different focuses of their curricula.

In France, after high school graduation, the student has various paths open to him or her. From the prestigious studies of the Grandes Écoles, to the Bac Technicien Supérieur (which could be positioned between an FP2 and a TS) or the Diplôme d’Études Approfondies, a five year university course, (in the fourth you get the Maîtrise degree, as for example the MIAGE, Maîtrise Informatique Appliquée à la Gestion d’Entreprise) and the Diplôme Universitaire en Technologie (which is more practical than Spain’s Technical Engineers but has a higher level than our TS).

Another, less well-known example is Germany. There are two types of higher centres of education there in which you can study engineering: the Universität and the higher professional schools, Fachhochschule. The former have a more “theoretical-abstract” orientation, more toward R&D, and the latter have a much more “practical-specific” orientation, or more applied.

In many European countries (Germany, Sweden, Austria, Italy, Denmark, Switzerland, etc) the qualifications similar to our Technical Engineering include periods of work experience
in companies of between 3 and 12 months. Thus not only do they achieve a greater integration of their graduates into the real world of work, but the university also benefits, since this focus requires a close and continuous contact between teaching centres and companies.

In Spain, society believes that technical engineering courses have, or should have, a more practical content than long engineering courses. In theory the universities themselves believe this too. For example, the LSI/UPC department (Dept. of Languages and Information Systems / Polytechnic University of Catalonia) explained in its 1994 annual report that in Technical Engineering Courses the teaching has “a very practical content”. But it is well known that in many universities this is not the case, since their Informatics curricula are designed in such a way that the first three years of the long course are basically the same as the short course, so that a Informatics Technical Engineer who has chosen the right options can graduate in Informatics Engineering in just two years. Obviously this design has some clear economic, organizational, and even social advantages. However, the distribution of subjects in the long cycle engineering courses is usually made in such a way that as the course goes on it moves from the general to the specialized, from abstract basics to the specific. One consequence of this common approach is that the most directly useful or practical subjects for a professional are dealt with towards the end of the course (The more specialized theoretical matters tend to be left for the doctorate). Therefore it may happen that, in spite of his apparent specialization, the short cycle Technical Engineer in Informatics (who can choose between two specializations) goes out into the world with a less specialized, practical and professional profile than a long cycle Informatics Engineer has.

7 Conclusion

The entry of technological courses, especially engineering courses, into the Spanish university (1972) introduced a major element of tension into the academic world. How does a Technical Engineering course for example, which had always had as its aim the training of skilled technicians to work in a profession, fit into the framework of a university traditionally oriented to the increase in scientific knowledge? In recent years this tension has been growing in intensity.

The university as a creator and critical transmitter of knowledge or science could disappear under the pressure of the current imperative to sell and compete. But the defence of the value of “knowledge for knowledge’s sake” [Blanco 97] is not necessarily incompatible with the defence of degrees (now incorporated into the university) oriented towards a professional role, as is the case with the Technical Engineer in Informatics. The idea of university today cannot be essentialist or narrow. As P. Etxenike says [Etxenike 95] “The present day University serves different ends and the problem is to get the balance right between these ends, their relative importance and their points of contact, while avoiding sterile reductionisms and trying not to sacrifice one end for the sake of another”.

The university is trying to take its first steps in a radical restructuring to adapt to the arrival of the higher education “market” [Bricall 97]. It is trying to adapt to the conditions of its age and its society, fulfilling the double mission of preparing students for work and for research.

At this juncture, one possible way the Spanish university can better serve society and meet its need for Informatics professionals, is to potentiate the practical, applied, and professional orientation of the Technical Engineer, while preserving and fostering a critical spirit and an ability to adapt to change.

Notes

The Spanish version of this article was published in Novática, issue 142 (Nov.–Dec 1999), pp. 56–61, with the title “La Informática de Gestión: ¿puede la Universidad española dar respuesta a lo que la sociedad le pide?”.


Bibliography

[Blanco 97]

[Bricall 97]

[Camps Paré 76]

[Denning 92]

[Denning 99]

[Etxenike 95]

[Glass 97]

[Knight et al. 94]

[Martí Recober/Puigjaner Trepat 80]

[Puigjaner/ Vergés 75]

[Sáez Vacas 92]

[Sanchis et al. 97]

Translation by Steve Turpin
The Changed Professional Model of the IT Project Manager: From the Design of Technical Solutions to Social Mediation

Beate Kuhnt and Andreas Huber

All workers must prepare themselves for the ever faster rate of change in the global economy and particularly in the IT industry. Acceleration is the magic word: IT projects, which in former times took five years, must be executed today in a year; more complex solutions must be created with less time and scarcer resources. How does this acceleration affect project management? How can social and organizational processes withstand technical developments? And how does this trend affect IT project leaders’ qualification requirements?

Keywords: IT Projects, Qualification, Human Resources, Systemic Perspective, Soft Factors.

Studies prove that IT projects fail from insufficient communication and information, from missing conflict management, from involving key people too late, and from inadequate basic conditions in organisations – not, as often assumed, because of technical or application problems [Weltz/Ortmann 92]; [Kraut/Streeter 95]. The Institute for Informatics of the University of Zurich has run the supplementary course “Leading and Managing complex IT projects” in the context of the centre’s “Man-Informatics-Organisation” programme for six years. From experience of the training activity in over 30 actual projects pursued by students, and from a qualitative study with the former students, future requirements applicable to the IT professional are derived. The requirements refer mainly to the so-called “soft factors”, which make IT projects successful in the long run.

From the software crisis to the acceleration crisis

The history of software engineering [Floyd 94] can be summarized as a sequence of crises, new challenges and corresponding strategies to master them, which can be assigned to the last four decades [Kuhnt/Huber 01]. Figure 1 shows the relationships.

After the founding years in the 60s, the software crisis followed. This was characterized by a rising error rate in programming. The first means for mastering the crisis was structured programming, egoless programming and so-called chief programmer teams. After some time of calm the introduction crisis followed in the 80s, which resulted from the fact that users of software products often did not see their requirements realised. Attempts were made to master the introduction crisis by means of participation concepts [Schuler/Namioka 93]; [Rauterberg et al. 94].

The result of the inclusion of users was an increase in the complexity of the projects: more and more people (stakeholders) signalled their needs, the requirements became more complex and the technical possibilities augmented. Project management became such a challenge that we can speak of a project crisis in the 90s that is well described in the already mentioned IPAS study1 by Frese, Weltz [Weltz/Ortmann 92]. With the mastering of the project crisis the “soft” success factors of IT projects became evident. Communication, coaching and teamwork were then also requested for IT projects [Pasch 94]. Furthermore evolutionary and iterative development models were preferred as prototyping supports the fast visualization of results. Another characteristic of the project crisis manifested itself as a change in the position of IT in the enterprise. The question asked was no longer “what is possible with IT?”, but “what and how much IT is needed in the enterprise?” Clients are taking more and more responsibility in their IT projects [Kuhnt 98].

---

1 The project IPAS (Interdisziplinäres Projekt zur Arbeitssituation in der Softwareentwicklung) was promoted financially by the Federal Ministry of Research and Technology. Three Subprojects from different domains co-operated: Industrial Psychology, IT, and Sociology.
The project crisis has not been mastered yet but a new crisis has appeared, which we will call the acceleration crisis. It has the following characteristics [Huber 00]:

- the complexity of the development processes and the resulting products grow dramatically.
- the development cycles of new IT business solutions shorten to continuous change and modification.
- the development of new IT business solutions becomes a corporate-wide, function-wide learning experience.
- combined IT and business solutions unify knowledge which determines the competitive strength of the enterprise.

The crises in the development of IT business solutions are mostly induced by complex social and organizational basic conditions, and not by technology.

Co-operation in IT projects

If IT projects are to keep up with the fast changes of technology and in enterprises, then the social and organizational processes in IT projects must also be accelerated. IT will have to prove effective during this process of change and thus give rise to change on various levels [Huber 00]:

- IT changes the relationship between enterprises, clients, suppliers, workers and financial sources
- IT changes market conditions and opens up new business possibilities
- IT standardizes and rationalizes work relationships
- IT changes the profiles required of workers involved.

Fig. 3 summarizes the role of IT in the development of new business solutions.

The development of new business solutions must include know-how from the areas of Human Resources (HR) and Organizational Development (OD). This is supported by a study by Martinsons and Chong. They determined there was a relationship between HR management and the success of IT projects. The incorporation of HR specialists during the planning and design phases reduces introduction problems, and the more the HR specialists get involved in IT projects, the greater is the success of the resulting IT solutions. However they state that the HR specialists wish to play a supporting and proactive role during the introduction of IT solutions, but are generally not involved. ([Martinsons/Chong 99] p.136.) They state four reasons for this:

1. IT professionals tend to underestimate the importance of human and social aspects of IT solutions.
2. A financial or competitive/political perspective fails to identify human factors and treats them inadequately.
3. Managers do not see a need for HR specialists.

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Properties</th>
<th>Qualification requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration and increase in complexity of IT projects</td>
<td>Scarcely resources, lack of time and money, shorter cycles</td>
<td>Basic attitude of curiosity and composure, holistic-networked thinking</td>
</tr>
<tr>
<td>IT business solutions determine the competitiveness of an enterprise</td>
<td>Engineering-oriented approach; too tardy involvement of the business side</td>
<td>Co-operation / Mediation between IT, Organisation development and human resource management</td>
</tr>
<tr>
<td>Organisational and social implementation of the solutions</td>
<td>Human and social aspects of IT projects are underestimated</td>
<td>Social-oriented methods and techniques; role-specific management</td>
</tr>
</tbody>
</table>

Fig. 1: Development of the Software Engineering Crisis

Fig. 2: Overview of the challenges and qualification requirements
4. The unsatisfactory IT knowledge of HR specialists prevents a partnership relationship with the IT specialist ([Martinsons/Chong 99], p. 127 ff).

In the future, IT project leaders will have a more integrating role that will mainly seek the co-operation of the various parties involved.

Social-oriented qualifications for IT project leaders

In the qualitative study into the effectiveness of the course, 120 former participants from the years 1994 to 1999 were questioned about the following topics: relevant topic areas, modification of basic approach, practical application of methods and techniques, changes in social competence, blockages and obstacles during the realisation. The return mounted to 21% with 26 responses. In the following section we summarize the results of the study in four categories:

Curiosity and attitude towards change

IT projects often entail changes in specialized areas: restructuring, work-organizational consequences, rationalization effects. This causes resistance, because humans do not like change but tend to leave everything as it is. Furthermore IT projects hold conflict potential since culturally different departments must operate and communicate together.

An IT project leader should know about such relationships in change processes. In the area of organizational development these phenomena have been well-known and described for years. In the area of informatics it often takes years until a project leader comes to this insight2. Much can be learned here from organization development and HR management.

50% of the participants of the course “Man-Informatics-Organization” (MIO) we questioned identified “composure in change processes” as one of the lasting effects of the course which they could put directly into practice.

Interest in and openness to different perspectives, aspects, opinions etc. are the second-most important output from the course they identified (change in social competence). In IT projects it is more and more necessary to co-operate with many different stakeholders; openness to different opinions, aspects and concepts in the same enterprise and in co-operation with external companies is the prerequisite for successful communication and understanding and hence for project success. The ability to know the limits of one’s own competence belongs here as well.

Holistic and networked thinking in complex problems

A trivial machine functions according to the input-output principle, where the output is determined by the input. A trivial system is independent of its past, can be analytically determined and exhibits a predictable behaviour. A nontrivial machine also follows the input-output principle; the output is however not determinable from the input. A nontrivial machine depends on its past, i.e. its present state, is analytically indeterminate and it behaves unpredictably. Complex IT projects behave like nontrivial machines, they are unpredictable.

IT professionals tend to isolate problems from their context and then to create the solutions bottom-up (classical systems engineering). This is an adequate and effective approach for well-structured problems. In the case of social problems however, this procedure can have a trivializing effect. With an “interference factor” in a team, removing the person concerned is often not sufficient. On the contrary: the problem is shifted, or another person takes on a disruptive role (scapegoat principle). The team must consider problems as a whole, and problems must be solved within that context.

In the study, new ways of thinking such as reality construction, networked thinking, perspective and systematic project management were rated in fourth place of the most frequently named topic areas, which occurred to them spontaneously in the MIO course.

Social-oriented methods and techniques for the acceleration of social processes

Apart from these changes in the ways of thinking and behaviour of IT project leaders dealing with soft factors requires new methods, techniques and concepts. We present here three methods that were most frequently mentioned in the study:

The starting point for each IT project should be an analysis of the project environment. It is a question of designating the relevant stakeholders in the environment of the project, weighting them, and eliciting their expectations and apprehensions with respect to the project. This is a simple, effective method, that involves little in expenditure, but serves as the basis for the control of social processes. It does not avoid resistance, differing interests, conflicts etc. but helps to foresee and localise them.

In the survey, this method was in the first rank amongst those applied by the participants (9 out of 26 persons) in their practical work.

2. In our course “Man-Informatics-Organization” more than 80% of the participants have more than five years of project experience.
A conceptual basis for proactive steering of IT projects is given by the intervention cycle consisting of the phases: observe, reflect, decide, intervene [Konigswieser/Exner 98]. (Figure 4).

The distinctions between observation, reflection on the observations, and action based on reflection assists acting in complex situations. In the survey, however, this procedure was only mentioned five times.

A kick-off workshop, in which the participants get to know each other and the project, serves to accelerate the social processes in IT projects. The central objectives of this workshop are the creation of the project team, the exchange of experience from similar projects, the identification of sources of apprehensions and obstacles for the coming project, and bringing the participants into relationships. In the study, however, this method was only mentioned three times.

Role-specific understanding of leadership

Rising complexity, acceleration and increasing relevance of social processes require changed leadership behaviour. By this we mean a mixture of different leadership roles, which we identify here using a metaphorical approach [Huber 00]:

The master is an individual who creates out of himself. His mastery has been gained from long years as an apprentice and a professional. The master exam is his certificate. He acts as an example and takes in hand whatever requires his mastery. His social influence, based on the introduction of apprentices and professionals, is limited. The actions of the master are what make him noticeable. He might not be visible to uninitiated people from the outside.

The perspective connected with a higher position provides the commander with an information monopoly. He decrees what has to take place, based on his superior knowledge and an information advantage. His social influence is justified by his function. After his retirement he is either glorified or decried. His perspective, his knowledge and his distance from events makes him the ideal observer.

The spaceship captain co-ordinates his team. Each team member has his own competencies and a restricted right of veto. He depends on his team to a certain extend, decides in disputes and functions as the last resort. He is often over-stretched with technical questions. His influence is based on the loyalty of his workers and he must constantly assure himself of their loyalty. This leadership role corresponds to one of a moderator, e.g. for development of innovative solutions in a dialogue, balancing individual and collective interests, or moderating workshops.

The gardener proceeds in accordance with the motto “good preparation is half the work” and prepares optimal basic conditions for his plants. He protects his project against negative external influences and strengthens positive influences. He intervenes in case of difficulties and problems but he lets things take their course. His authority is based on a delicate balance of confidence and distance from his team. He is predestined to create optimal basic conditions for a team and to protect it against negative influences from the outside.

Fig. 4: Intervention cycle

Ideally, a project leader embodies all four roles described, and above all knows when he must adopt each one. Leadership is a topic that only came into focus in the course in recent years. Due to an inquiry with last year’s participants, about which topic they missed, this however was in the first rank. The forth coming course will therefore put more emphasis on this topic.

Outlook

In summary we can determine three trends for the future career profile of the IT project leader:

Co-operation with advisors, project advisors, HR and OD specialists.

IT project leaders must seek co-operation with other specialists effectively. This fact must have an influence on project management and the corresponding methods and manuals. Project field analysis, a kick-off workshop and a coordinated termination of the project must appear in the manuals. The involvement of HR specialists, process companions, operation organisers etc. becomes the factor which achieves success if that involvement is initiated early enough.

Training on the job

Social-oriented action, networked thinking, confidence in dealing with changes and evolution, sensitivity for the appropriate level of intervention cannot be taught as university knowledge. Also in the course mentioned, “Leadership and management of complex IT projects” we reach the limits of what can be taught. However the study shows that some of the course’s targets have lasting effects. In the long run, however, we plead in favour of training on the job, as soon as there is an awareness of the problems.

Departure from engineering-like procedure

The attitude “nothing is too difficult for the engineer” leads to presumption in the project situation and to IT solutions over-dependent on technology. Engineering-like solutions are justified in the case of purely technical problems. In the context of project management they must give way to integration and intervention-oriented leadership principles.
Literature

[Floyd 94]

[Huber 00]

[Königswieser/Exner 98]

[Kraut/Streeter 95]

[Kuhnt 98]
Kuhnt, B. *Softwareentwicklung als systemische Intervention in Organisationen.* Zürich, Dissertation am Institut für Informatik der Universität Zürich, 1998.

[Kuhnt/Huber 01]

[Martinsons/Chong 99]

[Pasch 94]

[Rauterberg et al. 94]

[Schuler/Namioka 93]

[Weltz/Ortmann 92]
What Must a Training in Information Technology Impart?

Hans-Peter Hoidn

After 30 years of working with computers – initially programming on punched cards and now working as a consultant – two main characteristics of a successful IT professional stand out: firstly a knowledge of basic concepts, and secondly practical experience and social skills.

Keywords: IT Professional, Basic Concepts, Practical Experience, Social Skills.

1 Introduction

I recently asked participants at an international conference what today’s budding IT professionals should learn. Unsurprisingly, the most common reply was “Java”. Thirty years ago, when “computer science” was not yet established as a discipline, I was advised that learning ALGOL and FORTRAN would be a good starting point. At that time (1971, when I was a maths student in my first term), I found these languages extremely interesting, although I had no real idea of their future use. My working years have shown me that mastery of a programming language is no longer enough for an IT professional. Therefore the question regarding the fundamental content of a training in IT cannot be answered simply by quoting the name of the latest programming language.

Another aspect which should be taken into account is the fact that it is very rare for IT projects to be run by trained IT professionals. Computer science has been available as a course at the Swiss Federal Institute of Technology for the last 20 years (and was available a couple of years earlier in Germany), and today there is still a demand for considerably more computer scientists than are being trained by further and higher education establishments (the Swiss Federal Institute of Technology, universities and colleges). And those who were trained in a different subject before embarking on a career in information technology also have to be offered appropriate further education or training.

2 Constant factors in information technology

I have seen quite a few changes during my 30 years in IT. Not all of the skills that were previously required are still relevant today: for example, there is no longer a need to be adept at handling piles of punched cards, something you got the knack of when your own pile got dropped on the floor. But despite these many changes, there are still a few key constant factors. They provide clues as to the universal content of a training which will not become obsolete quite so quickly. So what are these basic constant factors in information technology? I believe they include the knowledge of concepts that are independent of hardware, operating systems and programming languages. Here are some examples to illustrate this.

Even the earliest programming languages included concepts for structuring programs. Then they were called modules and subroutines, now they are called components and methods (even though these terms do not have precisely the same meaning). Then internal processing was structured using locally defined variables, now it is arranged using internally defined objects. It will always be necessary to divide software into individual units that are easy to understand, with well-defined interfaces and internal processing that is not externally visible.

There are certain similarities in the way different programming languages are constructed. FORTRAN uses memory overlay, whereas C obtains similar results with pointers. An IT professional must be able to write programs, and must therefore be able to handle these language constructions, as well as having extensive experience of at least one programming language. Knowledge of a specific language is of less significance: today, Java is important, a few years ago it was Smalltalk, before then it was C and C++, and COBOL is still one of the most commonly used programming languages. It is very useful for project leaders if their programming experience can give them a feel for how long it will take to complete a particular task.

Comparisons can also be drawn in the case of database modeling. Something that is modelled today using UML (Unified Modelling Language) was previously depicted using ER (Entity-Relationship) diagrams (although here again, we are not comparing like with like). The important question here is not whether someone has UML skills, but rather whether they are able to model.

Another clue regarding constant factors can be derived from the fact that old insights from the “COBOL era”, such as those described in the well-known book “The Mythical Man Month”, are still valid in IT today. These insights apply mainly to working IT professionals and help to demonstrate why IT projects fail time and time again.

Hans-Peter Hoidn has been working since April 2001 as a Principal Consultant at PwC (PricewaterhouseCoopers) Switzerland. Following his diploma and doctorate at ETH Zurich, he has worked for a software company, a management consultancy, as a Technology Consultant specializing in software architecture and software development, as well as for a computer manufacturer and a major bank. <hans-peter.hoidn@ch.pwglobal.com>
3 IT Professionals

Specialist knowledge is only one aspect of a successful career in IT. I have seen questionnaires which gave the opportunity to categorize one’s knowledge of operating systems, programming languages and software products etc in the most diverse ways. However, the boxes ticked on these forms are not of central importance when assessing an employee, even a potential future employee. Since specialist IT knowledge can rapidly become obsolete, it is the ability acquired alongside the specialist knowledge that is more important.

The potential of IT professionals is most evident during job interviews. In such a situation, great importance is attached to their social skills, in particular their ability as team players, as well as their technical skills. These social skills are seldom covered in depth during education and training courses. Although such training does involve group work, and social skills are assessed under the heading “Project Leading and Project Management”, I believe that these aspects are given short shrift in IT training. I could well imagine “industrial psychology” becoming an integral part of IT training courses.

4 Conclusion

IT training must on the one hand convey the basic concepts, and on the other hand reinforce skills by providing the opportunity to learn an appropriate trade. A training that consists entirely of theory is as unsatisfactory as one that is oriented simply towards Java. Theoretical training does not impart experience, and a training that is primarily practical in content does not adequately promote the understanding of concepts. Here, as is often the case, it is not easy to find a balance.

A training in IT must impart social skills. It is often shortcomings in this area, rather than technical inadequacies, that are responsible for the failure of projects; this situation can be improved.

*English Translation: Pat Moody*
Computers will Enforce Professionalisation

Torsten Rothenwaldt

As an IT specialist working in industry, the author has experienced a change in the public perception of our profession: the specialist in bits and bytes is now considered the driving force behind innovations, and the dearth of skilled IT staff is seen as an obstacle to economic growth. What does this external change mean for the profession? The author puts forward the view that the current changes are further highlighting the absence of fundamental characteristics of professionalisation.

Keywords: Professional Ethics, Profession, Technical Knowledge, Professionalisation, Qualification, Quality

Our profession’s public profile is higher than it has ever been. Daily news broadcasts include reports from the world of computing; the demand for skilled workers appears insurmountable. Is there a rosy future in store for the IT profession?

Today, “something to do with computers” is as good as an insurance policy for one’s professional career. Studies are also forecasting an enormous shortfall in skilled workers in the years to come. Estimates for Germany vary between 75,000 and 350,000 too few IT specialists out of a current total of around one to two million ([Meyer et al. 2001]). Even if this shortfall does not turn out to be quite so large as the economy would have us believe, what is more obvious for someone interested in the field than to jump onto the bandwagon as quickly as possible?

As an employee of a computer group, the author is often asked where the best longer-term prospects lie. The sobering answer is that everywhere is much the same (except that some places are better known than others, because good people happen to have worked there). And one should also ask oneself whether, as well as an interest in the field, one also possesses a lot of talent and, above all, the energy to keep starting from the beginning over and over again, because in this, one of the most unprofessional of industries, it is only stamina and ability that guarantee a long-term income.

Why IT specialists are missing

In order to understand this view, we need to remember how we arrived at the present situation in the first place. First of all, it is well known that the industry has brought its recruiting problem on itself (some of us say). IT professionals, even the “relatively pure” ones (not hybrid ones or even mathematicians) have not always been in demand. The author well remembers the first half of the nineties, when job advertisements for IT staff were a rarity. Whereas today about two thirds of the jobs advertised are for end user organizations ([Meyer et al. 2001]), at that time the opposite was the case. Outsourcing was the objective of managers who were aware of industry trends, and the question discreetly asked by a new manager of his colleagues “Why do we need these techies anyway?” clearly indicated our worth. Specialists in large computer companies were having to worry about their jobs because the industry giants were making losses. We obviously couldn’t advise young people to choose our profession. At that time, a business management diploma guaranteed a job, with known consequences for universities. The fact is that our profession is much more dependent on economic trends than that of doctors, lawyers or chimney sweeps, and the next downturn is never far away.

Incidentally, it would appear that the recruitment problem is linked to a cultural symptom of our society: in a society where people can openly brag about the poor maths marks they got, we need not be surprised at the quality and quantity of applicants for corresponding places in higher education.

The second reason for the present situation lies in the dynamics of IT demand based on technical developments and their commercialisation: PC, Internet, media integration. Demand is not just limited to activities directly related to these areas, but rather there is a domino effect that extends into areas previously considered defunct. The IT industry has not anticipated these developments, and has done little to prepare itself for them. The situation is unique due to the combination of a dip in student numbers, the new economy and the absence of a long track record in the industry. Every day we are reminded how much our techie expertise is in demand. But it’s not every decade that we get a new Internet.

Thirdly the industry has not only a recruiting problem, but also primarily a demand problem. A classic phrase in job advertisements: “You will fit in with our team best if you are not..."
more than 35 years old.” All in all, the age issue is a delicate subject in our profession. Of course an older doctor or lawyer would have certain difficulties finding a new job. But are 50 year old doctors, lawyers and judges “talked into” early retirement because their ability to perform (flexibility, tolerance of frustration) is tacitly denied? What strategies have I got to help me survive my thirty or more years of working life as an IT professional in industry? In order to survive the obsolescence of large parts of our professional knowledge, the ability to transfer skills to completely new areas is just as important as being quick on the uptake. That requires a capacity for abstraction, something that is hardly taught at all these days.

Are we a profession?

This explains why talent and stamina are important if one wants to be successful in our profession. But is our trade really a “guild of professionals”? Professionalisation means (according to [Schinzel et al. 2001]) reserving the profession for an elite (with membership being awarded by virtue of education and training), and the delimiting of areas of responsibility from other professions. For many IT professionals working in industry, this bears no relationship to reality.

The professionalisation of a subject takes place at two levels: institutional and intellectual. Institutionally, IT is firmly established (professional associations, IT institutes, state money for research). Moreover, an active professional life gives a sense of belonging to a “professional community”.

On an intellectual level the situation is different, at least in large parts of the industry. Paradoxically, the present dynamics are having a downright antiprofessionalising effect because, due to the shortage of skilled staff, fewer and fewer computer solutions (particularly software and systems architecture) are being developed by trained specialists. The result is often products of a quality that would never be accepted by clients in other industries; popular comparisons here are cars and aeroplanes. Speculation as to why our products are nevertheless purchased leads to a broad spectrum of theories, ranging from resignation to cynicism. IT is certainly a very young and heterogeneous discipline. But the dynamics already mentioned are hindering its process of maturing, because they are restricting the development of the classic characteristics of professions: a professional code of ethics, technical knowledge, monopoly and autonomy.

On the subject of a professional code of ethics, there have been quite a few discussions and publications in recent years (ethical guidelines of the German Society for Informatics, the ACM’s “Codes of Ethics” [ACM 1992], Software Engineering Code of Ethics of the ACM and the IEEE Computer Society [ACM 1998]). But in practice these are relatively unknown, and training does not appear to include such issues.

Technical knowledge is another tricky issue. An exclusive knowledge base such as that of doctors or lawyers requires not only a different external view of the profession but also a different concept of oneself. There is of course informative technical knowledge that is available only to those on the inside (i.e., as the result of years of experience). But imagine a situation where a doctor is making a home visit, and the assembled relatives start discussing the dosage of drugs, quoting from recent popular medical magazines. Anyone who makes regular site visits to attend to large MS-Windows systems (containing, for example, all the accounts, production planning and logistics applications of a medium-sized company) is familiar with this scenario. Our profession is often seen as a trade in which one can go a long way by virtue of experience alone (like a do-it-yourself enthusiast who is perfectly able to put up shelves – only they don’t look quite like the ones in the DIY superstore). All too often, this is the way we ourselves tackle our work.

The downside to this lack of a monopoly on knowledge is that there is no monopoly on the service offerings of IT professionals. And along with this goes less professional autonomy (in the sense of freedom from external control of the profession).

But the situation is not uniformly gloomy across the entire industry. Talk in the public domain (and in articles such as [Schinzel et al. 2001]) is often focused on the development of applications software, an area where these characteristics of professionalisation are in fact not very much in evidence. That is the nature of things. In the field of systems software, matters are different. Although Linux is used as an example to the contrary, it has now become so complex that innovations in the kernel are being carried out more and more by paid professionals. And in the case of microprocessors, bus and channel standards or networks, most professional characteristics are in evidence. But wait a minute – is this still really IT? Or are we talking about telecommunications and semiconductor technology, in other words classic engineering disciplines outside the scope of our subject matter? But then again, doesn’t the development of such hardware and software require significant scientific and technical input on the part of IT?

One fundamental reason for the lack of professionalisation in IT is precisely this problem of delimiting the subject – do we see ourselves primarily as belonging to an engineering discipline, or an abstract formal one? Despite his original training, the IT professional in industry who is paid to develop, maintain and advise on a marketable product does not perceive the subject as abstract formal, even if programming languages do force him to think formally (at the end of the day, languages are only tools).

Another related reason is that scientific computing and information technology often develop only in parallel, rather than together. The frequent reproach concerning the impracticality of training and research cannot be dismissed: applications are seldom objects of scientific rigour. There are no scientific publications (as opposed to those merely containing rehashed descriptions of individual products) available on many vital topics for practising IT professionals, or alternatively a gigantic formal tome provides, at best, trivial statements (trivial in comparison with the complexity of mature commercial systems). Getting into these tomes requires a lot of enthusiasm and a total mastery of such publications, in the hope of better understanding the objects we work with on a daily basis and thus becoming more professional.

Fear not, the strident demand to remove theory from the curriculum will not be repeated here; that would be far from
beneficial for the development of the profession. But applications (yes, even commercial systems) must be freed from their exile outside academia. Because conversely many so-called professionals in the industry can simply not imagine that there is sometimes nothing more practical than a good theory, and that their well-meant solutions suffer from limitations ("will be solved in a future release") because they have not taken account of basic scientific facts. The excuse given is the universal cliché of lack of time. The really good industry IT professional has to have a very good grasp of his theory, and be interested in research. It is, however, very difficult for him to find research references to many of the objects with which he works. (Is this also the case for practising doctors?) The splitting off of courses of study such as software engineering and media information technology is a sign that applications-oriented elements are not sufficiently integrated into IT courses.

Unlike many others, the author is firmly of the opinion that IT can still be saved, because the extent to which computers have infiltrated our lives means there will be changes to our profession that have long since taken place in other disciplines. The monopolization and autonomy already mentioned above as being lacking will inevitably materialize. We are increasingly dependent on computer systems. But anywhere that a high level of damages could result from faulty or indifferent workmanship, and where the complexity of tasks can no longer be handled by "semi-skilled" staff, professionalisation is of particular significance.

In order to avert this risk, other professions establish norms, standards and supervision. These are first defined for the work products. But anywhere that working people have to make decisions on an ad hoc basis, we are moving towards examining and assuring the professionalism of the people involved rather than the quality of the products. This happens by means of entry requirements (solicitors, doctors), and is the job of professional and industry associations, who lay down certain norms for their members (estate agents, cleaning companies), or is imposed on service providers by product manufacturers (garages). Society is increasingly demanding similar quality and safety checks in our industry as well, and that is changing the face of the profession. The author assumes that a hierarchy of responsibilities will develop, similar to that in medicine, with a strictly enforced hierarchy of qualifications (in sensitive areas, at least). But professionalisation takes time (the history of medicine includes the questionable medical practices of the Middle Ages). Boom times like the present are, on the one hand, not suited to this development; but on the other hand, they do draw attention to the problems.

To reiterate the comparison with cars: just as these days hardly anyone would consider doing anything more complicated than changing a light bulb themselves, so computers are also becoming the exclusive domain of appropriate occupational groups. Of course the public will retain a considerable interest, and there will always be people who are involved in the subject matter of IT in a semi-professional capacity, either as users or as a hobby. But this also happens with other subjects, without bringing the self-image and the external image of the profession into question.

We must be responsible for technical knowledge, a professional code of ethics and self-regulation (including the quality of our work and hence our reputation), and above all we must also ensure that researchers and practitioners feel a sense of belonging to the same profession. We are faced with no alternative: the omnipresence of computers is quite simply compelling us to become more professional.

References:
[ACM 1992]
[ACM 1998]
[Meyer et al. 2001]
[Schinzel et al. 2001]

English translation by Pat Moody
The Future of Software Engineering

Karol Frühauf

Three issues stimulate my thoughts. The three aspects are the nature of software, the people creating software, and the essential factors in the creation of products that contain software.

Keywords: Software, Methodology, Tools, Certification, Social Environment.

The Unknown Nature of Software

Communications of the ACM dedicates its March 2001 issue very immodestly to “The Next 1,000 Years”. The first contribution, by Philip G. Armour [Armour 01], postulates:

1. Software is a knowledge storage medium, not a passive medium like a book: software activates knowledge.
2. Software is the future currency of the world.

It is an established fact that we live in an age of knowledge work. But the idea that software can be used as currency is astonishing and enticing. Money simplifies the exchange of goods: I cannot build a fence from apples, but I can exchange the apples for money and buy slats, nails, etc. to build the fence.

In line with Armour, software will be used instead of cash in the near future. Yet some problems need to be solved, such as that of exchange rates (standardisation of the interfaces between software products or software bundles), the problem of inflation (multiple availability of the same functionality, blowing up programs with unnecessary code) or the protection of large nominal values (e.g. the programs that control universal milling machines for the carving of atom bomb cores), to name a few. Once all these problems have been solved, then 100 kg of apples will buy one thousand MS Word 2095 licences, and a 10 mm nail will cost the equivalent of a user licence.

A currency must meet certain requirements: We expect from a currency that it is safe from falsification, we must ensure that it can be modified by nobody after it has been brought into circulation. We must also make sure that modifications are not necessary, i.e., that the encapsulated knowledge is always applied correctly when the software is used. The software will not need to be error-free, but it must not do anything wrong. Thus, every software will have to have a comparable reliability, like the software in today’s sensitive applications. Such software cannot be produced by adventurers, turncoats, or enthusiasts without any software engineering knowledge.

A currency must be robust against inflation. A sculptor with academic titles who joined an Interned company as Art director, proudly reported to me that he designed software that solved the problem of multilinguality. My reaction was that he was not the first to solve this problem clouded somehow his pleasure but fortunately did not damage his self-assurance. Software that solves already solved problems, that therefore does not encapsulate new knowledge, when used as currency, heats up inflation. Today, this is still tolerated by informatics and even demanded by management.

If software is to become a useful currency in the future, it is necessary that all we know about software engineering is consistently applied. It is also necessary that we continue to develop procedures, methodologies and tools so that our currency is not too expensive. Integrability, correctness, portability, testability, reliability are in the forefront; user friendliness, efficiency, flexibility, maintainability must be granted. In other words: the quality of software is crucial.

It is not surprising that the “light methodologies” that came up in the last few years are all without compromise with respect to quality. There are three areas in which action is required:

1. research in the area of software engineering must supply better bases for methods for the inexpensive development of more reliable software;
2. software vendors must focus on product quality and behave accordingly;
3. software developers must know their business in order to develop software that meets the requirements for a currency.

The People Creating Software

The IEEE Computer Society has developed the Certified Software Engineering Professional (CSEP) program, “a credential that will recognize the comprehensive set of skills needed by a professional software engineer” [IEEE CS 01]. It will cover the breadth of topic areas important to software engineers rather than focusing on narrow tool-specific skills recognized by vendor-sponsored certification programs. The CSEP is not a license, but a peer recognition certification designed to measure an individual’s mastery of the fundamental knowledge required to perform the functions of a software engineer. Very similar projects exist within CEPIS (EPICS, European Professional Informatics Competence Service) and in Switzerland (I-CH).

Certification is particularly popular in English speaking countries. Certifications as Software Project Leader and as Software Testers have existed for some time. They are unlike

Karol Frühauf is co-founder of INFOGEM AG in Baden (Switzerland), a company dedicated in consulting and training in Software Engineering. For more than 25 years Frühauf hopes that the quality of the future will outdo the software of the past. <karol.fruehauf@acm.org>

© Novática and Informatik/Informatique
the software vendor specific certificates that can be acquired for a large amount of money; these are licences giving authorisation to perform as an expert in that particular proprietary software palette. Licences in the true sense are those obviously necessary for those professions such as physicians, attorneys.

Also in the 21st century, professional associations were the institutions that defined the “necessary basic knowledge” [SWEBOK]. It is the educational institutions that are responsible for leading to the mastering of this knowledge. With the support of the SoftNet Initiative of the Federal Government in Switzerland the Higher Technology Colleges (Fachhochschulen) try to jointly adjust their curricula to SWEBOK.

Fashioning of Software

[Norman 99] pleads for the breakthrough with the universal computer, like today’s PC; it can do a lot, but nothing can be managed simply with it. It should be replaced by dedicated “informational appliances”, devices that, ideally, support certain human activities. “People should learn the task, not the technology. They should be able to take the tool to the task, not as today, where we must take the task to the tool.”

Someone who has purchased a PC in order to do e-mail, i.e. writing and sending messages, and receiving and reading messages, and has successfully carried out all the necessary software settings has to perform several steps which, apart from the first one, are not directly associated with mailing: a) switch on the computer, b) wait for the operating system to start up, c) make a three-finger exercise, d) establish the modem connection, e) start the mailer program.

When I start my car, the display immediately informs me about oil pressure, fuel available in the tank, the time and many other details necessary for driving. All I have to do is to turn the key. There is no waiting and no starting of any programs.

To design the appliance that will realise our “computerless future”, Norman proposes the following principles: a) observe the users, find out what they do, b) analyse the market, find out as much as possible about it, c) formulate and validate the users’ and the market’s needs, d) build, jointly with the user, mock-ups and prototypes of the future product, e) write the operating instructions of the future product, f) design the product on the basis of the operating instructions, the mock-ups and the prototypes, g) check the design and modify it continuously.

These principles assume certain qualifications of the workers: Behaviour observer, Behaviour designer, Model constructor, User tester, Graphic and industry designer, Technical author. With the exception of the industrial designer I know of no training centre in Switzerland where any of these skills can be acquired.

In order to design the products of the future, the software engineers must be able to bridge the gap between the computer and the application domains, and to make the computer disappear. They must be prepared for that task. If the educational institutions do not provide this skill, the professional associations have to do it.

The Software Engineer in the Social Environment

An additional aspect: Software Engineering is above all communication. Communication among people, managers, designers, vendors. Communication between different terminologies, and finally in natural language that should bind everything together. [Beck 00] promotes verbal communication as the only adequate one. Verbal and written expression must be learned, it must be a crucial point in the training for software engineering. Together with the ability to work as a team, to share tasks, the ability to agree, be reliable and gain respect.

Conclusions

The economic role of software could change in the future. The importance of software engineering will have to be significantly increased, and this will only be possible with the aid of efficient means. The challenge will need to be met by a strong change of the content and format of the training at all levels. Not every user of informatics is an informatician. The knowledge of a programming language is no sufficient to be a software developer. This truisms cannot be repeated enough.

References

The Future of Informatics as a Scientific Discipline

Pedro G. Gonnet

Informatics is established since many years already in many universities as a scientific discipline. It usually has its own building, is on equal foot with the other faculties and has research and teaching like the other disciplines. Nevertheless, informatics finds little or no acknowledgment as a science outside the university, often even outside its own field. On a long term this will lead to disastrous consequences for informatics as science and application.

Keywords: Informatics, science, college, education.

1 The problem

Whoever starts a computer science study at the ETH Zurich (The Swiss Federal Institute of Technology) must be prepared for quite a lot. I mean not the fact that in the first year one sees rather little of the computer and must face up intensively with analysis, algebra and physics, but the fact that what we study is not understood outside the university.

At the beginning of my study, I was swamped from friends and family members with questions about the configuration of the Web browser, the installation of the printer, or the format battles among word processors, until it became clear to these people that studying informatics it is not about application software and their vexations, but about something much deeper – something that is more related to mathematics, statistics, thinking with pencil and paper, and very little to the hacker picture that is in all heads.

Now where I am about to complete my study, and thinking about taking a doctor’s degree, I am faced with even more inquiring faces: hardly anyone can imagine that one could graduate, let alone doing research on university level in informatics. Hardly anyone can imagine that after completed study one is capable of anything but entering features into user applications, or starting-up a company with Internet services of the newest generation, throwing about with technical terms and then retiring with 40 – simply anything but a scientific career.

However, one cannot expect from everyone to give serious thoughts to, and understand current informatics research topics. I don’t. It is confusing, however, that most people acknowledge chemistry, biology, physics or mathematics as sciences, although they don’t know what is researched in these disciplines. Everyone knows, for example, if his car breaks down he has to go to the mechanic and not to the mechanical engineer. In the “classical” disciplines everyone can distinguish between science and application, but not in informatics.

2 The even bigger problem

That only few know what it is all about in informatics as a science is not actually alarming… scientists do not care much whether non-scientists understand what they do. Unfortunately this is different, again, in the case of informatics. Many people – above all in industry and economy, and under their influence politics, too – do not estimate the value of informatics as a science.

The pharmaceuticals industry knows very well the difference between a well trained chemist and someone who mixes LSD in his leisure time. It has therefore also a very strong interest promoting chemistry in universities, because it knows that it depends very much on their product, namely qualified chemists. Similarly, airlines employ only graduated engineers for the construction of their aeroplanes, and building contractors entrust only to qualified structural engineers with the building of skyscrapers. The reason for this behaviour is clear: the concerned businesses want to sell effective medications, secure aeroplanes are needed, and houses that survive storms without collapsing. Otherwise one goes bankrupt or is brought to court for the faults.

In the informatics industry a different culture prevails: Users accept unreliable software systems, is has become part of their everyday life. The user community should vociferously demand better quality so that the informatics industry realizes that it needs better qualified specialists. Otherwise the informatics industry doesn’t have any need to promote the scientific education of informaticians.

3 The biggest problem

Informatics as science – computer science – is not yet in trouble: Compared with other academic disciplines informatics is quite inexpensive and does therefore not depend on industry sponsoring. Because qualified informaticians are not really in demand, the already too high number of informatics students...
could return to acceptable dimensions, and the teaching burden of professors and assistants sink so that they can dedicate themselves increasingly to research.

In the other disciplines, the distinction of science and application is common. Unfortunately this is not the case in informatics, because the class of application specialists hardly exists. The informatics apprenticeship still is in its beginnings, the "Swiss informatics certificate (SIZ)\(^1\) still contains nothing about programming, and the technical colleges have not yet found their vocation.

Looking down from the ivory tower, scientists may grin at the situation and make sport of the poor computer science industry. However, they should not, because instead of promoting and developing an "informatics middle class", the computer science industry tries to turn the universities into what they urgently need: practice-oriented educational establishments that produce in great number the "informatics middle-class" they need. The present-day rush for informatics graduates has nothing to do with their quality but rather with the demand for a great number of informaticians and the fact that universities produce them. Industry wants to modify the syllabuses, theory and mathematics shall be removed, more product knowledge shall be imparted, the students shall receive some notion of management and project management, and the relationships to the economy shall be strengthened. This is where the biggest threat to the informatics lies.

4 Why does informatics need science?

In the context of the role of the universities much thought is being given about whether informatics needs a science when it doesn’t want to cooperate with the industry.

Let’s for a moment have a closer look at informatics: every problem, however trivial, and its solution, can always be reduced to the same: algorithms and data structures. Efficiency and correctness of a solution will finally depend on their choice. Here we are lost without informatics science. For sorting data, for example, it is much simpler to copy a quicksort algorithm from one’s favourite algorithm book than to consider whether a binary tree data structure is more suited for this problem. Even if one knows that a binary tree is necessary, developing a much more superior, but also more difficult AVL (high balanced) tree is not an obvious task.

If one hasn’t sufficiently examined the matter, the solution is usually quite far from the optimum. The choice of suitable algorithms and data structures necessitates a deep knowledge of theory and mathematics. Even better understanding is necessary for the development of new algorithms and data structures. This knowledge can only be maintained, taught, and developed only by science.

Moreover, the hardware on which informatics runs, does exactly what it is told, and hence already the smallest error in a

many thousand lines program suffices to throw the whole into the domain between useless and life threatening. Consequently, writing each program line requires utmost accuracy and expertise which, again, can be maintained, taught, and developed only by science.

However, some readers will point out that the development of new algorithms and data structures takes place outside universities, too. This is correct. The pharmaceuticals industry develops very many medications and procedures, and nevertheless chemistry remains in the universities, because the industry would not be capable to train the next generation of chemists. Why does this thinking not have become generally accepted within the informatics industry yet?

I also cannot agree with the claim that informatics should concentrate on developing better tools – i.e. compilers and programming environments – in order to protect the application informatics from their own ignorance. A compiler will never make a quicksort out of a bubble sort, or a b-tree from a doubly linked list, or a correct algorithm from an erroneous one.

5 About the future of informatics

Now about the actual topic, to which I was asked to express my view as a university informatics student: “Present and future of the informatics profession”.

As for the present of the discipline, we are not altogether in a bad situation, at least at the ETH. The science is treated and taught as such. The students get a solid base, including analysis, algebra, physics, statistics and logic, which are of great importance during the study.

The future however, makes me worry. Not taking the subject seriously will have nasty consequences: the cooperation with other departments\(^2\) limits itself increasingly to service lectures, where product knowledge is required to be taught. Each department tries to build its own informatics, and the informatics department becomes more and more the target for media-impressive pilot projects, like for example Web-based study with subsequent virtual diploma – unthinkable in the “proper” sciences like mathematics, chemistry, or physics.

Informatics also gets more and more exposed to the pressure of the industry, which campaigns for a more practice-oriented curriculum and tries to exert influence on all levels. The mathematics-suspected lectures die out, while topics in vogue enjoy a boom. At the end of their study most of my fellow students will try their luck in the industry. It is however questionable, whether they find this luck, because the industry informatics – particularly in large companies like banks and insurances – does not offer the intellectual challenge we are accustomed from the study. If no quality consciousness develops in future in the informatics industry, what we learned during study will never be applied because good products are not at all required. I look at these developments with concern.

---

1. SIZ (Schweizer Informatik-Zertifikat) a computer user skills certificate comparable to the "European Computer Driving Licence" (ECDL), but known and recognised only within the German-speaking part of Switzerland.

2. ETH is divided into departments, such as Informatics, Electrical Engineering, etc.
The Shortfall of Informatics Professionals and the Legislation Regulating the Informatics Profession in Spain

Rafael Fernández Calvo

This article* describes the shortfall situation concerning Informatics professionals in Europe and looks at the legislation regulating the legal practice of the Informatics profession in Spain by laws passed by the Autonomous Communities (regions) and its potential effects on this shortfall.

Keywords: Informatics Profession, Shortfall of Informatics Professionals, Skills Gap, Legislation, Professional Colleges, Spain, European Union.

1 The Shortfall of Informatics Professionals

It is no secret that European experts on the subject estimate that the current shortfall, also called skills gap, of Informatics professionals of all levels and functions in EU countries will have doubled by the year 2003. We will need (or, to be more precise, we need now, urgently) all kinds of professionals: from information system directors to programmers or analysts, from project managers to systems operators, from webmasters to maintenance technicians or call centre staff…, university graduates or postgraduates, specialized in Informatics or not, or non-university graduates. In short, professionals of Information and Communications Technologies (ICT) in the broad sense described by Peter Denning in the interview published in the online magazine Ubiquity, from ACM, [Denning 00a] and reproduced in Spanish by the magazine Novática [Denning 00b].

In European Union (EU) countries, according to data supplied by the ICT Consortium [ITC Consortium] – formed by IBM Europe, Nokia Telecommunications, Philips Semiconductors, Thomson CSF, Siemens AG, Microsoft Europe and British Telecommunications –, the shortfall will be 2,362,000 Informatics and Telecommunications professionals by the year 2003, the equivalent of 19% of all the professionals in this sector. By countries, Germany will be in the worst position, where no fewer than 546,791 will be needed (you can’t help marveling at the incredible precision of these statistics) while in Spain the estimate is 83,538.

The European Information Technology Observatory (EITO) [EITO–EEIG 01] boosts this shortfall estimate for the same area and year to 3,670,000 (110,000 in Spain), while the shortfall figure supplied by the consulting firm IDC [Milroy/Rajah 00], see Figure 1, and the Union Network International (UNI) [Union Network International], while still very high are not so pessimistic (1,740,000 and 1,700,000 respectively). But the UNI believes that the policy of recruiting non-EU professionals as Germany is doing (especially from Eastern Europe and Southern Asia) can only be a provisional solution and will not solve the problem in the mid to long term since it is rooted in the fact that educational centres at all levels do not produce enough Informatics specialists, and, furthermore, that European students receive insufficient ICT training at all educational levels, as was pointed out in the conclusions to the recent E-Learning Summit [European eLearning].

These shortfalls in professionals are occurring in the context of a strong growth in computerization and connectivity of companies, even of small and medium size ones, of public administrations and of the homes of all the developed world, including of course the countries of the EU. Specialists and high-level authorities of the EU recognize that ICT represent a key factor for innovation, service to customer and citizens, and competitiveness, and warn that if these shortfalls are not satisfactorily covered, the development of our continent within the framework of the so-called “Information Society” may be seriously compromised and, with it, our countries’ economic and technological progress and social cohesion.

So in the last two years, with the industry’s backing and the collaboration of the unions and other social agents, measures have been taken to find a solution to this worrying situation. These measures go from the German solution of encouraging the immigration of Asian and Eastern European Informatics

*Rafael Fernández Calvo is a Law Graduate of the Universidad Complutense of Madrid, Graduate in Journalism of the Escuela Oficial de Periodismo (Official School of Journalism) of Madrid and Diplomate in Management Skills from ESADE. After a long career in the world of Informatics as a specialist in Databases, Data Design and Strategic Consultancy for Information Systems in IBM, he is currently editor of Novática, the journal and magazine of ATI (Asociación de Técnicos de Informática – Association of Informatics Technicians), acting editor of this association’s website, co-editor of Upgrade and associate professor in the Faculty of Law / ICADE of the Universidad Pontificia Comillas, Madrid. He is a senior member of ATI, and was member of the Board of the association and the first president of its Madrid Chapter. <http://www.ati.es/GENTE/rfcalvoi.html>

© Novática and Informatik/Informatique
professionals that was mentioned previously, to the training of ICT professionals and users (the ECDL European Computer Driving License program, promoted by CEPIS, is an important initiative in this respect [ECDL]).

2 Spanish Regulatory Legislation

In recent years Official Colleges1 of Informatics Engineers (higher level graduates) and of Informatics Technical Engineers (middle level graduates) have been set up in the different autonomous communities, or regions, of Spain, that have recognized competence to draft legislation in matters of Professional Colleges. The profession’s regulation is one of the factors which could have an effect (positive or negative) on the eradication, or at least the drastic reduction, of the Informatics profession’s recruitment shortfall.

This drafting of legislation is being promoted by two associations of Informatics Engineers, AII, also known as AI2, (Asociación de Ingenieros en Informática – Association of Informatics Engineers –, <http://www.aii.es>) and ALI (Asociación de Ingenieros, Licenciados y Doctores en Informática – Association of Informatics Graduates, Doctors and Engineers, <http://www.ali.es>). However the ATI (Asociación de Técnicos de Informática – Association of Informatics Technicians, <http://www.ati.es>), which has the largest membership by far amongst the Spanish Informatics professional associations – let’s remark that about 30% of the ATI members are university graduates in Informatics –, while recognizing the rights of Informatics graduates to create Colleges, points out the dangers that in their view an exclusionist legislation would have [ATI]. ATI and ALI are member societies of CEPIS (Council of European Professional Informatics Societies). All three associations are naturally behind the progress of the Informatics profession, but each gives their support to this progress from the perspective of their own traditions, principles and constituencies. With regard to legislation regulating the profession they do not always agree, although ATI and AI2’s positions do converge on some key points.

According to the data currently available to the author, laws authorizing the creation of Official Colleges of Informatics Engineers and Informatics Technical Engineers have been passed in Asturias, Catalonia, Murcia, the Basque Country, and Valencia; they are at different legislative stages in Castile and Leon, and Galicia; while in Aragon they were rejected at the consideration stage. There was a proposal of a law in favour of a National College presented by the Spain’s ruling Partido Popular in 1999 but its course through Parliament was interrupted by the calling of legislative elections in March 2000, and the subsequent dissolution of the Lower House. So far no other white paper on this issue has been presented, in spite of the fact that the Partido Popular enjoys an absolute majority in that House. For a summary of the legislative situation see table 1.

3 Some Appraisal Criteria

What repercussions might the creation of such Colleges have on the progress of the profession in our country in general, and the making up of the shortfall described in the previous section in particular?

---

1. College: Government approved society of persons of the same profession
In general terms, we can all agree that any initiative that helps informatics professionals to get together to improve their profession and enhance their prestige, their training, their ethics, and their influence on society is to be welcome, especially in a country in which the culture for professional associations is weak outside the traditional professions (basically lawyers and doctors) who are obliged to belong to Colleges to be able to practice.

But alongside this general and abstract positive appraisal, the laws governing the creation of these Colleges, and their own by-laws and statutes, should all be carefully examined so as to be able to reach more specific conclusions. An analysis of the situation can be made by taking two important criteria as a starting point: the need for compulsory membership of a professional College to be able to practice and the possibility for informatics professionals with proven experience who have university qualifications in non-informatics subjects to join a College during a transitional period.

### 3.1. Compulsory College Membership

The first criterion, the need for compulsory membership to practice, is established in the laws governing the setting up of the Colleges of Murcia (promoted by AI2) and Valencia (promoted by ALI), or in their statutes. It makes it theoretically impossible for any informatics engineer or technical engineer who is not a member of a College from practicing as an informatics professional in these regions, following the corporativist and monopolist model of the traditional professions mentioned previously.

Bearing in mind that of the 400,000 informatics professionals of all levels and functions currently estimated to exist in Spain, informatics engineers or technical engineers would barely make up 25,000 (scarcely 6.25% of the total) today, it seems clear that the strict application of the those regions’ College laws would not only not help to alleviate the shortfall in professionals described in the first section, but would paralyze the overall economic and social activity by effectively expelling from the market 94% of the informatics professionals. At present the official figure regarding the number of members of the Colleges of Murcia and Valencia is not available but unofficial information suggests that only a small percentage of informatics graduates of either region have joined the Colleges, which, if this is true, would be another argument for those who think that in practice it is impossible to apply the laws passed in the two autonomous regions.

Some legal experts are of the opinion that the College laws of Murcia and Valencia could be considered unconstitutional since, on the one hand they call for compulsory membership which general legislation regulating professional Colleges, supported by jurisprudence, limits to rare exceptions and, on the other hand, they would be contravening European legislation on free circulation of workers, by establishing conditions for practising as an informatics professional that are not demanded for that profession in any other member state of the EU. For example, if an informatics professional from Finland, even with a postgraduate qualification in informatics, were to want to work, say in the city of Alicante, in the region of Valencia, contracted by a local company, he or she would not be able to, by law, practice this profession if he or she didn’t join the College. And if that same Finnish informatics professional did not have a university qualification in informatics, this person would not be entitled to join and therefore would have no chance of working as an informatics professional.

On the other hand the laws of Asturias, Catalonia and the Basque Country do not establish College membership as a requirement to practice the profession, which creates a favourable situation for the profession: the coexistence of graduate informatics professionals with non-graduates.

### 3.2. Joining During a Transitional Period

The application of the second criterion (the possibility for informatics professionals of proven experience with university...

---

### Table 1. Situation of the laws regulating Official Colleges of Informatics Engineers in Spain

<table>
<thead>
<tr>
<th>Autonomous Community (Region)</th>
<th>Date the law was passed</th>
<th>Law promoted by</th>
<th>Does the law or the statutes of the College oblige you to be a member of the College to practice?</th>
<th>Can professionals with other qualifications become members of the College?</th>
<th>Does the College already exist?</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aragón</td>
<td>Rejected by the autonomous parliament May 2001</td>
<td>AI2</td>
<td>?</td>
<td>Yes</td>
<td>No</td>
<td><a href="http://www.ali.es/infcolpv.htm">www.ali.es/infcolpv.htm</a></td>
</tr>
<tr>
<td>Asturias</td>
<td>21 May 2001</td>
<td>AI2</td>
<td>No</td>
<td>?</td>
<td>Yes</td>
<td><a href="http://www.cii-murcia.es">www.cii-murcia.es</a></td>
</tr>
<tr>
<td>Basque Country</td>
<td>29 June 2000</td>
<td>ALI</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td><a href="http://www.ali.es/infcolpv.htm">www.ali.es/infcolpv.htm</a></td>
</tr>
<tr>
<td>Catalonia</td>
<td>9 April 2001</td>
<td>AI2</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td><a href="http://www.cii-murcia.es">www.cii-murcia.es</a></td>
</tr>
<tr>
<td>Murcia</td>
<td>28 April 1998</td>
<td>AI2</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td><a href="http://www.fiv.upv.es/coicv/">www.fiv.upv.es/coicv/</a></td>
</tr>
<tr>
<td>Valencia</td>
<td>19 May 2000</td>
<td>ALI</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td><a href="http://www.fiv.upv.es/coicv/">www.fiv.upv.es/coicv/</a></td>
</tr>
</tbody>
</table>
degrees in non-Informatics subjects to join the College during a transitional period) would allow those Colleges to have more clout by being able to admit more members that are highly qualified. This condition is currently only fulfilled by the Catalonian law, promoted by AI2, which provides for that possibility for the two years subsequent to the law’s coming into force. AI2, who oppose this provision, has lodged an appeal against this law.

4. Conclusions

The large shortfall of Informatics professionals in the European Union, today and in the near future, acts as a brake on the development of the Information Society in our countries. This shortfall has several causes and the EU authorities, as well as its member States, who recognize the gravity of the situation are introducing different measures to bring about a significant reduction in the size of the problem.

In Spain we have the added problem that Autonomous Communities are passing laws to regulate the Informatics profession that, while they have the advantage of strengthening the social fabrics of the Informatics profession, in some cases they have the disadvantage of not being enforceable in practice due to their probable contravention of EU and Spanish legislation, and above all, of going against what the economy and society really need. The divergence between key aspects of the regional laws, such as whether membership should be a compulsory condition for being able to practice in the profession, may even lead to a situation of chaos which will do nothing to enhance the prestige of Informatics professionals in Spain.

References

[ATI]
Professional and associate affairs section of ATI’s website, with extensive documentation on this issue, including ATI’s position in its regard. http://www.ati.es/DOCS/

[Denning 00a]

[Denning 00b]

[ECDL]
European Computer Driving License (ECDL) http://www.ecdl.com

[EITO–EEIG 01]

[European eLearning]

[ITC Consortium]

[Milroy/Rajah 00]

[Union Network International]

Translation by Steve Turpin