REQUIREMENTS ENGINEERING: A CLOSE LOOK AT INDUSTRY NEEDS AND MODEL CURRICULA

Oliver Minor
University of Koblenz-Landau
oliver.minor@web.de
Jocelyn Armarego
Murdoch University, Perth
jocelyn@eng.murdoch.edu.au

ABSTRACT

Research endeavours in software development have found that failures and deficiencies of software systems are often rooted in the requirements activities undertaken. One possible cause for poor requirements activities is the appropriateness of the education of those engaged with the requirements component of software development. This education is largely based on model curricula used as guidelines. This paper examines the requirements component of model curricula in the disciplines of computer science, information systems and software engineering. These are compared to the opinions of a small but representative group of practitioners, assembled through personal interviews. The results reveal that the model curricula address to a high degree the expectations on the formal education preparing for requirements activities practitioners have mentioned. However, the results also show that practitioners see shortcomings in formal education, particularly with respect to more generic skills, such as communication and team skills.

INTRODUCTION

The Information and Communication Technology (ICT) sector is considered to have major relevance for the whole economy of Australia. A recent study states that if a better and increased education in this field were conducted it would have substantial influence on the productivity and the overall performance of the economy (CIE 2001). RE, as a foundational element of the development of computer-based system, is central for ICT.

RE as a fundamental discipline in the development of systems and software has been widely recognised as crucial within the last several years (Alexander and Stevens 2002; Ferdinandi 2002; Hull, Jackson et al. 2002; Young 2002; Bray 2003). As early as 1976 Bell and Thayer observed that inadequate, inconsistent, incomplete, or ambiguous requirements have a critical impact on the quality of the resulting software (cited in van Lamsweerde (2000)).

Surveys and studies underline the pivotal character of Requirements Engineering (Standish 1994; ESI 1996; Al-Karaghouli, AlShawi et al. 1999; Lee, Dutta et al. 1999; van Lamsweerde 2000).

Other studies reveal problems in communication (Al-Rawas and Easterbrook 1996), monolithic and overloaded requirements in Commercial-off-the-Shelf-Software projects (Karlsson, Dahlstedt et al. 2002) or in cultural differences in multi-site software development organisations (Zwoghi, Damian et al. 2001). Yet another research project reveals that contingencies of the project, characteristics of the project managers and the composition should be considered (Carroll and Swatman 1999).

This shows a variety of challenges have to be met, and also reveals opportunities to improve the RE process. To tackle these challenges and make use of the opportunities, novice requirements engineers should be equipped with appropriate skills and knowledge. Yet Conn (2002) reports that it is a *surprise* to graduates that requirements is a major cause for software deficiencies.

Despite the number of books, articles and research findings published, the transfer and adaptation from these sources into practice has been seen as difficult (Nikula, Sjaniemi et al. 2000). Morris et al. (1998) examined through workshops how companies absorb knowledge/knowledge diffuses from the academic world into practice. The workshop participants identified training as a key problem (amongst other problems). A main way of technology transfer into practice is training. Training and education is often based on literature, though Nguyen et al (2002) recognised that the actual practice of requirements engineering does not conform to its presentation in the literature.

Not only publications transfer knowledge into practice but education also has an influence (due to its roots in literature). Lethbridge (2000) surveyed software practitioners in order to find out the relevance of their software engineering and computer science education. 60% of the respondents considered that requirements gathering and analysis is under-taught in education (Lethbridge 1998; 2000). As a result it can be assumed that teaching does not reflect the needs of the practice.

Considering weaknesses in the requirements analysis or requirements engineering, and hints found in surveys about the education in computer science and software engineering as well as information systems, this paper examines the relationship between the opinion of practitioners and current model curricula in the respective disciplines.

Model curricula built the fundament and guidelines for tertiary education. Before describing the research design the model curricula considered in this research are summarised. Based on the research design the results and the implication for education are then presented.

BACKGROUND

Reviewing literature on requirements engineering and systems analysis (Davis 1993; Macaulay 1996; Robertson and Robertson 1996; Kotonya and Sommerville 1997; Leach 1999; Nickerson 2001; Kendall 2002), several skills and knowledge areas arise.

Table 1: Knowledge and Skills Topics

Requirements Engineering Activities
Feasibility Study
Elicitation
Determination
Analysis
Documentation
Verification
Requirements Management
Success factors
Generic Skills
Management of Self
Management of Information
Group/team skills
Management of Tasks
Problem/opportunity identification
General problem-solving strategies
Communications Skills
Cultural insight / Professionalism

Meta-cognitive strategies
Analytical Skills

These areas derive from process activities at the beginning of a software development/enhancement project. Table 1 gives an overview of relevant topics.
This table is neither perfect nor comprehensive nor complete. However, it does give an underlying framework for

- the comparison of the model curricula
- the questionnaire used in the interview study and
- analysing the answers of the interview study.

Formatted: Bullets and Numbering

The table enables a transparent and reproducible process for analysing the curricula.
A curriculum should reflect up-to-date research as well as the practice (Avison, Fitzgerald et al. 2001) in a discipline. Work-groups of ACM and IEEE-CS (among others) have integrated requirements engineering in the model curricula and BOKs of *Computing Curriculum – Computer Science* (CCCS) (Engel and Roberts 2001), the *Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems* (IS2002) (Gorgone, Davis et al. 2002; Gorgone, Davis et al. 2002a) and the *Computing Curriculum – Software Engineering* (CCSE) (Engel and Roberts 2001; Díaz-Herrera and Hilburn 2003). They are part of the so called Computing Curricula (Engel and Roberts 2001) effort. What the curricula have in common is that they present knowledge areas that each graduate of the respective discipline should know to a certain degree.
Several research endeavours have examined the industry expectations of graduates. Doke and Williams (1999) give in their article an overview of published research in the field of information systems. Lee et al. (1995) examined the importance of different topics of information systems with the help of focus group interviews, forums and a survey among practitioners. In another study Noll and Wilikens (2002) examined what information system workers perceive as important skills and knowledge for future employees in information systems. Turner and Lowry (1999) asked students and company representatives, mainly human resource employees, about their perception of what is considered to be required on the job.
Lethbridge (1998; 2000) examined the relevance of computer science, computer engineering and software engineering education. *Requirements gathering and analysis* was ranked among the top five regarding overall importance, although the amount learned by the respondents during their formal education was evaluated rather low. Therefore a significant difference exists between the amount learned in formal education and the current knowledge level: this may indicate that *Requirements Gathering & Analysis* is not considered in formal education to the extent it should.
Macaulay and Mylopoulos (1995) compared courses from ten international (mainly from the UK, but also US, Spain, Canada, Israel) universities and an industrial perspective of requirements engineering.
All the above described studies cover in some way skills and knowledge needs for systems analysis and requirements engineering. However, they mostly only examine the general importance perceived by different stakeholders, such as practitioners, human resource staff or students. None of them examines in detail the activities that are necessary to perform systems analysis/requirements engineering and whether skills and knowledge needed for these activities are reflected in the respective model curricula.
As the *requirements process is a human endeavour* (Kotonya and Sommerville 1997,

p.141) it is highly complex to find out what makes it a successful endeavour. Therefore, it does not seem enough to find the underlying success factors of requirements engineering in asking for the importance of this topic.

It seems rather interesting to search for topics that are crucial for successful requirements engineering. This search has been pursued already; the result of that can be seen in literature, which was presented very selectively in this chapter. The influence of topics that have been identified as crucial for successful requirements engineering on the practical undertaking of requirements engineering and their reflection in model curricula has not been subject of research yet (at least it did not occur to the authors during their background research).

Therefore, this paper tries to answer the following questions:

- Which knowledge and which skills are necessary to successfully conduct requirements activities?
- Which personal characteristics are needed?

In order to find this out it seems necessary to ask how requirements engineering is conducted.

In a second step the answers can be used to examine existing model curricula with respect to their accordance to the given findings.

- How do model curricula guide towards required skills and knowledge?
- Which areas are neglected and which are covered by existing model curricula?

RESEARCH METHODS

Requirements engineering is a real world discipline applied in many areas of contemporary industry and since it involves heavy human interactivity it is suitable to use qualitative research methods (Loucopoulos and Karakostas 1995; Leedy and Ormrod 2001). Teaching is also considered as a process highly involving human behaviour. The analysis of model curricula as the basis of teaching is also of qualitative nature because textual content (data) will be interpreted by means of human thinking and structuring tools.

This research is therefore based on a qualitative approach with a small portion of quantitative analysis when analysing learning objectives. The questions asked as research questions above are aimed at evaluating model curricula.

A three-step process was applied:

- Data gathering
- Analysis
- Presentation of Analysis Results

The data gathering process was aimed at establishing a comprehensive view on (a) what is understood by requirements engineering and (b) practitioners' perception of required skills, knowledge, and personal characteristics.

In order to achieve a basic understanding of requirements engineering the results of a literature review were used. The structure and the content of the questionnaire as well as the analysing frameworks were based on these results. However, it is acknowledged that the view presented in the literature review is limited and biased by the selection and interpretation of authors and their texts. Since the model curricula were also used for designing the interview guidelines certain limitations must be considered.

The practitioners' perceptions were captured with semi-structured personal or telephone interviews. Patton (1990) calls it *general interview guide approach*. Through the interviews opinions and experiences have been explored. Personal interviews have the advantage that complex issues can be examined and discussed. Furthermore, personal interviews raise a more conversation-like interview (Patton 1990).

An initial pilot interview and several informal reviews were made in order to improve the effectiveness of the questions. The pilot interview was used to improve the questionnaire with respect to wording and question sequence.

Thirteen interviews (excluding the pilot interview) had been conducted in six organisations of which two are considered heavy on computer science, two on information systems and two on software engineering. The selection of the organisations was a purposeful sampling (Patton 1990). The organisations appointed staff members as interview partners under the conditions that the interviewees work in the field of requirements engineering and are not graduates of School of Engineering Science at Murdoch University. The interviews were conducted with the ethics approval of Murdoch University Human Research Ethics Committee with the permit number 2003/220. Interviewee and company names are recoded for privacy reasons.

After having set up appointments for the interviews, the interviewees were sent a letter of consent and the skills matrix to give them a first impression of the research. A three-page questionnaire was handed over to the interviewee at the beginning of each the interview.

The interviews were recorded on tape and transcribed afterwards for examination. Two interviewees denied their approval to the tape recording. Some of the interviewees were later contacted for clarification, verification and probing questions via telephone.

A second means of capturing information from the interview participants was a web-based questionnaire (Armarego 2003). This questionnaire was mainly used to verify findings of the interviews. Ten out of 14 (including the pilot interviewee) interviewees filled out the questionnaire.

Besides the interviews and web questionnaire, an internet search was conducted to uncover general company information of the interviewed organisations.

During the data gathering phase a first, mainly implicit, analysis of data was done. The main analysis however, was performed in the aftermath of the data gathering.

The interviews and the subsequent telephone follow-up were transcribed. The analysis of the interview transcriptions was done through a framework analysis, also called template analysis (Richie and Spencer 1994; King 1998; Lacey and Luff 2001).

Identifying the thematic framework (or template (King 1998)) was based on the topics that occurred in the literature and curriculum review and were therefore also represented in the interview questionnaire. Reading, re-reading and listening to the interviews was the core activity for finding statements on the categories identified in the thematic framework. The thematic framework was modified during the course of analysis.

Derived from the structure of the questionnaire three main frameworks were used for coding the data:

- company settings,
- interviewee's education and career path, and
- the interviewee's perception of requirements engineering, the needed skills and knowledge.

After having categorised the data, they were analysed systematically for commonalities, differences and interrelationships.

Framework (3) is also used for analysing the model curricula. That enables the core examination on how practitioners view requirements engineering in comparison to model curricula.

To overcome credibility issue (Patton 1990; Miles and Huberman 1994) in this study data from personal interviews and from an accompanying web survey were used for triangulation.

The content of the answers must be viewed critically. As Argyris and Schön ((1974) referred to in (Anderson 1997)) describe in their work about *Espoused Theory* and *Theory-in-use*, the interviewees' answer may not completely reflect their actions. That means that the interviewee might say something about their requirements activities (*Espoused Theory*) but act differently (*Theory-in-use*).

RESULTS AND DISCUSSION

All companies had an international focus for their software development. Besides two global players with more than 10,000 employees, the companies were in the range between 2,000 and 3,000 (IS1, SE1) or less than 50 (CS1, CS2). The number of people involved in software development at the premises in Perth ranges from 12 up to 200 people.

The companies are involved in such industry areas such as geographical information systems, image-processing software, financial sector, business information systems, defence industry, and telecommunication.

All interviewees had a senior role in their company. They can be classed in the middle management and upper management. The interviewees' involvement with requirements activities can be categorised in *ReceivingDevelopers*, *ActivelyInvolved*, or *Supervision*. *ReceivingDevelopers* primarily receive requirements that they have to turn into design or code. One interviewee oversees these activities, so he can be classed in *Supervision*. The other interviewees are *ActivelyInvolved* in requirements activities. These activities can either involve direct contact with the customers and users of the future system or be through sales and support people.

Ten interviewees have a Bachelor of Science degree or equivalent, one holds a Diploma in Education, one an Associate Diploma in Computing and another did not attend any tertiary institution. The interviewees studied subjects such as Computer Science, Information Science, Software Engineering, Mathematics, Physics, Biology, or Chemistry. Seven interviewees received their degrees (including the Associate Diploma) in Australia, five from universities overseas.

Their practical experience in software related jobs measured in years is between seven and 25 years.

Analysis presentation of interview statements by category

This section presents the analysis results of the statements given by above described interviewees.

Requirements Engineering Process

The interviewees stated that between 5% and 25%, mostly around 10%, of their working time is spent on requirement related topics. Some interviewees see the requirements phase as a *distinct process* (Thomas, SE1). The reason for that might lay in the strict compliance to a standard process. Compliance to process standard is also mentioned: *the one line code-*

change is very controlled (John, SE2).
CS2 does not seem to have a formalised way for requirements,

> *I didn't really write any of this down. It just goes without saying I guess at some level.*
> *(Simone, CS2).*

This shows that there are fundamentally different ways of approaching requirements activities. It varies from much formalised processes with well-defined sign-off points to requirements activities that are more implicit.
Two main demands on curricula could be drawn out of these statements:

- Students should know that the requirements process can vary tremendously.

CCCS (Engel & Roberts, 2001)

demands that students understand the importance of the requirements process. Software engineering students should *comprehend* the process and

> *apply current theories, models, and techniques that provide a basis for problem identification and analysis* (Díaz-Herrera & Hilburn, 2003, p.10).

IS2002 (Gorgone et al., 2002) postulates a knowledge about the life cycle model in general. The model curricula demonstrate certainly theoretically sound processes. They do not mention that real life processes might work differently. It might be a political matter, whether to *poison* students with imperfect processes or to teach them clinical processes.

- Students should understand the sense of process standards.

The curricula CCCS (Engel and Roberts 2001) and CCSE (Díaz-Herrera and Hilburn 2003) demand students to *comprehend* process standards. *Comprehension* means the ability *to grasp the meaning* (Díaz-Herrera and Hilburn 2003). In contrast, IS2002 (Gorgone, Davis et al. 2002) does not mention process standards.

Feasibility Study

Through the feasibility study the economical, political or technical feasibility for a project and parts of it is tested. Most companies have no formalised feasibility study before entering a project. One interviewee describes the feasibility study as

> *looking at the requirements or we are* analysing *whether it [requirement] makes sense (John, SE2).*

It indicates that feasibility is tested with the experience and knowledge in the area where the requirement occurs. Estimation techniques play a role in the feasibility study. Depending on the initially estimated size of the proposed project either an informal estimation or a formal estimation is performed. The informal way of doing can be boiled down to *gut feeling* (John, SE2). In literature it is called *expert judgement* or *educated guess* and relies on experience (Pfleeger 2001). Formally, estimation techniques such as the lines of code method are applied. In other companies they discuss a proposed list of requirements and prioritise them in a common effort of senior software developers and sales and support personnel.

This leads to the conclusion that students should have at least a basic understanding of the feasibility study and its purpose.
The model curricula of computer science and software engineering do not or only mention the feasibility study indirectly. IS2002 (Gorgone, Davis et al. 2002) expects the students to *know* the basics.

Elicitation

The interviewees reported several communication ways over which information about requirements is elicited. These are informal telephone conversations, formal telephone conversations such as customer hotlines or teleconferences, emails, web feedback forms, documents such as existing code, work-groups, JAD-sessions, prototypes, or surveys. Immanuel (SE2) summarises that *research skills* are necessary for eliciting requirements.
All model curricula include elicitation. CCCS (Engel and Roberts 2001) and IS2002 (Gorgone, Davis et al. 2002) expect student to be able to *apply* instruments for elicitation, whereas software engineering students should *comprehend* them. The lower level makes sense when arguing that in companies such as CS1, CS2, IS1 or SE2 the tendency exists that domain experts do some of the elicitation tasks.

Analysis and Determination of Requirements

Domain experts also mostly do the market-oriented evaluation. The interviewees were more involved in the technical evaluation and analysis of the requirements. According to the interviewees, the analysis can be seen as an interactive process in which employees with market competence and those with a technical understanding negotiate requirements.
Difficulties are mentioned,

> *that sort of communication [with the sales and support people] is very, very difficult* (Karl, CS1).

It is even stated to determine the requirements correctly is a matter of *luck* (Karl, CS1). In other environments, *technical people [are] talking to technical people* (Charlotte, IS2). That means that they have to have knowledge about the technology, e.g. architectural issues, they apply or they have to use because of system constraints.
Two main points can be identified:

- Domain knowledge is important.
- Ability to analyse relevant information and communicate with people with a different (not computing) background and with a technical background.

Domain knowledge cannot be considered to be a demanded part of a computing curriculum. Determining requirements is based on the analysis and communication between the involved stakeholders. Analysis techniques such as modelling are considered by all curricula to be learned up to the *application* level. Communications is stressed as an essential issue by all curricula. CCCS (Engel and Roberts 2001) even proposes projects with other disciplines.

Documentation

The documentation of the requirements gathering and analysis results differs in the degree of formalisation. At the informal end of documenting no formal document deliverables are required. At the formal end templates for the documents are given and the documents are formal sign-off points. An informal documentation is described,

It gets up drawn on a white board and people take notes (Simone, CS2)

during meetings. On the other end of the scale, SE1 or IS1 have well defined documents (e.g., Anne, IS1). Obviously, like in the case of IS1 the degree of formality is higher because the activities of requirements and design/implementation are assigned to different teams or even different departments.

The representation of documentation is again manifold. For all companies it depends on the audience they are primarily trying to reach with the documentation. Sometimes they split it into a part for non-technical people and one for people with a computing background (CS1). In all kinds of documentation natural English plays a major role, *we will go down to a literal description* (Charlotte, IS2).

This leads to the demand that students should have the ability to produce documents with a wide variety of representations, such as modelling or natural English. All three curricula expect students to achieve a knowledge level of *application*, which matches the requirements of the practice.

Verification

Documented requirements get tested against the *real* requirements. IS1, IS2, CS2 and SE2 apply formalised reviews, walk-through or prototypes. Besides the knowledge of the techniques for verification a demand for the ability to accept criticism can be derived.

CCCS (Engel and Roberts 2001) and CCSE (Díaz-Herrera and Hilburn 2003) included verification in their curricula matching the described practice. IS2002 mentions verification only in the context of programming (Gorgone, Davis et al. 2002).

Only the curriculum CCSE includes *Individual Cognition* (Díaz-Herrera and Hilburn 2003). Learning about individual cognition helps to recognise personal limits, such as limits of knowledge and skills. Knowing personal limits is a prerequisite to accepting criticism and the development of personal skills and knowledge. Verification activities can involve criticism and suggestion for the improvement of requirements.

Requirements Management

Requirements may change during the course of a project, for example due to changes in organisations or due to legal changes, like one interviewee reports (Sophie, IS1). John underlines the difficulty,

we keep getting requirements almost everyday…
the requirement is the one that changes a lot (John, SE2).

These changes must be managed systematically throughout the development process (Kotonya and Sommerville 1997).

This shows that the ability to handle not only large amounts of information but also changing information is needed. Despite this need, the CCCS (Engel and Roberts 2001) does not cover it. CCSE (Díaz-Herrera and Hilburn 2003) and IS2002 (Gorgone, Davis et al. 2002) expect students to be only *knowledgeable* about requirements management. Here, a mismatch between the practice and the curricula can be seen.

Generic Skills

Table 2 includes some generic skills. The interviews revealed that two issues stand out in that list: communication skills and team skills. All interviewees regarded these two as highly important. Other generic skills were mentioned relatively seldom in the interviews: if these topics were touched it was with minor importance. Therefore, this section concentrates on communication and team skills.
The requirements determination can be described that

> *it's then a back and forth sort of process* (Karl, CS1).

This process involves the negotiation of requirements between, in that instance sales and support people and the software development team. That example can be seen as explanatory. It underlines the demand for negotiation skills, as part of communication skills. Lethbridge (2000) already found that there is a big knowledge gap compared to the perceived importance.
The need for communication skills in requirements activities is expressed as follows.

> *The communications means to be able to talk to people extract stuff out, document it and understand it and agree to it.* (Anne, IS1).

All curricula refer to the need for effective communication skills more than once and emphasise it similarly as the interviewees perceive it.
Team skills are also mentioned by the interviewees and regarded also as generally important. People *don't get pigeonholed* (Charlotte, IS2) in *a strong team-oriented environment* (Arthur, CS1). Again, the model curricula include team issues in their guidelines. Students should learn the *dynamics of working in teams* (Díaz-Herrera and Hilburn 2003, p.25) be able to work in teams through team projects ((Engel and Roberts 2001, p.236), (Gorgone, Davis et al. 2002, LU 80)). All curricula recommend that undergraduate students should participate at least in one team project.

Summary

The above examined topics can be presented in a table. To simplify the representation a scale is applied to evaluate whether the findings match (+), partly match (o) or do not match (-) with the guidelines given in the curricula.

Table 2: Topics match: How do curricula match the needs perceived by the interviewees?

Topics	*CCCS*	*IS2002*	*CCSE*
Requirements Engineering Process	o	-	o
Feasibility Study	-	o	-
Elicitation	+	+	+
Analysis	+	+	+
Documentation	+	+	+
Verification	+	-	+
Requirements Management	-	o	o
Generic Skills			
Communication Skills	+	+	+
Team Skills	+	+	+

The model curricula have some insufficiencies as compared to the interviewees' opinions. All three curricula do not fully match the topics of RE Process, Feasibility Study and Requirements Management. The IS2002 furthermore lacks a sufficient coverage of requirements Verification.
The other topics can be regarded as sufficiently covered by the curricula.

Interviewees' Expectations

The following section discusses what interviewees explicitly expect from graduates for requirements activities.
In all companies it is, as one interviewee put it, *very rare* (Albert, SE1) that newly hired graduates are involved in requirements activities. The interviewees mention that almost exclusively *more senior people* (Eva, IS2) do requirements activities. Some interviewees argue that experience is necessary. That confirms findings made by Macaulay and Mylopoulos (1995). One interviewee expects *credibility and presence* (Thomas, SE1) from somebody doing requirements activities. These characteristics are considered to be reserved to people more mature than most graduates are.
Although requirements engineering is no typical task for graduates the interviews revealed some issues that are of relevance for performing requirements tasks. These issues can be classed into four groups:

- Personality
- Interpersonal Skills
- Technical Skills
- Personal Work Organisation

Interviewees talked about certain *Personality* characteristics that influence the performance of requirements activities positively. General personality qualities are that graduates should be *confident and faithful* (Simone, CS2). Confidence can be underpinned by the knowledge and skills that are required in certain circumstances. To be faithful can be considered as rooted in upbringing. The model curricula include that in ethical concerns (Engel and Roberts 2001, p.64; Gorgone, Davis et al. 2002, LU12; Díaz-Herrera and Hilburn 2003, p.16). The software engineering curriculum notes that confidence and a strong work ethic, also demanded by an interviewee (Sophie, IS1), can only be influenced subtly by university education.

Another general characteristic that is expected is to be *proactive* (John, SE2) or *self-started* (Simone, CS2).

For requirements activities graduates should have an *inquisitive nature* (Anne, IS1). They should have the ability to *ask people questions* (Simone, CS2) and *accept to appear stupid* (Immanuel, SE2). Although the interviewee talks about nature, she thinks that techniques for questioning can be learnt (Anne, IS1). *Perseverance* is also described to be of advantage (Sophie, IS1). These techniques are considered in general in the curricula, as noted above.

Finally, people should be *teachable* and *willing to learn* (Sophie, IS1). The curricula emphasise that need also with respect to the rapid changes in software development (Engel and Roberts 2001; Díaz-Herrera and Hilburn 2003).

Interviewees as well as the literature, in particular literature about systems analysis (Lee, Trauth et al. 1995; Marakas 2001; Hoffer, George et al. 2002), regarded interpersonal skills as important.

Interpersonal skills are also considered to be only teachable to a certain degree. Communication skills are considered as not teachable, by one interviewee, *You've got 'em or you haven't* (Marie, CS2). Other interviewees see the possibility to improve it (Sophie, IS1; René, SE2). To have the ability is considered to be *up to the individual* (Sophie, IS1). The interviewee also says that issues such as communication and team skills can be influenced best when people are *young and amenable* (Sophie, IS1). That leads to the conclusion that curricula have to consider these issues.

Technical skills that are of particular relevance for requirements are mentioned. *Architecture 'cause quite often that comes into play in requirements* (Charlotte, IS2). It must be remarked that architectural issues are a favourite of the interviewee, so a bias might be possible. Furthermore, techniques such as facilitation of groups, estimation techniques and interviewing were mentioned.

Depending on the division of labour, background knowledge about the problem domain is needed to perform requirements tasks. (John, SE2; Sophie, IS1). In the case of IS1 they have a separation between business and technically oriented activities. In SE2 the interviewee meant the knowledge about the technical background in which the piece of software that is to be developed will be integrated. In other cases such as SE1 people, with a computing background tend to perform these tasks. They adopt the domain knowledge.

Personal Work Organisation is a more general issue that is not exclusively necessary for requirements activities but because of the usually high amount of information that must be handled it is considered here.

The general expectations for graduates are in line with a variety of other job profiles. Interviewees expect the combination of good communication skills paired with good team skills and a sound technical understanding. As special for the requirements activities it can be mentioned that the ability to handle large amounts of information is expected.

Implications for Education

As general learning and teaching advice interviewees point out their preference for *more exposure to real life, exercises, team assignments* (Immanuel, SE2) or *industry projects.* Nguyen and Swatman (2000) found that the requirements process as it is described by the literature and therefore taught at universities does not match reality. That can be confirmed by the presentation above. In a subsequent research Nguyen et. al. (2002) postulate that curricula have to take an insight and creativity driven approach towards requirements into account. They demand an educational framework for requirements engineering based on the constructivist learning theory (Hobmair 1994). That includes gaining experience in an *authentic context* (Nguyen, Armarego et al. 2002). That is confirmed by the demands stated by the interviewees. Although the model curricula do not mention learning theories explicitly, they recommend *unsupervised practice* (Gorgone, Davis et al. 2002), a *significant team project* (Engel and Roberts 2001) and projects with a *significant real world basis* (Díaz-Herrera and Hilburn 2003, p.42).

As already mentioned briefly under the item of *Requirements Process* the demand for real-world basis does not reflect the knowledge given in textbooks. Textbooks are usually the basis for tertiary education. Furthermore, teaching with real-world basis does have implications on the education and background of teaching staff and the equipment of the faculties. A shift towards a real-world basis would also mean that other, maybe more underlying and scientific topics have to be shortened.

With the current education, graduates are employed in the companies in a similar way. The interviewees state that graduates usually get tasks such as programming, code inspections or minutes of meetings (Thomas, SE1; Immanuel, SE2).

CONCLUSION AND FURTHER RESEARCH

This paper has examined the relationship between the opinion of a small but representative group of practitioners and current model curricula on the topic of requirements engineering/systems analysis in the disciplines of computer science, information systems and software engineering. These three disciplines are considered to be the most visible of computing disciplines (Glass, 1992). In order to describe that relationship, practitioners were interviewed personally and model curricula were examined. The results of these two activities (interviews and examination of curricula) have then been subject of a qualitative comparison.

The comparison revealed a high degree of conformity between the recommendations of the selected model curricula and the expectations of interviewed practitioners. The conformity relates to the question of which topics students should learn and which level of knowledge they should achieve. These topics have been classed into a group of directly related topics to requirements activities and topics that cover generic skills such as communication or team skills.

The interviewees consider requirements activities such as requirements elicitation, analysis and documentation as regarded appropriately by the curricula. Only the topics of the requirements process, the feasibility study and requirements management can be seen as areas of neglect in the model curricula.

A difference between the perceived importance by the practitioners as well as given importance by the curricula and the awareness of graduates seem to exist. For the CCCS the explanation can be that only 2% of the recommended lessons are dedicated to requirements activities. However, the interviewees did not see a difference between graduates of different disciplines.

The interviewees mentioned weaknesses in the areas of written as well as oral communication and team skills. The discussion of these skills revealed the question whether these skills are learnable and teachable. Interviewees' opinions on that question ranged from learnable and teachable to not teachable and not learnable.

Also, differences seem to exist between the objectives of the curricula and the final employment of graduates. The curricula suggest graduates to be equipped for performing requirements activities. However, new employees usually do not get assigned tasks related to requirements. The reason for that is mainly rooted in a mixture of experience and personality usually only more senior people have.

Although conformity between expectations and the recommended contents exist, graduates appear not to be equipped in an optimal manner to perform requirements activities. In order to find out how to improve the formal education of future employees several recommendations can be made.

In general, the question has to be asked whether formal education is able to produce graduates that are prepared for requirements activities immediately after graduation or experience and on-the-job training is not substitutable. Particular research endeavours can be suggested

- *Recommendation 1* - the above described results should be tested with a larger sample and a broader regional horizon. Such research could also include the question of whether and to what extent differences exist between graduates of different disciplines

- *Recommendation 2* - how the model curricula are applied in actual curricula can be examined. These applied curricula can then be tested for their relevance and effectiveness in practice

- *Recommendation 3* - a third strand of research could examine the teaching and learning methods of the relevant topics. This could include whether certain topics are learnable and teachable in a formal setting.

In order to provide an understanding of under which circumstances the research described was conducted and the results achieved weaknesses and problems must be mentioned. As the research was based on a qualitative approach credibility and interpretation of results can be seen as a problem. Technical weaknesses such as the denied permission to tape the interviews or the selection of interviewees not directly involved in requirements activities did occur. Furthermore, a strict questionnaire was not used and therefore absolutely comparable interviews were not conducted (which can be seen as part of the nature of the qualitative approach).

Despite these weaknesses the results can be used for further developments and improvements of model curricula in the examined disciplines as well as related areas where requirements activities are also of importance (Nguyen and Swatman 2000).

Improvements for the model curricula can be recommended for the topics of the requirements process, the feasibility study and requirements management. These topics are neglected and should be expanded in further curricula revisions. In particular, the CCCS needs to put more emphasis on the importance of requirements activities in general. Although all curricula regard communication and team skills as highly important it does not

seem to be sufficient according to the interviewees' statements.
An improved education in the field of requirements activities could lead to improved development processes and hence, to software systems that do what their users want them to do.

REFERENCES

Alexander, I. and R. Stevens (2002). **Writing Better Requirements**. Harlow, London, New York, Addison-Wesley.

Al-Karaghouli, W., S. AlShawi, et al. (1999). **Enhancing the Understanding of Customer Requirements in Business Software Engineering**, University of Westminster, Brunel University.

Al-Rawas, A. and S. Easterbrook (1996). Communication Problems in Requirements Engineering: A Field Study. **First Westminster Conference on Professional Awareness in Software Engineering,** London, Royal Society.

Anderson, L. (1997). Argyris and Schon's theory on congruence and learning. **2004**.

Argyris, C. and D. Schön (1974). **Theory in practice: Increasing professional effectiveness.** San Francisco, Jossey Bass.

Armarego, J. (2003). **Requirements Engineering Education Relevance Survey** (2003). **2003**.

Avison, D. E., G. Fitzgerald, et al. (2001). "Reflections on information systems paractice, edcation and research: 10 years of the Information Systems Journal." **Information Systems Journal**(11): 3-22.

Bray, I. K. (2003). **An Introduction to Requirements Engineering**. Harlow, London, New York, Addison-Wesley.

Carneson, J., G. Delpierre, et al. (2000). **Designing and Managing Multiple Choice Questions**. Cape Town South Africa, University of Cape Town.

Carroll, J. M. and P. A. Swatman (1999). **Managing the RE Process: Lessons from Commercial Practice**.

CIE (2001). **Breaking the Skills Barrier: demonstrating the benefits of investment in ICT higher education in Australia**. Canberra, Centre for International Economics.

Conn, R. (2002). "Developing Software Engineers at the C-130J Software Factory." **IEEE Software**(September/October 02).

Davis, A. M. (1993). **Software Requirements: objects, functions and states**. New Jersey, Prentice Hall.

Díaz-Herrera, J. L. and T. B. Hilburn, Eds. (2003). Computing Curriculum - Software Engineering - Public Draft 1, The Joint Task Force on Computing Curricula; **IEEE Computer Society, Association for Computing Machinery.**

Engel, G. and E. Roberts, Eds. (2001). Computing Curricula 2001 - Computer Science, **ACM**, IEEE-CS.

ESI (1996). ESPITI Project: European User Survey Analysis,, European Software Institute.

Ferdinandi, P. L. (2002). **A Requirements Pattern: Succeeding in the Internet Economy**. Harlow, London, New York, Addison-Wesley.

Gorgone, J. T., G. B. Davis, et al. (2002a). Appendix 5 - Body of Information System Knowledge. **IS 2002: model curriculum for undergraduate degree programs in Information Systems**. J. T. Gorgone, G. B. Davis, J. S. Valacichet al. Park Ridge (IL), ACM.

Gorgone, J. T., G. B. Davis, et al. (2002). **Model Curriculum and Guidelines for**

Undergraduate Degree Programs in Information Systems, Association for Computing Machinery (ACM), Association for Information Systems (AIS), Association of Information Technology Professionals (AITP).

Hobmair, H. (1994). **Pädagogik**. Köln, Bildungsverlag Eins.

Hoffer, J. A., J. F. George, et al. (2002). **Modern Systems Analysis and Design**. New Jersey, Prentice Hall.

Hull, M. E. C., K. Jackson, et al. (2002). **Requirements Engineering**. London, Berlin, Heidelberg, Springer.

Karlsson, L., A. G. Dahlstedt, et al. (2002). Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study. **Eighth International Workshop on Requirements Engineering**, Essen, Foundation for Software Quality.

Kearsley, G. (2000). **Explorations in Learning & Instruction: the theory in practice database**. Washington (DC), George Washington University.

Kendall, K. E. K., Julie E. (2002). Systems Analysis and Design. New Jersey, Prentice Hall.

King, N. (1998). Template Analysis. **Qualitative Methods and Analysis in Organizational Research**. C. Cassell. London, Thounds Oaks, New Delhi, Sage Publications**:** 118-134.

Kotonya, G. and I. Sommerville (1997). **Requirements Engineering: Processes and Techniques**. Weinheim, Brisbane et. al., John Wiley & Sons.

Lacey, A. and D. Luff (2001). Qualitative Data Analysis, Trent Focus Group. **2003**.

Leach, R. J. (1999). **Introduction to Software Engineering**, CRC Press LLC.

Lee, D. M. S., E. M. Trauth, et al. (1995). "Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academic / Industry Investigation." **MIS Quarterly 19**(3): 313-340.

Lee, M., S. Dutta, et al. (1999). **An empirical Analysis of Software Production Problems in European Software Units**. Fontainebleau, France, INSEAD.

Leedy, P. D. and J. E. Ormrod (2001). **Practical research: planning and design**. New Jersey, Prentice Hall.

Lethbridge, T. C. (1998). **A Survey of the Relevance of Computer Science and Software Engineering Education.** CSEE&T'98, Atlanta, IEEE.

Lethbridge, T. C. (1998). **What Knowledge Is Important to a Software Professional? - A summary of the most important results**. **2003:** 44-50.

Lethbridge, T. C. (2000). "What Knowledge Is Important to a Software Professional?" **Computer 33**(5): 44-50.

Loucopoulos, P. and V. Karakostas (1995). **System Requirements Engineering**. London, McGraw-Hill.

Macaulay, L. and J. Mylopolous (1995). **Requirements Engineering: An Educational Dilemma. 2003**.

Macaulay, L. A. (1996). **Requirements Engineering**. Berlin, Heidelberg, New York, Springer-Verlag.

Marakas, G. M. (2001). **Systems Analysis and Design**. New Jersey, Prentice Hall.

Miles, M. B. and A. M. Huberman (1994). **An Expanded Sourcebook Qualitative Data Analysis**. Thousand Oaks, London, New Delhi, Sage Publications.

Morris, P., M. Masera, et al. (1998). Requirements Engineering and Industrial Uptake. **Third IEEE International Conference on Requirements Engineering**, Colorado Springs, Colorado, IEEE Computer Society.

Nguyen, L., J. Armarego, et al. (2002). **Understanding Requirements Engineering: a**

Challange for Practice and Education. School of Management Information Systems. Melbourne, Deakin University.
Nguyen, L. and P. Swatman (2000). **Managing the Requirements Engineering Process**. **2003**.
Nickerson, R. C. (2001). **Business and Information Systems**. New Jersey, Prentice Hall.
Nikula, U., J. Sjaniemi, et al. (2000). **A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises.** Lappeenranta, Telecom Business Research Center Lappeenranta.
Noll, C. L. and M. Wilikens (2002). "Critical Skills of IS Professionals: A Model for Curriculum Development." **Journal of Information Technology Education 1**(3): 143-154.
Patton, M. Q. (1990). **Qualitative Evaluation and Research Methods**. Newbury Park, London, New Delhi, Sage Publications.
Pfleeger, S. L. (2001). **Software Engineering: Theory and Practice**. New Jersey, Prentice Hall.
Richie, J. and L. Spencer (1994). Qualitative data analysis for applied policy research. **Analysing Qualitative Data**. Burgess. London, Routledge**:** 173-194.
Robertson, J. and S. Robertson (1996). **Vollständige Systemanalyse**. München, Wien, Carl Hanser Verlag.
Sly, L. and L. J. Rennie (1999). Computer managed learning: its use in formative as well as summative assessment. **Third Annual Computer Assisted Assessment Conference**, Loughborough, Loughborough University.
Standish (1994). **The Chaos Report.** West Yarmouth (Mass), Standish Group International, Inc.
Turner, R. and G. Lowry (1999). Educating Information Systems Professionals: Towards a Rapprochement Between New Graduates And Employers. **10th Australasian Conference on Information Systems**, ACIS.
van Lamsweerde, A. (2000). "Requirements Engineering in the Year 00: A research perspective." **ACM**.
Young, R. R. (2002). **Effective Requirements Practices**. Boston et al., Addison-Wesley.
Zwoghi, D., D. Damian, et al. (2001). Field Studies of Requirements Engineering in a multi-site software development organization: research in progress. **6th Australian Workshop on Requirements Engineering (AWRE'01)**, Sydney.