

Towards a competency model for requirements analysts

Ruth Klendauer,^{*} Marina Berkovich,[†] Richard Gelvin,[§] Jan Marco Leimeister[¶] & Helmut Krcmar[‡]

^{*}TUM – Technische Universität München, Department of Sociology, Lothstr. 17, 80335 Munich, Germany, email: klendauer@wi.tum.de, TUM – Technische Universität München, Chair for Information Systems, Boltzmannstr. 3, 85748 Garching, Germany, email:

[†]berkovic@in.tum.de, [‡]krcmar@in.tum.de, [§]Independent Software Consultant, West Portland Street, Troon, KA10 6AB UK, email: gelvin@gmx.org, and [¶]Kassel University, Chair for Information Systems, Nora-Platiel-Str. 4, 34127 Kassel, Germany, email: leimeister@uni-kassel.de

Abstract. *Requirements engineering is a software development discipline, executed by requirements analysts (RAs), that includes requirements elicitation, analysis, specification and validation. Its successful outcome is very often essential to overall project success. However, there is a lack of systematically conducted empirical research on the competencies of RAs. This paper addressed this gap by conducting 64 interviews at eight major North American and European financial services companies. Our qualitative research design follows an interpretive approach and uses critical incident technique. We develop a competency model, which specifies 16 critical competencies, and integrates contextual and situational factors as well as results variables. ‘Consulting others’, ‘Testing assumptions and investigating’ and ‘Explaining concepts and opinions’ were the most frequently identified competencies. This indicates that for an effective analyst, close interaction and communication with customers is indeed crucial – but of equally importance is the critical questioning of the expressed needs. Surprisingly, applying specific tools and advanced techniques did not seem to play a significant role from the interviewees’ perspective. This study contributes to theory as it is the first to elaborate a competency model for RAs. It also provides a foundation for the development of competency-based training in companies and universities.*

Keywords: requirements engineering, competency model, critical incident technique, software development

The results of this paper have been developed in the collaborative research centre SFB 768 – cycle management of innovation processes – supported by the DFG. For further information see <http://www.sfb768.de>. Parts of this research article have been supported by the HWP program of the LMU Munich.

INTRODUCTION

Research has indicated that a large proportion of the problems in software development can be traced to requirements engineering (RE) (Byrd, Cossick, & Zmud, 1992; Hofmann & Lehner, 2001; Lamsweerde, 2009). Requirements refer to the requisite functions or characteristics of a system. RE 'is the process of discovering the purpose [of the software], by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation' (Nuseibeh & Easterbrook, 2000).

The set of the activities included in the RE process and the terms by which they are known varies according to the methodology used in the project. However, the workflow is generally accepted to include elicitation, analysis, specification and validation (Sommerville & Kotonya, 1998; Abran & Moore, 2004). Requirements elicitation concerns the discovery of the requirements from sources, such as the project stakeholders, organisational documentation or, as is often the case in a redevelopment project, from the existing system specifications. In the analysis and negotiation activity, the various stakeholders examine the elicited data, and then discuss and agree upon the set of, often prioritised, requirements to be implemented in the system under development. Requirements specification concerns the chronicling of the requirements in the software requirements specification (SRS), which contains all the requirements for the system. In the validation activity, the specified requirements are verified to ensure that there are no incorrect, ambiguous or contradictory requirements, and to ensure that they adhere to organisational quality standards.

Failures during the RE process have a significant negative impact on the overall development process (Finkelstein & Dowell, 1996; Boehm & Basili, 2001; Hall, Beecham & Rainer, 2002). Reworking requirements failures may take 40% of the total project cost (Aurum & Wohlin, 2005). If the requirements errors are discovered late in the development process, e.g. during maintenance, their correction can cost up to 200 times as much as correcting them during the early stages of the development process (Niazi & Shastry, 2003). Adequate requirements are essential to be sure to build the system that the customer expects and that unnecessary efforts are avoided (Hall, 1997). As a result, the requirements analyst (RA) plays one of the key roles in ensuring project success. According to Young (2003), the RA – also called requirements engineer, requirements manager, business analyst, system analyst or simply analyst – is the organisational role that has the primary task to elicit, analyse, validate, specify, verify and manage the needs of the project stakeholders. The analyst's responsibility is to assure that all requirements are prepared in a form that the software development team can understand them without any knowledge about the customer's domain (Hull, Jackson & Dick, 2004). Previously, in many organisations, the role of the RA was not defined clearly. The division of responsibilities, tasks and activities varied depending on the organisational structure, project circumstances or personal capabilities (Aurum & Wohlin, 2005). Software engineers frequently adopted the RE tasks. They were then usually overstrained because of the variety of tasks to fulfil and the different skill sets needed. Many organisations have, therefore, formally introduced the role of the RA in recent years. Based on a literature review, Al-Ani & Sim (2006a) conclude that the appropriate competencies of an analyst are essential for the

successfulness of a project. Also, incorrect assumptions about the competencies needed can result in severe problems in project execution.

However, the question 'What competencies are crucial for RAs to effectively perform their tasks?' has not yet been answered by systematically conducted empirical research (Al-Ani & Sim, 2006b; Paech, 2008). The goal of this paper is to present an empirically based competency model for RAs derived from a series of in-depth interviews and content analyses. We define competencies as 'sets of behaviors that are instrumental in the delivery of desired results or outcomes' (Bartram *et al.*, 2002), thus focusing on the performance-based competency approach. The model specifies the most critical competencies for the RA, and integrates contextual and situational factors as well as results variables, thus extending recent research on competency modelling. It is mainly limited to organisations that have established the formal role of the RA. In the following sections, we describe the background of the study, our research questions and the methodology used. The main results are then presented, and we conclude by discussing the theoretical and practical implications of our findings.

BACKGROUND

Concepts of competency

The concept of competency is used in various areas of research, including psychology, education, management, human resources and information systems (Bassellier, Reich & Benbasat, 2001). The main competence movement started in the 1970s, primarily triggered by the influential paper 'Testing for Competence Rather than for Intelligence' (McClelland, 1973). It was argued that traditional intelligence and aptitude tests were poor predictors of high job performance or success in life, proposing competency testing as a viable replacement. Since then, the use of competencies has gradually become widespread in organisations (Boyatzis, 1982; Shippmann *et al.*, 2000).

Despite its increasing popularity, there is no standard definition of the term competency, not even in human resource management research and/or practice (Shippmann *et al.*, 2000; Stoof, Martens, Merriënboer & Bastiaens, 2002; Markus, Cooper-Thomas & Allpress, 2005). It is used in many different ways within and across research disciplines, thus creating some confusion in the literature about the concept and its dimensions (Bassellier, Reich & Benbasat, 2001; Skulmoski & Hartman, 2010). In general, there are two ways of inferring competency: performance-based and attribute-based (Napier, Keil & Tan, 2009). The former focuses on skilful practice, i.e. the ability of the individual to demonstrate job 'know-how'. Competent RAs can be identified through effective actions. The attribute-based perspective concentrates on 'skills-as-attributes', i.e. the extent to which the individual has acquired the necessary set of knowledge and personal characteristics. Competent RAs can be identified through their 'know-what'. Napier, Keil & Tan (2009) acknowledge that both approaches can contribute to research.

In this study, we adopt a performance-based perspective and define competencies as 'sets of behaviors that are instrumental in the delivery of desired results or outcomes' (Bartram *et al.*,

2002). Competencies then refer to behavioural repertoires, i.e. the range and variety of behaviour one can perform, and outcomes one can achieve. Competencies relate to how knowledge and skills are used in performance, and about how knowledge and skills are applied in the context of some particular set of job requirements. According to Bartram and colleagues, 'a competency is not the behavior or performance itself but the repertoire of capabilities, activities, processes and responses available that enable a range of work demands to be met more effectively by some people than by others' (Kurz & Bartram, 2002). The behaviours that an employee has to be skilled and adept at are not identical with performance as the latter relates to a stream of behaviours that will be judged overall as '(in)effective' or '(un)successful'.

This behavioural view of competencies has, in contrast to the attribute- or trait-based approach (Boyatzis, 1982), the advantage of being able to clearly describe the relationship between competencies as constructs on the one hand, and psychological constructs such as motives and personality traits on the other hand. For example, it is common in human resource management to predict job or training performance through (standardised) ability and personality measures (Hunter & Hunter, 1984; Hurtz & Donovan, 2000). Competencies, when defined in terms of observable workplace behaviours, can provide the basis for a differentiated measurement of performance. If, on the contrary, personal characteristics are operationalised as parts of competencies (e.g., Lucia & Lepsinger, 1999), predictors and performance measures are mixed up, and the work of human resources researchers and practitioners is restrained, thus decreasing the value of personnel selection and development.

Models of competency

In general, competency models simply consist of a (hierarchical) list or catalogue, describing desirable competencies (Mirabile, 1997; Lucia & Lepsinger, 1999; Markus, Cooper-Thomas & Allpress, 2005). They usually include operational definitions for each competency, together with measurable or observable performance indicators against which to evaluate individuals. Depending on the desired application (e.g. performance management vs. succession planning, staffing or 360-degree feedback), the list of competencies can be described at different levels of detail.

Research, however, has indicated that models of competency or performance need to contain not only competency components but also situational variables and outcome criteria. For example, Russell (2001) concluded in his empirical study that future research must examine which situational circumstances (such as group characteristics) moderate the relationships between executive competencies and performance, as exploratory cluster analysis suggested a main effect for situations on performance. Tett & Burnett (2003) emphasised the situational specificity of personality-job performance relations and proposed an interactionist model of job performance that distinguishes among five situational features relevant to trait expression, operating at task, social and organisational level. Hollenbeck, McCall & Silzer (2006) took a critical view of the current state of theory and practice with regard to leadership competency models. The authors expressed the importance of building a more comprehensive, integrated model of leadership effectiveness that encompasses behaviours, situations

and outcomes. They suggested developing separate models for (a) leadership situations; (b) leadership outcomes; and (c) interactions among competencies, situations and outcomes. Alternatively, current competency models could be expanded by specifying the mix of competencies for the most critical job roles and situations in an organisation. As the vast majority of the work on defining models of job performance has focused on management functions, the need for a more comprehensive model can be generalised to other functional areas (such as RE). In a similar vein, Bartram, Robertson & Militza (2002) addressed some of the shortcomings in the literature in their generic theoretical model of competency by including context factors (variables that facilitate or inhibit desirable behaviours, e.g. reward systems, physical environment) and results (the outcomes or goals of behaviour that have been defined either explicitly or implicitly by the person, line manager or organisation).

Overall, studies on job performance in human resource management have concentrated on the predictor side in terms of ability tests, motivation or personality measures. Researchers, however, expressed the need for diverting more attention from prediction towards understanding and explanation of the criterion construct individual job performance, which refers to scalable actions, behaviour and outcomes that employees engage in or bring about that are linked with and contribute to organisational goals (Viswesvaran & Ones, 2000). Detailed studies on specific competencies and results are also further needed as recent research has indicated that these constructs are superior to the widely used general performance ratings (Bartram, 2005). The latter are usually assessed by the variable 'overall job performance' (OJP) through questions, such as 'After considering everything you know about person X, how would you rate his/her overall performance?'. A more differentiated conceptualisation of the criterion domain allows better prediction of job performance once the competency requirements of a particular job or project role are understood. Predictors (e.g. personality measures) and criteria (i.e. competencies) can then be matched on a one-to-one basis, drawing upon previous (meta-analytic) research. In contrast, considering job performance in an undifferentiated manner (such as OJP ratings) hides the pattern of relationships among ability tests, motivation or personality measures and more specific competency factors.

Literature on competencies for RAs

For the literature review, we chose a concept-centric approach, meaning that the analysed papers were assigned according to their concepts (Webster & Watson, 2002). The goal of the review was to get an overview of existing work on competencies needed by the RA and on the research methods used to explore these competencies. The selection of literature for our analysis relied on multiple sources. With regard to the software engineering discipline, we analysed the publications in A-journals and A-conferences according to the WI-lists (2008), and also the Requirements Engineering Journal, Requirements Engineering conference and International Conference on Software Engineering in the years 1999–2010. To find the relevant publications, we examined journals' and conference proceedings' table of contents by the composition of the keywords 'requirements engineering analyst', 'requirements engineer', 'software engineer' and 'skills', 'competence', 'competency/competencies'. We took a broader

approach when conducting the literature review, as the meanings of the concepts skill, competence and competency vary, and are sometimes used interchangeably in the literature. Moreover, even though our definition of competency focuses on behaviour, it also includes how knowledge and skills are used in performance. We identified 68 papers, of which 36 were regarded as relevant. Furthermore, we examined 'Google scholar' for relevant papers by the keywords described earlier. Also the top five books on RE according to amazon.com (accessed on February 10, 2010) were included in the literature review.

Different approaches in the academic literature describe the analyst's competencies and expertise. Al-Ani & Sim (2006a) state that depending on the type of RE activity, different expertise is needed. The authors define five RE activities (requirements elicitation, analysis, communication, validation and management) and six levels of expertise (naïve, novice, beginner, professional, expert and elite). In their model, a set of required expertise is assigned to each activity. Similarly, Penzenstadler, Haller, Schlosser & Frenzel (2009) developed a categorisation of soft skills – defined as communicative abilities for interacting with other people – based on a literature review. For each RE activity (elicitation, documentation, reviewing, etc.), they derived skills that are needed for that activity. The skills are categorised into psychological core identities, sociocultural competencies and professional competencies.

In another paper, Al-Ani & Sim (2006b) introduce different types of knowledge of RAs: practical skill, theoretical knowledge and problem domain knowledge. They also propose a framework for analysing methods and tools based on the expert knowledge of RAs. Young (2003) distinguishes between two different types of analysts: mid-level analysts with two to four year of experience who have knowledge on the RE activities and some experience in applying them, and senior-level analysts with more than five years of experience who have practical experience with all RE methods and also have additional interpersonal skills. Becker, Carmel, & Hevner (1993) state that during requirements elicitation, the communication skills of the analysts are of extraordinary importance. This insight is summarised by Hevner & Mills (1995) as follows: 'Requirements determination entails close cooperation between the system development team, the customer, and system users. Behavioral skills are just as important as technical skills in order to get the system requirements "right" '.

Furthermore, the tasks an RA must fulfil are elaborated in the literature (Alexander, 1998; Young, 2003). The analyst must elicit requirements, identify and resolve conflicts between requirements, and he or she must be familiar with different types of requirements, with criteria for good requirements and with tools for RE. It has to be noted that these papers do not rely on empirical studies to reason about the competencies of RAs. We found many papers relying on literature reviews to categorise different types of skills, e.g. Kovitz (2003) and Maiden (2009) analyse the needed skills depending on project parameters.

A special competence of an RA that is emphasised in the literature is that of decision-making. In RE, a large number of decisions have to be taken. This process has to be supported by the RA (Curtis, Krasner & Iscoe, 1988; Alenljung & Persson, 2008). Andrew, Buchan & Petrova, K. (2009) and Davis & Hickey (2002) report that RAs often lack the appropriate knowledge of trends and needs in practice. In a case study, Alenljung & Persson (2008) conducted interviews to understand the RE decision-makers' decisions in several projects. One of their primary results

is that RAs often lack expert knowledge about the products and the domain for which the requirements are elicited. As a consequence, developers tend to hold back information because they fear that the RA may distort the information. Often, RAs are not trained well enough so that they are not able to support the requirements' activities in an adequate way.

Curtis, Krasner & Iscoe (1988) conducted case studies using interviews with systems engineers, senior software designers, project managers, division general managers, customer representatives, testing and quality assurance team leaders. The main result of their study is that the communication between RAs and stakeholders, as well as between RAs and developers, has to be enhanced. Also, Kovitz (2003) and Paech (2008) emphasise the importance of communication skills of RAs. Communication problems between the analyst and the user are common in practice (Davis, 1982; Browne & Ramesh, 2002).

Another set of papers report on the education of RAs. Some focus on the enhancement of education methods to attain an education adequate for practice (Barnes, Gause & Way, 2008; Regev, Gause & Wegmann, 2008; Andrew, Buchan & Petrova, K., 2009). Beatty & Alexander (2008) developed a method to enhance education by the use of RE games. In their learning methods, they focus on improving the skills in visualisation techniques. During elicitation workshops, an RA must be able to select an adequate visualisation technique and deploy it properly. Furthermore, the education focuses on analytical skills, like understanding, summarising and categorising stakeholder statements.

Moreover, we conducted a literature search in Psycinfo, the premier database in psychological, social, behavioural and health sciences. As keywords, we used combinations of the terms 'requirements engineering', 'software engineering', 'software development' and 'software design' on the one hand, and 'skill*', 'competenc*', 'expert*' and 'performance' on the other hand. We limited our search to the years 1990–2010, peer-reviewed journal articles and book chapters. Only three of the 76 results were relevant to the topic discussed in this paper.

Sonnentag (1995; 2000) compared moderate and excellent software engineers by conducting interviews and analysing their daily work activities. When peers were asked to nominate and describe an excellent performer, they frequently mentioned competencies referring to cooperation and communication. Overall, excellent software engineers were described as having high technical and computational knowledge, a high level of social skills and as using a method-oriented working style. They had a broader, not longer professional experience. Statistics showed that excellent and average software engineers did not differ with respect to time spent on typical software development activities, such as design, coding or testing, but excellent software engineers were more often engaged in review meetings and consultations than other team members.

In their review of empirical research on expertise in software design, computer programming and related tasks, Sonnentag, Niessen & Volmer (2006) drew on literature mainly from cognitive psychology and also from the software design literature within computer science. The authors described the following differences between experts and non-experts in the area of 'RE and design tasks'. Experienced software designers spend more time on clarifying the program requirements compared with students (Jeffries, Turner, Polson & Atwood, 1982; Batra & Davis, 1992). In contrast, no differences were found between high and moderate performers

in analysing requirements in early phases of the design process. Moderate performers spent even more time analysing requirements in later stages of the design process compared with high performers, who might have developed an adequate problem representation earlier. In summary, expertise in software design (including RE) was characterised by more local planning activities, more feedback processing, less task-irrelevant cognitions, more solution visualisations and more knowledge about design strategies (Sonnentag, 1998). Differences between top and average performers could not be attributed to cognitive ability, whereas top performers considered interpersonal network abilities as highly important (Kelly & Caplan, 1993). However, it remained unclear to what extent the results of these studies are valid specifically for activities in RE.

Overall, there are several shortcomings in the current literature on expertise in software design (including RE) (Sonnentag, Niessen & Volmer, 2006):

- 1 The main body of research has focused on cognitive processes without taking into consideration real-world settings; in general, communication and coordination processes have rarely been systematically studied.
- 2 Despite a general consensus about the definition of expertise as 'outstanding performance' (Ericsson & Smith, 1991) at the conceptual level, expertise has been operationalised as years of experience in most empirical studies; novice students were usually compared with more advanced students (or professionals). However, long years of experience are not necessarily related to a high performance level.
- 3 There is a lack of research with regard to possible moderators, such as task or situational characteristics; most studies used simple tasks, taking less than two hours to accomplish, with questionable external validity. There is a need for more complex real-world tasks and situations that require the coordination and prioritisation of subtasks in the context of multiple constraints (e.g., economic, ergonomic and domain-specific).

All in all, there are several research papers about the competencies needed by RAs to effectively perform their tasks. However, little empirical, systematic evidence on this matter has been collected to date. Moreover, there is a lack of research with regard to the relevance of contextual or situational factors when analysing the appropriate competencies of an RA.

RESEARCH QUESTIONS

Our empirical study addressed the following research questions:

- 1 What competencies are critical for an RA to effectively perform his/her role in organisations?
- 2 What are typical and important work situations for the RA? What are the antecedents of these work situations (context)?
- 3 What are the results of the (in)effective handling of the work situations by the RA?

The research questions are illustrated by the conceptual framework depicted in Figure 1.

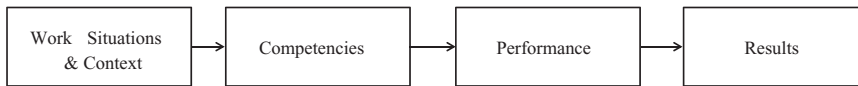


Figure 1. Conceptual framework.

METHODOLOGY

In this section, we describe the research design and sample of our study based on the classification scheme proposed by Paech, König, Borner & Aurnum (2005), which is intended to function as a framework for interviews or questionnaires on RE. The main categories of this scheme are (a) set-up of the study; (b) information on the context and background of the sample; (c) general information on RE process; and (d) information on specific parts of the RE process. Furthermore, the procedure for coding the data is explained in detail.

Set-up of the study

First, the purpose of the study has to be determined. We aimed at conducting a prescriptive study by analysing effective and ineffective behaviour of the RA. Our research approach can be categorised as interpretive research with the goal of gaining a deep understanding of the contextual factors and the interviewees' construction of (in)effective behaviour (King, 1996; Silverman, 1998). In this qualitative study, open-ended questioning and non-numeric analysis were used. The unit of analysis was the critical incident reported by the participants. When comparing many incidents in a certain area, the emerging concepts and their relationships can be seen as probability statements. In our research study, the critical incident referred to a work situation. We conducted semi-structured interviews, following an interview protocol with a list of questions and topics to be covered, while still allowing for the flexibility to discuss further relevant topics that arose during the interview. This technique allows the interviewees the freedom to express their views in their own terms, helps avoid misunderstandings and can create reliable, comparable qualitative data.

We used the Critical Incident Technique (CIT) to derive all critical competencies (Flanagan, 1954; Koch, Strobel, Kici & Westhoff, 2009). Interviewees are asked to recall a typical, important work situation in the context of RE, which they have either personally experienced or observed and that has, in their opinion, been effectively or ineffectively performed by the RA. Interviewees are also asked the following subquestions: (1) What led to this situation?; (2) What happened exactly in this situation?; (3) How was the concrete behaviour of the analyst and how did he/she proceed in detail?; (4) What were the consequences of the analyst's behaviour regarding the concrete work situation?; (5) What relevance did the effective problem-solving have for the further advancement of the work?

The interview technique, thus, captures the subjective perceptions of key stakeholder groups. As Napier, Keil & Tan (2009) point out, this is a common feature of studies that investigated competency requirements in the area of information technology (IT). According to

the authors, this approach has several benefits. For example, it is useful for developing a lay theory ('implicit theory') (Cammock, Nilakant & Dakin, 1995). Also, individual perceptions influence behaviour. That is, if there is a match between the ideal and perceived competency repertoire, the individual is more likely to cooperate with that person. As RAs are working with a variety of stakeholder groups, it would be valuable to understand the expectations of each group concerning the adequate behaviour of the RA. Even more importantly, inferring effective behaviour of the RA from 'objective' measures of project success and vice versa would have several shortcomings. There are many reasons why projects succeed or fail, and even though the role of the RA is critical, there are, for example, external context factors such as market conditions that can significantly influence the project. Besides, the evaluation of a software development project is sometimes shaped by political reasons rather than effective behaviours of the key actors. It is, therefore, essential to assess the concrete behaviours of the RA and their specific consequences ('results') to derive the analyst's critical competencies. The CIT requires a detailed description of a specific project on the part of the interviewees instead of solely asking about their opinion with regard to competency requirements. It, thus, allows the reconstruction of the particular events. Moreover, in the data analysis phase of our research, subject matter experts assign the RAs' behaviours to competency components based on a systematic approach, using specific text analysis software.

To determine how many people needed to be sampled, we used the concept of theoretical saturation (Glaser, 1992). Saturation is given when the collection of further data yields no additional information to the properties of the categories already developed. It fosters general confidence and dependability in the study findings.

Information on the context and background of the sample

Sixty-four interviews (around two hours each) at eight companies have been conducted with RAs and other practitioners, whose job roles bring them into direct contact with, and/or have dependency upon, the requirements activities and resulting artefacts (e.g. SRS). Specific project roles, which determine the viewpoint and the degree of involvement in the RE activities, included RA, IT project manager, user/customer, quality expert, architect and developer (see Table 1). The selection of the interviewees was based on their expertise and years of

Table 1. Project roles of interviewees

Roles	Frequency
Requirements analyst	16
User/customer	16
IT project manager	15
Developer	11
Quality expert	4
IT architect	2

IT, information technology.

Table 2. Project type

	Frequency
New development	36
Enhancement	15
Redevelopment	4
Technical migration	1
Missing value	4

experience, the mean value of which was 14 years in our sample. In addition, we aimed at interviewing people who had been involved in the same projects in order to acquire multiple perspectives. The interviewees' nationalities were US-American, German, Swiss, and East-European.

The sample consisted of eight North American and European based financial services enterprises with in-house software development (four Fortune Global 50, four Fortune Global 500); because of our strict confidentiality rules, it is not possible to provide any further company details. Our main selection criterion was that the organisations had already established formal roles for RE. For methodological soundness, we focused on one business sector, the financial services sector. This was selected because of the complexity of the business domain, a heavy reliance on complex software systems, and the challenge inherent in communicating and understanding between the business and technology domains. Typically, around eight interviews were conducted per company by two interviewers with a background in psychology and software engineering, respectively; this is, of course, not a representative sample. The goal of the projects was primarily to produce new software without using commercial off-the-shelf software (see Table 2). In 36 cases, interviewees described effective behaviours of the analyst, which did not automatically mean that the overall projects were perceived as successful. In 28 cases, interviewees described their perceptions of ineffective behaviours of the analyst, which was not always connected with overall failures of the projects. The project customers were in all cases business units internal to the company. The number and type of users varied from a small group of investment bankers to thousands of branch employees. The average duration of the projects was 2.3 years.

General and specific information on the RE process

We asked about the usage of standard software development processes, RE tools and techniques. In most cases, a proprietary (homemade) process did exist, but the extent to which the defined process was adhered varied greatly within the organisations and depended primarily on the project manager. In all participating companies, defined role(s), responsible for the RE activities in a project, existed. Two companies, in an effort to address the barriers between the business and technology domains, had created specific units to act as the primary interface between the business and IT departments. In another company, two distinct roles

existed: business analyst and business systems analyst. Overall, specific RE tools were rarely used and only basic RE techniques were applied.

Coding procedure

The majority of interviews were recorded and transcribed. Where recording was not possible, the co-interviewer took detailed notes. We coded the transcripts using the software MAXQDA (VERBI GmbH, Marburg, Germany) to conduct content analysis. Overall, coding was very helpful to organise and analyse the data. Although we asked the interviewees all subquestions (e.g. 'What led to this work situation?') together with the main question about a critical incident ('Please recall a typical and important work situation . . .'), interviewees often did not describe the incident in chronological order, and instead skipped back and forth from one stage of the software project to another. We then asked follow-up questions, taking into account the temporal progression of the project and the selective, gradually emerging recollections of the interviewee.

We started with open coding to commence the unrestricted labelling of data, and to assign representational and conceptual codes. Similar incidents and phenomena were then compared and contrasted with each other, and where found to be similar were correspondingly coded. With newly added data, iterative reflection of the already coded data was carried out. For example, we conducted several iterations to narrow down the level of granularity of work situations and results (i.e. consequences of the analysts' behaviour). Subsequently, we compared existing competency models with our initial codes to select a model that fitted the data well. The best match was the SHL Universal Competency Framework (Bartram, 2005), which is a means of developing tailored, individual competency models that are linked to a common, generic, foundation. This framework was derived from an analysis of a wide range of both academic and practitioner models, covering managerial and non-managerial positions. It consists of three hierarchical levels, with 112 component competencies at the finest level of detail, i.e. they can be considered to be competencies broken down to the point where no competency is subsumed by any other competency. These component competencies provided the basis for further coding of the competency data. The competencies of the framework contained detailed definitions, including behavioural indicators, which were adapted to the role of the RA. However, our coding rule was to not impose any existing competencies of the universal competency framework on the data if there was no good match.

Overall, choosing the SHL Universal Competency framework had the following advantages:

- 1 It builds on and moves ahead of the current state of the art in competency modelling in that it supports a more structured approach. Instead of a pure collection of competencies, it offers a genuine framework: (a) it is an articulated set of relationships; (b) it defines the nature of the components of a model; (c) it specifies how those components relate to each other; and (d) it is not just based on content analysis but evidence-based and psychometrically meaningful (due to factor analysis, validation studies, etc.).

- 2 The framework specifies the generic 'deep structure' of the competency domain. Rather than merely describing a set of behaviours as, for example, 'adapting and coping', it uncovers what this actually means through several layers of competency components that make up that set of behaviours. For example, the competency factor 'adapting & coping' can be broken down into subdimensions (e.g. 'adapting & responding to change', 'coping with pressures & setbacks'), which in turn are defined by specific components. Examples for specific behaviours that define the component 'adapting' are 'adjusts to change positively', 'modifies approach in the face of new demands'.
- 3 It offers a means of growing structural intellectual capital by providing the potential for comparing competency models across jobs and industry sectors.
- 4 The framework specifies how the competencies relate to motivational, ability and personality factors (see Figures 2 and 3).

Upon the identification of the core codes in our research study, all interview data were selectively coded by two raters (with an educational background in work psychology and business informatics, respectively). They independently coded the data according to the preliminary definitions of the core codes and code examples. Only in a few cases did inconsistencies exist and, when this occurred, the two raters discussed and refined the codes until mutual agreement was reached.

RESULTS

We identified 16 competencies that are critical for the role of the RA (Table 3; 16 competencies that had an overall frequency of \leq two were omitted). As can be seen in Figures 2 and 3, these competencies are not clustered around one or two of the competency factors at the top level, but rather cover six of the eight high-level competency factors. Thus, RAs need to have or develop a wide variety of different competencies. No significant differences existed with regard to participants' nationalities or project roles – with the following exception: users very often described 'Focusing on customer needs and satisfaction' and 'Consulting others' as effective behaviour of the RA, whereas they never mentioned 'Testing assumptions and investigating'. The latter was mainly depicted by RAs.

The five most frequently described competencies 'Consulting others', 'Testing assumptions and investigating', 'Explaining concepts and opinions', 'Working systematically' and 'Driving projects to results' can be assigned to the four high-level competency factors: 'Supporting & Cooperating', 'Analyzing & Interpreting', 'Interacting & Presenting' and 'Organizing & Executing'. The associated underlying ability and personality factors are agreeableness (the extent to which a person is good-natured, helpful, trusting and cooperative), general mental ability (general intelligence), extraversion (the extent to which a person is sociable, talkative, lively, active and excitable) and conscientiousness (the extent to which a person is organised, careful, self-disciplined and responsible) (see Figures 2 and 3).

On average, important work situations described by the interviewees required three competencies to manage the situation effectively. Thus, job effectiveness in RE often depends on

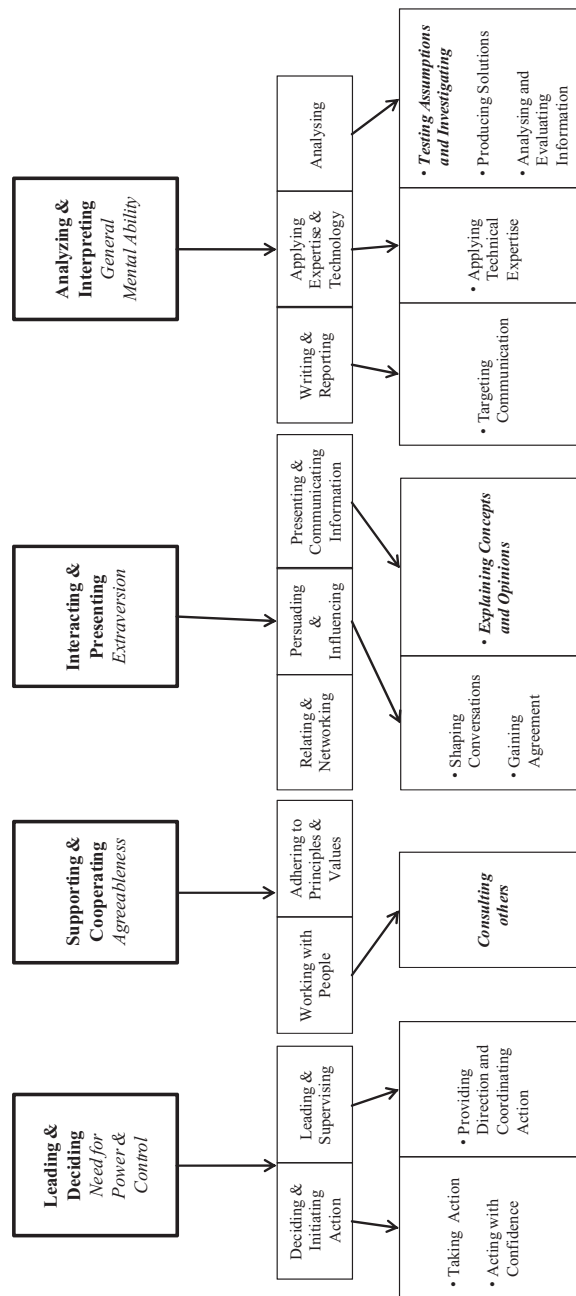


Figure 2. Critical competencies of requirements analysts, embedded in the SHL Universal Competency Framework (© 2004 by SHL Group). The top five competencies are shown in bold and italics. The underlying motivational, ability and personality factors most strongly associated with each of the competency factors at the top level are shown in italics (part 1).

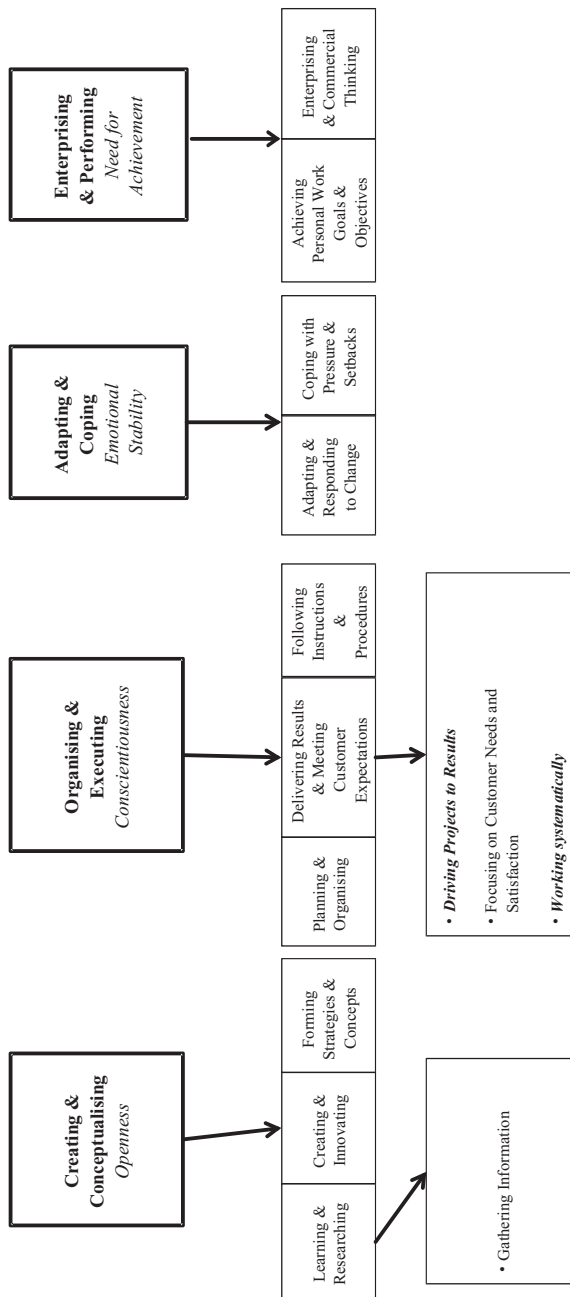


Figure 3. Critical competencies of requirements analysts, embedded in the SHL Universal Competency Framework (© 2004 by SHL Group). The top five competencies are shown in bold and italics. The underlying motivational, ability and personality factors most strongly associated with each of the competency factors at the top level are shown in italics (part 2).

Table 3. Titles and definitions of critical competencies for requirements analysts*

Title	Definition	Example	Effective behaviour (frequency)	Ineffective behaviour (frequency)	Overall frequency
Consulting others	Involves key stakeholders (mainly customers, but also developers and RAs) to a large extent early in the decision-making process.	<p>'We were in constant touch with the business. We repeatedly reviewed all the requirements with the business via interviews and email and checked whether we had understood correctly what they wanted. We, however, were also in early touch with the development [team] to check whether what the business wanted was viable'.</p> <p>'These analysts . . . when they went in to the business all they did was gather the reports of what had been done in the old system and it ended up that four weeks of time was lost because it had to be completely redone. Because they never took the time to learn about the new system and never really asked the business about their business problems, the work that they had spent to get in all this information could not be converted and interpreted into the new business model and then reengineered for them. So that was then lost time.' (negative example)</p>	13	9	22
Testing assumptions and investigating	Clarifies vague or inconsistent information, and questions statements and assumptions. Identifies and detects problems. RA is not intimidated by business language and complex formula, and does not hesitate to check with the customers about business aspects the RA is not familiar with.	<p>'Although the business had reassured us that no current processes would be impacted by the current project, we investigated the current process flows none the less and it was found that there was impact. This led to a re-scoping of the project to include additional processes'.</p> <p>'I was not really questioning whether what my requirements providers had asked us to do really made sense. In the end if we had been a little more critical about what they had given to us we could have spotted this a bit earlier and figured out that some things didn't really make sense.' (negative example)</p>	13	9	22

Explaining concepts and opinions	Clearly explains the logic of a concept in a simple way. Summarises key points of the SRS. Explains why the recommended concept is the best solution and how it is going to look like.	'So we had to be very clear, this is the definition of the requirements and this is what we are proposing as the design. Once they agreed, we ran through the demos, mocked up screens that showed how it was going to look, we talked through the solution, which is your functional specification document, how are we going to address their requirements'.	13	5	18
Working systematically	Integrates systematic work processes into RE activities (e.g. in review or working sessions; SRS). Emphasises a structured, methodical and consistent work style.	'So he could basically translate in order for me to service my customers. He was able to articulate real world real life customer language . . . , and he was able to translate that to programmers'. 'Then you go to talk to the person who conceived this, which is the functionality, and spend a couple of days with them to sort of understand why they did this, what are they looking at, and then go to the senior management . . . So that's the first set of the discussions that happened. And then these people go and say it to the operating people . . . Once you get that input then you go to the first pilot countries plus some of the business people . . . So that is the 4th level of the interviews that happened with them.'	9	4	13
		' . . . determined process, who reacts to whom and how [between customer and RA]. Generally written, but also oral if necessary. And then, finally, to document the content in a written form, that was also important'.			

Table 3. cont.

Title	Definition	Example	Effective behaviour (frequency)	Ineffective behaviour (frequency)	Overall frequency
Driving projects to results	Works in a goal-oriented way. Keeps in mind that creativity is desirable but not without any limits, and focuses on the minimum necessary to reach project goals. Keeps the scope in mind when additional wishes or changes are expressed.	'From my perspective, he should have been driving this project . . . later on, meetings had been going on for a number of months and nothing was really coming out of it, so the analyst was at the meetings and sort of listening to what they [business people] were saying, maybe recording them, requirements were being discussed, but the analyst should have been taking more control of the situation' (negative example)	4	6	10
Gathering information	Gathers relevant and complete information and documents them. Uses a range of sources (e.g. intranet, documentation, observation, interviews, prototypes).	'What we found is that the IT group that we began to work with a few years ago knew nothing about our organization. Therefore, they had to start from understanding what we do and how they could help us. Some of the things that I've noticed that really helped were that they came to the table with lots of questions about our business, so they were prepared initially with questions about what we did and then as they drilled into those questions they were very efficient and effective at taking notes and using those notes throughout the last 3 years to pull back upon when they were taking a new piece of development for us. . . . Their understanding of what it is that our team does allowed them to notice whenever there was something that what we didn't really tell them. This was necessary because we often take for granted some of the things that we do on the day-to-day basis. So it was the detailed questioning, the documentation and actually being on site with us that provided a good foundation for them to build our requirements.'	7	1	8

Focusing on customer needs and satisfaction	Is responsive to the wishes of the customers and understands their (unexpressed) needs. Takes into consideration the daily activities of the users.	'Because the project was on the west coast and we were on the east coast, there was a lot of expense involved in getting the end users and the business analysts together, so we developed the system that we thought would meet the needs When we were there to do the testing [we found that] it [our solution] wasn't going to work for them, for their day-to-day activity. A consequence was that we had to re-write the system, and [therefore] additional expense to the company.' (negative example)	4	2	6
Applying technical expertise	Shows detailed job knowledge and expertise, and uses specialist skills in own work (e.g., knows how to ask questions and draw use cases).	'The analyst was extremely successful in soliciting the requirements from the client, really clarifying their needs, and applying subject matter expertise to the work sessions to explore all the possible outcomes, the exceptions, and what they really meant by this type of situation. So in many situations, the client was unable to articulate the requirements because they simply didn't know. So the business analyst would ask questions and say "okay, do the systems that you use need to comply with standards for . . .". In a relatively short order and [with] minimal effort the use cases were realized and the description of the behavior of the system was reflected back to the client "Okay, this is what we heard. Why did the system do this, this and this." Essentially doing use case realizations by walking through sequence diagrams and such like'.	3	3	6
Producing solutions	Produces a workable solution independently, taking into account relevant needs and constraints.	'It was great that they came to the meeting with alternative solutions and variants. The RAs walked the customer through the solutions and then the customer chose one of them. And he [the RA] is capable of this, because he has the business knowledge'.	4	2	6

Table 3. cont.

Title	Definition	Example	Effective behaviour (frequency)	Ineffective behaviour (frequency)	Overall frequency
Acting with confidence	States views in a confident and assertive way.	'The business analysts were junior and didn't know how to ask additional questions, because they felt that the business was the authority and they didn't have the confidence to ask additional questions'. (negative example)	2	3	5
Shaping conversations	Actively guides the course of conversations.	'The analyst guided the conversations at the workshop very well, he could communicate well with everybody even though there were differing viewpoints on tools, processes etc . . . he was well prepared and knew when to interfere in conversations among the business people'.	4	1	5
Taking action	Takes measures without hesitating too long. Takes preliminary steps to do what needs to be done without direction.	'He was always quick to escalate and report whenever he assessed that the quality of the work packages was not good enough'.	3	2	5
Targeting communication	Writes SRS in a structured way and avoids technical jargon. Communicates with each stakeholder group at an appropriate level.	'The problem was that the specifications were very unstructured, the locality was missing. For example, the specification for the dialog [screen dialog box] was totally distributed in the SRS'. (negative example) 'The requirements analysts generated some UML [Unified Modeling Language] diagrams, sent them to the business and asked: "Is this what you ordered?" Then the business wrote back: "If this represents what we said then yes". That means they talked past each other. UML is important for the people who know how to read it, but one has to speak the language of all the different stakeholders'. (negative example)	3	2	5

Gaining agreement	Gains agreement among customers or between business and IT. Makes each party feel that their individual needs were reacted to and that the agreement is fair and not political.	'He made everybody feel that their individual situation was considered and that each party gets a big chunk of it'.	3	1	4
Analysing and evaluating information	Identifies key relationships within and between systems, relevant patterns and effects at both high and detailed level of analysis.	'So we have a business analyst who's trained, who has taken courses on business analytics, and she was able to draw the high level requirements. We received packages for the project and started to work down through them and to go through and identify new requirements that were not originally identified as being part of the project ... So she was very good at that, well trained, certainly determined about what looked like it belonged and what didn't ... she had a good acumen of business and analytical knowledge, they need to think systematically'.	3	0	3
Providing direction and coordinating action	Coordinates group activities and provides the business with clear direction about how to go through the requirements engineering process. Ensures roles and responsibilities within the requirements team are clear.	'If we keep on allowing the requirements to expand then we will never be able to deliver. So we had to be very clear, this is the definition of the requirements and this is what we are proposing as the design. ... and this is how we do requirements specifications'.	1	2	3

*The competency titles and definitions are based on the SHL Universal Competency Framework™ Profiler and Designer cards (© 2004 by SHL Group), where each competency is described in detail, and have been extended and adapted to the context of RE.
IT, information technology; RA, requirements analyst; RE, requirements engineering; SRS, software requirements specifications.

specific combinations of competencies in a given situation – the most frequent combination patterns were (a) ‘Consulting others’ and ‘Testing assumptions and investigating’; (b) ‘Focusing on customer needs and satisfaction’ and ‘Testing assumptions and investigating’; and (c) ‘Consulting others’ and ‘Gathering information’. In the interviews, all of these four competencies focused on the interaction between the analyst and the customers. Although ‘Consulting others’ also referred to early opinion seeking with regard to the developers, it mainly concentrated on the active integration of customers into all aspects of the decision-making process. Regular communication with the customers, early presentation of concepts and feedback-seeking differentiated between effective and ineffective behaviour of the RA. However, it was not always sufficient to solely concentrate on customer needs and satisfaction. In some cases, this even led to unsuccessful project outcomes – when users, for example, wanted to keep doing their work exactly the same way despite possible improvements or when they underestimated the technical complexity of the desired software product. Another (‘classic’) example included constant or late change requests by the customers that led to cost and time overruns (‘scope creep’). Only when analysts demonstrated the ability to both focus on customer needs and question assumptions made by the customers, the results of their behaviour were described as effective.

Research question 2 referred to the types of typical and important work situations the RA had to deal with. Apart from three general work situations that corresponded to the broad tasks requirements elicitation, analysis and specification (requirements validation was not mentioned at all by the interviewees), we identified six types of particular work situations. The definitions of these work situations including the contextual factors in terms of antecedents that led to these challenging, particular work situations are shown in Table 4. These antecedents can also be considered specific risk factors.

Research question 3 referred to the results of the (in)effective handling of the work situations by the RA. Table 5 presents six dimensions of the variable ‘result’.

DISCUSSION

This paper is the first to elaborate a competency model for RAs, thus contributing to theory in a critical area of software development. Our study sheds light into a specific job profile of the IT profession that has not been systematically analysed yet. This needs specific attention as we know from research that not having a structured, competency model-based human resource process will lead to inappropriate job staffing.

Overall, our research results emphasise the dual importance of both social and analytical competencies. However, tradeoffs of analytical skills with interpersonal skills have often been moaned about in the personnel development literature. Nevertheless, precise balancing of these competencies seems crucial for the effectiveness of an RA.

An interesting research result is that advanced RE techniques and tools, although often described as important in the academic literature, did not seem to play a significant role from the interviewees’ perspective. Tools were not mentioned at all in the critical

Table 4. General and particular work situations for the requirements analyst

Work situation	Definition and context	Frequency
General work situation		
Requirements elicitation	Requirements elicitation concerns the discovery of the requirements from sources such as the project stakeholders, organisational documentation, or, as is often the case in a redevelopment project, from the existing system specifications.	26
Requirements analysis	In the analysis activity, the various stakeholders examine the elicited data and then discuss and agree upon the set of, often prioritised, requirements to be implemented in the system under development.	11
Requirements specification	Requirements specification concerns the chronicling of the requirements in the software requirements specification, which contains all the requirements for the system.	7
Particular work situation		
Product scope is too big	The product scope is too big in that not all of the requirements can be implemented within the given budget and time frame. This is often caused by early agreements that are not based on a detailed review of the forecast requirements, or by provision of the requirements in pieces and over long periods of time. Antecedents of this situation are frequently a lack of understanding of requirements engineering processes as well as distrust towards the information technology (IT) department on the part of the customer.	4
Requirements have differing qualities	The requirements specifications have differing qualities because of various authors (requirements providers) who either worked together on a bigger project or worked separately on several systems, which, however, did not function independently of each other. This is usually caused by lack of planning and communication about the definition of the boundaries and the format of the requirements specification. They then become difficult to integrate, and interrelationships or gaps are hard to identify.	3
Breakdown in communication	A breakdown in communication takes place between project members in that project members are not willing to communicate with each other or expect others to reach out to them. This is often triggered by the overall difficult relationship between the business units and IT department, unclear project roles or lack of training as analysts. Moreover, the different terminology and acronyms used in the two domains can lead to a communications breakdown.	2
Customers disagree among themselves	Customers from various departments disagree among themselves regarding the systems requirements. They represent the departments' particular interests, which are sometimes incompatible with each other. This can be exacerbated by the differing cultures of the departments.	2
Requirements are old world-oriented	The users wanted to keep doing their work exactly the same way, and therefore refused to accept the requirements specification for the new system that would make the usage of the system more efficient and easier to learn for new employees. This can be exacerbated when very few users are experts for an important business domain and are indispensable for the organisation as their knowledge has not been documented.	2
Requirements are solution-oriented	The customers already described 'how' instead of 'what' when specifying the requirements. That is, a specific solution is already defined, but the explanation of the business needs and context is missing. This is more likely when customers possess IT knowledge but lack an understanding of requirements engineering processes.	2

Table 5. Result dimensions

Dimension	Definition	Example	Frequency
Project progress	A good foundation for the further advancement of the project was laid (e.g. the largest part of the requirements was prioritised and specified precisely and correctly; consensus was reached). Further inquiries and rework were reduced. Collaboration between project members (e.g. customers, requirements analysts, developers) functioned smoothly and efficiently.	'... because we even got the thumbs up from the technical manager that what we had done in terms of the analysis and requirements work was more than sufficient for them to move ahead'. '... Having a basis on which we could work. First of all, it was hard work to pull off the review, to get the "okay" from everybody, this is the basis on which the technical design is built upon'.	26
Timeliness	The system was delivered on time.	'So it was a faster delivery to the end user ... That allowed them to get to the market quickly'.	19
Product fit	The optimum technical solution was implemented and not impeded by political factors. The product's functionality met the needs of the customers/users and the requirements.	'And the outcome of the behavior was that the scope was increased but the system met the bank's requirements far better than it was originally scoped'. 'It wasn't cancelled, but it did have a big impact especially with regard to how fast someone who doesn't know the system and who joined the bank recently can work their way into it. And that is, of course, easier with a relatively simple accounting logic than with accounting transactions that nobody needs, where someone has to execute 30 instead of 10 transactions'. (negative example)	19
Cost	The system was delivered on budget. There were no indirect costs in terms of opportunity costs.	'... there were multiple projects that had to be to be either abandoned or deferred so that resources could be transferred to this project ... deferring of other projects which delayed other customers being added or other products being launched'. 'We would have saved money in the implementation phase ... when there is a more complex logic, it has an impact on all other phases, and then it is more costly and labor intensive than something that everybody understands and is relatively simple'. '... we went into production and then some clients received very strange orders ... so this costs a lot of money'. (negative examples)	13
Customer satisfaction	Satisfaction with and trust in the information technology (IT) department was enhanced. Customers felt that their needs were recognised and taken seriously. Customers became more confident in the IT's expertise and reliability.	'It increases the businesses faith in IT and what a great partnership we can provide as opposed to just throwing requirements over the wall ... the people on the business side have been consulting with this analyst a lot more on other initiatives coming up and recognising her as a true partner and not just someone that's an order taker'.	11
Maintenance and support	The documentation of requirements was not missing or incomplete, which increased the comprehensibility for technical people working in the maintenance and support area. Bugs had not to be fixed after completion of the project.	'The current situation is that we sometimes have to perform error corrections, because the users often find themselves in a situation where they cannot go back. Because the validation doesn't allow for it, that one undoes a change. So this is the problem, because it always happens unexpectedly. We folks from the maintenance and support group then have to create a script'. (negative example)	3

incident interviews, and advanced techniques were only mentioned twice. Related important topics such as requirement's traceability were not mentioned either. A possible explanation could be that problems in these areas are considered ordinary as, for instance, most RAs have never worked with or been trained on any RE tools, and therefore had no knowledge of how they could assist them in their work. Barriers seem to be that RE tools and advanced techniques are either not widely known by the participants or they tend to avoid the effort of introducing new tools and techniques. Additionally, companies hesitate to invest in RE tools as the cost-benefit ratio is often unclear. Our findings are in contrast to the study conducted by Sonnentag (2000) on excellent software engineers. The criticality of having high technical and computational knowledge does not seem to apply to the specific area of RE.

In the interviews with software developers, we discovered that the formal introduction of the organisational role RA can facilitate the success of projects considerably. Through the clear task divisions that this introduces, the developers can focus on their primary job – the development of the software itself, while the analyst takes over all communication-related tasks. However, to achieve this result, the RA must have the appropriate competencies in order to be effective. If not, they can be very obstructive. In two cases, the developers bypassed the analyst and directly contacted the customer for further inquiries.

Our study sheds light on a specific job profile of the IT profession that had yet to be systematically analysed. The presented competency model can be helpful to both individuals and organisations (companies, universities) in developing RE competencies. Competencies assist individuals by (a) summarising the experience and insight of seasoned practitioners; (b) providing a tool that individuals can use for their self-development; and (c) outlining a framework that can be utilised to help select, develop and understand the effectiveness of RAs. Based on our work, it might be possible to connect the identified competency profile with existing standard tests that are already successfully used in human resource management. For example, the personality factors 'Agreeableness', 'Extraversion' and 'Conscientiousness', which are associated with the top five competencies, can be measured by the 'NEO-Five Factor Inventory', containing 60 questionnaire items (McCrae & Costa, 2004). This would enable a better, cost-efficient and easy-to-apply recruitment and selection process. It also facilitates better individualised training programs for existing RAs. Moreover, this paper shows how other IT-related job profiles can be analysed, and correspondingly be explored and developed.

The main limitation of our research refers to the derivation of the model from interviews. However, the participants explained in detail how a specific behaviour of the RA had impacted the work situation and the further advancement of the project. Statements were often repeated independently from each other in various companies and countries, and corresponded to each other to a surprising degree. Also, the multidimensionality of the project results has to be considered. For example, the dimension 'customer' was, of course, especially emphasised by the customers and users. In contrast, project managers sometimes seem to prioritise the compliance with time and budget over the bug-free quality of the requirements specifications.

This can be explained by the common practice that these managers are focused on project delivery, and are not responsible for maintenance and support after completion of the development project.

Another limitation is that we exclusively interviewed practitioners from the financial services industry, which reduces the generalisability of our work. Finally, we analysed only the current work situations. Future research should consider improvements currently being implemented or being planned as well as future trends.

REFERENCES

- Abran, A. & Moore, J.W. (2004) *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, CA, USA.
- Al-Ani, B. & Sim, S. (2006a) So, you think you are a requirements engineer? *International Requirements Engineering Conference*, pp. 330–331.
- Al-Ani, B. & Sim, S. (2006b) Using expertise as a framework for evaluating requirements technology. *International Workshop on Comparative Evaluation in Requirements Engineering*, pp. 3–7.
- Alenljung, B. & Persson, A. (2008) Portraying the practice of decision-making in requirements engineering: a case of large scale bespoke development. *Requirements Engineering*, **13**, 257–279.
- Alexander, I. (1998) Engineering as a co-operative inquiry: a framework. *Requirements Engineering*, **3**, 130–137.
- Andrew, C., Buchan, J. & Petrova, K. (2009) Bridging the research-practice gap in requirements engineering through effective teaching and peer learning. *Sixth International Conference on Information Technology: New Generations*, pp. 678–683.
- Aurum, A. & Wohlin, C. (2005) Requirements engineering: setting the context. In: *Engineering and Managing Software Requirements*, 1st edn, Aurum, A. & Wohlin, C. (eds), p. 478. Springer, Berlin, Germany.
- Barnes, R., Gause, D. & Way, E. (2008) Teaching the unknown and the unknowable in requirements engineering education. *Requirements Engineering Education and Training*, 3rd International Workshop on Requirements Engineering Education and Training, 30–37.
- Bartram, D. (2005) The great eight competencies: a criterion-centric approach to validation. *Journal of Applied Psychology*, **90**, 1185–1203.
- Bartram, D., Robertson, I.T. & Callinan, M. (2002) Introduction. A framework for examining organizational effectiveness. In: *Organizational Effectiveness. The Role of Psychology*, Robertson, I.T., Callinan, M. & Bartram, D. (eds), pp. 1–10. John Wiley & Sons, Baffins Lane, Chichester, UK.
- Bassellier, G., Reich, B.H. & Benbasat, I. (2001) Information technology competence of business managers: a definition and research model. *Journal of Management Information Systems*, **17**, 159–182.
- Batra, D. & Davis, J.G. (1992) Conceptual data modelling in database design. Similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies*, **37**, 83–101.
- Beatty, J. & Alexander, M. (2008) Games-based requirements engineering training: an initial experience report. *International Requirements Engineering Conference*, pp. 211–216.
- Becker, S.A., Carmel, E. & Hevner, A.R. (1993) Integrating joint application development (JAD) into cleanroom development with ICASE. *System Sciences, 1993, Proceedings of the Twenty-Sixth Hawaii International Conference on*, **13**, pp. 13–21.
- Boehm, B. & Basili, V.R. (2001) Top 10 list [software development]. *Computer*, **34**, 135–137.
- Boyatzis, R.E. (1982) *The Competent Manager: A Model for Effective Performance*. John Wiley, New York, USA.
- Browne, G.J. & Ramesh, V. (2002) Improving information requirements determination: a cognitive perspective. *Information and Management*, **39**, 625–645.
- Byrd, T.A., Cossick, K.L. & Zmud, R.W. (1992) A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Quarterly*, **16**, 117–138.
- Cammock, P., Nilakant, V. & Dakin, S. (1995) Developing a lay model of managerial effectiveness: a social constructionist perspective. *Journal of Management Studies*, **32**, 443–474.

- Curtis, B., Krasner, H. & Iscoe, N. (1988) A field study of the software design process for large systems. *Communications of the ACM*, **31**, 1268–1287.
- Davis, A.M. & Hickey, A.M. (2002) Requirements researchers: do we practice what we preach? *Requirements Engineering*, **7**, 107–111.
- Davis, G.B. (1982) Strategies for information requirements determination. *IBM Systems Journal*, **21**, 4–30.
- Ericsson, K.A. & Smith, J. (1991) Prospects and limits of the empirical study of expertise: an introduction. In: *Toward a General Theory of Expertise: Prospects and Limits*, Smith, K.A.E.J. (ed.), pp. 1–38. Cambridge University Press, Cambridge, UK.
- Finkelstein, A. & Dowell, J. (1996) A comedy of errors: the London ambulance service case study software specification and design. *The 8th International Workshop on Schloss Velen*, pp. 2–4.
- Flanagan, J.C. (1954) The critical incident technique. *Psychological Bulletin*, **51**, 327–358.
- Glaser, B.G. (1992) *Basics of Grounded Theory Analysis. Emergence vs Forcing*. Sociology Press, Mill Valley, Calif.
- Hall, A. (1997) What's the use of requirements engineering? *Third IEEE International Symposium on Requirements Engineering*, pp. 2–3.
- Hall, T., Beecham, S. & Rainer, A. (2002) Requirements problems in twelve software companies: an empirical analysis. *IEE Proceedings Software*, **149**, 153–160.
- Hevner, A.R. & Mills, H.D. (1995) Box-structured requirements determination methods. *Decision Support Systems*, **13**, 223–239.
- Hofmann, H.F. & Lehner, F. (2001) Requirements engineering as a success factor in software projects. *Software, IEEE*, **18**, 58–66.
- Hollenbeck, G.P., McCall, M.W., Jr & Silzer, R.F. (2006) Leadership competency models. *The Leadership Quarterly*, **17**, 398–413.
- Hull, E., Jackson, K. & Dick, J. (2004) *Requirements Engineering*. Springer, Berlin, Germany.
- Hunter, J.E. & Hunter, R.F. (1984) Validity and utility of alternative predictors of job performance. *Psychological Bulletin*, **96**, 72–98.
- Hurtz, G.M. & Donovan, J.J. (2000) Personality and job performance: The Big Five revisited. *Journal of Applied Psychology*, **85**, 869–879.
- Jeffries, R., Turner, A.A., Polson, P.G. & Atwood, M.E. (1982) The processes involved in designing software. In: *Cognitive Skills and Their Acquisition*, Anderson, J.R. (ed.), pp. 255–283. Erlbaum, Hillsdale, New Jersey.
- Kelly, R. & Caplan, J. (1993) How Bell Labs creates star performance. *Harvard Business Review*, **71**, 128–139.
- King, S. (1996) Case tools and organizational action. *Information System Journal*, **6**, 173–194.
- Koch, A., Strobel, A., Kici, G. et al. (2009) Quality of the critical incident technique in practice: interrater reliability and users' acceptance under real conditions. *Psychology Science*, **51**, 3–15.
- Kovitz, B. (2003) Hidden skills that support phased and agile requirements engineering. *Requirements Engineering*, **8**, 135–141.
- Kurz, R. & Bartram, D. (2002) Competency and individual performance: modelling the world of work. In: *Organizational Effectiveness. The Role of Psychology*, Robertson, I.T., Callinan, M. & Bartram, D. (eds), pp. 227–255. John Wiley & Sons, Baffins Lane, Chichester, UK.
- Lamsweerde, A. (2009) *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley & Sons, Chichester, England.
- Lucia, A.D. & Lepsinger, R. (1999) *The Art and Science of Competency Models*. Jossey-Bass/Pfeiffer, San Francisco, USA.
- Maiden, N. (2009) Oi, analyst – you're barred! *IEEE Software*, **26**, 13–14.
- Markus, L.H., Cooper-Thomas, H.D. & Allpress, K.N. (2005) Confounded by competencies? An evaluation of the evolution and use of competency models. *New Zealand Journal of Psychology*, **34**, 117–126.
- McClelland, D.C. (1973) Testing for competence rather than for 'intelligence'. *American Psychologist*, **28**, 1–14.
- McCrae, R.R. & Costa, P.T. (2004) A contemplated revision of the NEO Five-Factor Inventory. *Personality and Individual Differences*, **36**, 587–596.
- Mirabile, R.J. (1997) Everything you wanted to know about competency modeling. *Training and Development*, **51**, 73–77.
- Napier, N.P., Keil, M. & Tan, F.B. (2009) IT project managers' construction of successful project management practice: a repertory grid investigation. *Information Systems Journal*, **19**, 255–282.
- Niazi, M. & Shastry, S. (2003) Role of requirements engineering in software development process: an empirical study. *Multi Topic Conference, 2003. INMIC 2003. 7th International*, pp. 402–407.
- Nuseibeh, B.A. & Easterbrook, S.M. (2000) Requirements engineering: a roadmap. In: *22nd International Conference on Software Engineering, Icsse'00, The Future of Software Engineering ed.*, Finkelstein, A.C.W. (ed.), pp. 35–46. IEEE Computer Society Press, Limerick, Ireland.

- Paech, B. (2008) What is a requirements engineer? *IEEE Software*, **25**, 16–17.
- Paech, B., König, T., Borner, L. et al. (2005) An analysis of empirical requirements engineering survey data. In: *Engineering and Managing Software Requirements*, Aurum, A. & Wohlin, C.W. (eds), pp. 427–452. Springer, Berlin, Germany.
- Penzenstadler, B., Frenzel, G., Haller, G. et al. (2009) Soft Skills REquired: a practical approach for empowering soft skills in the engineering world. *RE CIRCUS Workshop at the International Conference on Requirements Engineering*, pp. 31–36.
- Regev, G., Gause, D. C. & Wegmann, A. (2008) Requirements engineering education in the 21st century, an experiential learning approach. *International Requirements Engineering Conference*, pp. 85–94.
- Russell, C.J. (2001) A longitudinal study of top-level executive performance. *Journal of Applied Psychology*, **86**, 560–573.
- Shippmann, J.S., Ash, R.A., Battista, M. et al. (2000) The practice of competency modeling. *Personnel Psychology*, **53**, 703–740.
- Silverman, D. (1998) Qualitative research: meanings or practices? *Information Systems Journal*, **8**, 3–20.
- Skulmoski, G.J. & Hartman, F.T. (2010) Information systems project manager soft competencies: a project-phase investigation. *Project Management Journal*, **41**, 61–80.
- Sommerville, I. & Kotonya, G. (1998) *Requirements Engineering: Processes and Techniques*. Wiley & Sons, Chichester, England.
- Sonnentag, S. (1995) Excellent software professionals: experience, work activities, and perception by peers. *Behaviour and Information Technology*, **14**, 289–299.
- Sonnentag, S. (1998) Expertise in professional software design: a process study. *Journal of Applied Psychology*, **83**, 703–715.
- Sonnentag, S. (2000) Excellent performance: the role of communication and cooperation processes. *Applied Psychology: An International Review*, **49**, 483–497.
- Sonnentag, S., Niessen, C. & Volmer, J. (2006) Expertise in software design. In: *Cambridge Handbook of Expertise and Expert Performance*, Ericsson, K.A., Charness, N., Feltovich, P. et al. (eds), pp. 373–387. Cambridge University Press, Cambridge, UK.
- Stoof, A., Martens, R.L., van Merriënboer, J.J.G. et al. (2002) The boundary approach of competence: a constructivist aid for understanding and using the concept of competence. *Human Resource Development Review*, **1**, 345–365.
- Tett, R.P. & Burnett, D.D. (2003) A personality trait-based interactionist model of job performance. *Journal of Applied Psychology*, **88**, 500–517.
- Viswesvaran, C. & Ones, D.S. (2000) Perspectives on models of job performance. *International Journal of Selection and Assessment*, **8**, 216–226.
- Webster, J. & Watson, R.T. (2002) Analyzing the past to prepare for the future: writing a literature review. *MIS Quarterly*, **26**, xiii–xxiii.
- WI-lists (2008) WI-Orientierungslisten. *WIRTSCHAFTSINFORMATIK*.
- Young, R.R. (2003) *The Requirements Engineering Handbook*. Artech House Inc., Norwood, USA.

Biographies

Ruth Klendauer is a Postdoctoral Researcher at the Department of Sociology, Technische Universität München, Germany. She received her PhD in social and organizational psychology from the Ludwig-Maximilians-Universität München and was a Postdoctoral Fellow at Harvard University from 2006–2008. Her research interests include strategic human resources selection and development, requirements engineering, organizational development, innovation processes, mergers & acquisitions, and change management.

Marina Berkovich is a Researcher at the Chair for Information Systems, Technische Universität München since she graduated from there in Computer Science in 2007. Her research interests include requirements engineering, software engineering, and complex solutions. She is engaged in the research project ‘Sonderforschungsbereich (SFB) 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products’, funded by German Research Foundation (Deutsche Forschungsgemeinschaft – DFG).

Richard Gelvin is an independent software consultant and received his MSc in Software Technology from Napier University, Edinburgh in 1994. He has since then gained extensive experience in major financial software development projects in London, Glasgow, Edinburgh, Vienna, Luxembourg, Frankfurt and Hamburg.

Jan Marco Leimeister is a Full Professor of Information Systems and holds the Chair for Information Systems at Kassel University, Germany. He is director of the IS research center ITeG at Kassel University and runs research groups on service, collaboration and IT Innovation engineering and management manages

several publicly funded research projects. His teaching and research areas include IT innovation management, service science, ubiquitous and mobile computing, collaboration engineering, and strategic IT management.

Helmut Krcmar is a Full Professor of Information Systems and holds the Chair for Information Systems at the Department of Informatics, Technische Universität München, Germany, since 2002. He worked as a Postdoctoral Fellow at the IBM Los Angeles Scientific Center, as

Assistant Professor of Information Systems at the Leonard Stern School of Business, New York University, and at Baruch College, City University of New York. From 1987 to 2002 he was Chair for Information Systems, Hohenheim University, Stuttgart. His research interests include information and knowledge management, IT-enabled value webs, service management, computer-supported cooperative work, and information systems in health care and e-government.