Electronic version of an article published as **Requirements Engineering Journal, Vol. 12, Issue 3, 2007, pp. 145-162**
[doi: 10.1007/s00766-007-0042-4]

Requirements Eng (2007) 12:145–162
DOI 10.1007/s00766-007-0042-4

ORIGINAL RESEARCH

A field study of the requirements engineering practice in Australian software industry

Emila Sadraei · Aybüke Aurum · Ghassan Beydoun · Barbara Paech

Received: 18 December 2005 / Accepted: 18 January 2007 / Published online: 15 February 2007

Abstract Empirical studies have demonstrated that requirements errors introduced during software development are most numerous in the software lifecycle, making software requirements critical determinants of software quality. This article reports an exploratory study which provides insight into industrial practices with respect to requirements engineering (RE). A combination of both qualitative and quantitative data is collected, using semi-structured interviews and a detailed questionnaire from 28 software projects in 16 Australian companies. The contribution of this RE study is threefold: Firstly, it includes a detailed examination of the characteristics of the RE activities involved in the projects. Secondly, it reconstructs the underlying practiced process models. Thirdly, it compares these models to one another and with a number of well-known process models from RE literature to give insight into the gap between RE theory and practice.

E. Sadraei · A. Aurum (✉)
School of Information Systems,
Technology and Management,
University of New South Wales,
Sydney, NSW 2052, Australia
e-mail: aybuke@unsw.edu.au

G. Beydoun
School of Information Systems,
Faculty of Information Technology,
University of Wollongong,
Sydney, NSW 2522, Australia
e-mail: beydoun@uow.edu.au

B. Paech
Institute for Computer Science,
University of Heidelberg,
Im Neuenheimer Feld 326,
69120 Heidelberg, Germany
e-mail: paech@informatik.uni-heidelberg.de

1 Introduction

Since the 1970s, problems with requirements engineering (RE) have persisted as a key factor in the inefficiency and failure of software projects [48, 52]. Improvements in the RE process have the potential to reduce development costs and development time, and to increase software quality [12, 16]. Unfortunately, the transfer of improvements identified in existing research literature into industry has been largely unsuccessful [3, 31]. Researchers have continued to develop new techniques and methods, but practitioners have experienced difficulty in applying them [23, 41]. A recent study of nine companies [48] indicated that the companies operate at the initial level of RE process maturity [44, 46], further emphasizing the infancy of RE practice uptake from research into industry [48]. Earlier studies made similar observations, indicating little improvement in uptake trends over the past decade [10, 31, 34, 41]. The challenge is thus to fill the gap between the research conducted by academicians and its implementation undertaken by practitioners. Developing a mutual understanding and improving communication between the two groups is essential and motivates challenging research on current RE industry practices.

The first rule of RE, ‘‘know thy customer’’, obligates RE researchers to know how RE is actually practiced

[14]. However, researchers do not always have ready access to current industry practices allowing a close analysis and the identification of flaws in RE. Such access is critical to enable researchers to suggest improvements and to deliver applicable solutions [17]. These observations highlight the importance of overcoming the communication barriers so that empirical research can be conducted with practitioners. In other words, in order to understand the state of practice, more empirical evidence about RE practice is needed [38].

Due to the uncertain and multidisciplinary nature of the RE process, a process model is a prerequisite for applying any structure or discipline to RE [51], managing its complexity. Our research is motivated by the realization that it is necessary for researchers to be familiar with the state of industry practice and to understand existing processes, since knowledge of actual RE practices reduces the risk of faulty modeling or impractical process engineering. It is also motivated by the view that software development is a knowledge-intensive process and knowledge management is an important mechanism for understanding and improving software development projects [4, 52].

This article presents the results of an empirical study, based on data collected from 28 software projects in 16 different Australian companies, in an attempt to assist practitioners and researchers to share their perspectives. Our main objective is to explore and give insight into the current RE practices. The contribution of this article is threefold: (1) it examines RE activities in projects in terms of project characteristics along with process, activity, role, and document awareness; (2) it measures the amount of effort used in each RE activity, determining if there are recurring patterns in performing these activities and identifying common activities across all the observed industry-based RE processes; (3) it reconstructs underlying practiced process models in the companies studied and compares these models to each others and with a number of well-known process models identified in the RE literature.

This research is of interest as accuracy and quality of software depend on the RE process and the activities that generate and analyze requirements [2]. By exposing how requirements are understood, this study can be used to analyze the root of many RE problems that stem from poor communication between researchers and practitioners [12]. It can also be used to improve the estimation of effort required for each stage of RE and for individual RE activities. This is critical for controlling project costs and preventing cost overruns. Project management can also be improved by avoiding unbalanced efforts on RE activities, which can otherwise be destructive to software quality [36]. This leads to a successful RE process which reduces the cost of reworking at the construction as well as at the maintenance stages, as opposed to the more expensive option of fixing problems at the later stages of the software development process [6].

The rest of this paper is organized as follows: Section 2 presents background knowledge and related work. Section 3 discusses our research questions and methodology. Section 4 presents the data analysis and the results. Section 5 discusses the validity threats and limitations of this study. Section 6 concludes with a discussion of future extension of this study.

2 Background

An extensive comparison of previous studies that focus on empirical work in RE practices was presented in [38]. The preliminary studies for this research were published in [29, 39]. The present work examines relevant RE studies, namely RE process models, effort invested in RE activities, implicit and explicit RE activities, and awareness.

2.1 RE process models

The quality of a software product is largely controlled by the quality of its development process. (This statement does not take into account RE for agile methods, as this is not in the scope of this paper. The reader may refer to [3] for further reading on RE for agile methods). An RE process model outlines procedures and methods as guidelines, in terms of activities, artifacts, roles and techniques or tools. A well-defined and structured software development process is crucial to delivering a quality product on time and within budget. Thus, understanding and modeling current RE processes as part of the software development process is an important step towards improving RE practices and consequently increasing the success of software projects [28].

There is a large variety of RE models. Sometimes they are conflicting in nature, ranging from linear and incremental to cyclical and iterative in structure [28, 32]. A survey of practitioners indicates the popularity of the waterfall model in RE [32], but other studies indicate that RE process models used in practice differ from RE process models represented in literature. In one case study [33], the RE process is found to be opportunistic, rather than systematic and incremental as the literature suggests. It is often hard to establish a monolithic RE process model for RE activities, because they are heavily intertwined [21]. We recognize this difficulty and acknowledge that the RE process is opportunistic rather than systematic as discussed by [33]. The process models that we selected from the literature are those that are most advocated by researchers. They include all main activities of the RE process and are "application domain-free": the waterfall model [43], a general linear model [26] and a conceptual linear model that indicates iterations between activities [47]. Non-linear models which introduce the RE process as iterative and cyclical in nature [25], the spiral model [8], standard "V" model and W model [49] are also considered in this piece of research.

This research is limited to the study and analysis of successful software development projects, in order to determine the features of a successful RE process model. The measure of "success" will vary, depending on how it has been defined [5]. Here, we use the classical definition of project success, i.e. projects that are completed within budget, on schedule, and that meet business objectives [50]. It is most likely that successful RE process models need to be highly situation dependent, since RE processes themselves depend on the nature of the software project, the customer–supplier relationship [26], organizational culture [16], technical maturity and any disciplinary involvement [48].

2.2 Effort in RE activities

The RE process is the sum of all RE activities, which primarily consists of gathering, documenting and managing requirements. Effective effort estimations for RE activities are vital, if the project costs are not to overrun dramatically. Effort estimations for future projects can be more accurate, if previous effort, data and experiences are collected and used appropriately [36]. Alas, little is known about the amount of effort needed in the requirements phase of software projects. One reason for this is that pre-planning estimates are often based upon very limited information about the product requirements [22].

Few studies on the effort devoted to different RE activities are available. A summary of previous research is provided in Table 1. There is no complete agreement amongst researchers on how this effort should be distributed to each activity [10, 27, 39]. Boehm [6] states that a mere 6% of the total cost of software development is devoted to RE. Recently, Alexander and Stevens [1] recommended a similar amount, suggesting that about 5% of the project effort goes into requirements, not including specification. This might be about 25% of the project duration (or no more than 3 months, dependent upon the project size). They state that system specifications might also take 20–25% of the project duration. Hoffmann and Lehner [20] examined 15 project teams on a number of projects and find that they spend on an average 15.7% of the project effort on RE activities. They also report that successful teams allocate on an average 6.4% of the total project effort to elicitation, 6.2% to modelling, and 3.1% to validation and verification. For the total effort in the RE process, this equates to 41% for elicitation, 39% for modelling and 20% for verification and validation [20]. Chatzoglou and Macaulay [11] surveyed 107 projects and found that requirements capture and analysis takes over 15% of the total project time. MacDonell and Shepperd [27] stress the level of uncertainty and subjectivity associated with the effort estimation in software project activities. In a smaller study of sixteen software projects in one company [27], their findings show that there were no "standard proportions" for the effort for particular development activities. They find that there is so much variance in the effort in the project planning and requirements specification phases and in comparison with the overall project effort, that no predictive patterns can be drawn. Their findings also show that 40–50% of the software development effort is dedicated to rework: locating and correcting defects found during testing. Rework increases as the project progresses through its phases, taking 66% of the total effort by the final integration and test phase [6]. Thus, more investment upfront may well lead to less rework. This is a common finding in the majority of recent industrial surveys, and employees favor that their companies invest more in RE activities upfront [32].

2.3 Explicit and implicit RE activities

The role of knowledge management as a success factor in large software projects is widely accepted [4]. Central to this task is storing project data as well as experiences and problems or issues noted by engineers into experience repositories or factories [4]. The intention is to share common knowledge and convert those implicit practices into shared, understood and adopted explicit practices that are then absorbed, so as to make them second nature, or implicit, to all. The ability to learn from earlier experiences, to share both formal and informal knowledge with colleagues and to collaborate with each other more efficiently, enhances communication and project well-being. One of the objectives of our study is to capture both explicit and

Table 1 Previous studies on RE effort distribution

Previous studies	Effort goes to RE
Boehm [6] (experience)	RE effort is 6% of the total cost of software development
Alexander and Stevens [2] (experience)	5% of project effort (not including specification) i.e., 25% of project duration, 20–25% system specification of project duration
Hoffmann and Lehner [20] (15 projects)	RE effort is average 15.7% of project effort (time). For successful projects 6.4% elicitation of project effort (which is 39% for RE effort) 6.2% modeling (which 36% of RE effort) 3.1% validation and verification (which is 25% of RE effort)
Chatzoglou and Macaulay [11] (107 projects)	15% requirements capture and analysis of elapsed time
MacDonell and Shepperd [27] (16 projects)	Too much variance in effort (time) in project planning and requirements specification 0.0% (median), 11% max requirements specification (mostly incorporated in planning or design) 40–50% rework (of total effort)

implicit activities to try to understand the nature of the requirements processes in selected companies.

We describe *implicit activities* as the activities performed by team members in an ad hoc manner; they do not involve the sharing of knowledge about the task [39] and ultimately lead to implicit or tacit organizational knowledge. Any attempt to convert this implicit and tacit organizational knowledge about the activities, steps and procedures involved in creating software into shared and reusable knowledge, creates the opportunity to share best practices and prevent duplication of effort in the software development process. The absence of such shared knowledge leads to the performance of tasks in an ad hoc manner, i.e. the success of the resulting system depends directly on the particular abilities and capabilities of the software development team.

Performing activities explicitly during the project lifecycle plays an important role in the prevention of such conditions. We define *explicit activities* as those activities that follow the stated requirements process in a visible and accountable manner to the whole project team [39]. Explicit activities provide retrievable information through developing explicit knowledge that can be used as new components in the construction of organizational knowledge. Furthermore, they increase the potential problem-solving ability of a development team and also facilitate decision-making activities during the software development process, through providing the possibility of knowledge transfer and knowledge integration; the latter has a significant role in reducing software development challenges [52]. Thus, capturing explicit and implicit activities during the project life cycle is a way of better understanding the nature of the RE process on the one hand, and obtaining an implicit evaluation of the level of explicit knowledge within the company, on the other.

2.4 The concept of awareness

The concept of ''awareness'' is widely studied in literature for problem solving activities [15, 54]. Awareness means different things to different people and it is used in different situations [24]. According to Dourish and Bellotti [15], ''*Awareness is an understanding of the activities of others, which provides a context for your own activity*''. Awareness is the beginning of self-understanding and it reduces the effort needed to coordinate tasks and resources and to anticipate other team members' actions [15, 18].

As software system complexity continually increases, software developers increasingly rely on their ability to share knowledge and coordinate their efforts. While there are several types of awareness in the context of RE, we focus on four specific types of awareness in our research: *process, activity, role*, and *document awareness*.

Process awareness is a sense of which stages software team members' tasks fit into the software project, what the next step is, and what needs to be done to move the process along [54]. *Activity awareness* relates to software team members' RE activities that take place over a long period of time. It is important that the software team maintains a ''big picture'' view of software project goals and focus their efforts on the key tasks that will help them achieve these goals [45]. Although, one of the most crucial activities in RE is documenting software requirements, unfortunately there is little attention paid to documentation during RE activities. *Documentation awareness* is the knowledge that team members have about the necessary documentation for RE activities. *Role awareness* relates to identification and delineation of the responsibilities of each software team member in the project.

3 Research method

In an attempt to understand industrial RE processes, this paper investigates the nature of RE activities, especially process awareness and their impact on management related activities. This includes allocation of roles and responsibilities, communication between analysts as well as total explicit and implicit effort invested in a selection of common RE activities. The performance of these activities is analyzed to determine the existence of recurring patterns and to identify common activities across all 28 RE projects in the 16 surveyed companies.

We conduct our field study in two stages: firstly the data relating to individual project cases in the context of each company is collected and analyzed. Secondly, the combined data from the cases across all the companies is collated and analyzed. The data is analyzed at the project level, the company level and the industry level. The unit of analysis in this study is the RE process within each company. Each unit of analysis consists of subunits of individual RE projects in the company. Cases within customer-specific projects are identified to explore the existence of patterns when performing RE processes. To achieve this, we select several cases within each industry to find probable similar results. RE practices across a range of industries are investigated and compared.

The risk of bias in terms of data collection is mitigated by collecting data from different organizations. This ensures a wide range of information from multiple sources on which we base our subsequent analysis. The criteria for selecting participating organizations are based on our cooperative contacts within the industry. To maintain the integrity of the data, the same data collection method is used throughout the study. A direct interaction mechanism in the form of semi-structured interviews using a questionnaire is employed. A combination of qualitative and quantitative data is collected, with emphasis on qualitative analysis in light of the statistical limitations introduced by a relatively small sample size.

In the rest of this section, the following research steps are detailed: the collection of data (i.e. the structure of the survey and how it is used); the data domain (an overview of the projects and companies involved); and the data analysis model (i.e. the attributes that are sought for carrying out later analysis).

3.1 Research design and data collection

The data collection instrument is based on a questionnaire developed and tested by the German Computer Society (Gesellschaft für Informatik, http://www.gi-ev.de). We customize this questionnaire for our study, by enlisting the help of other academicians and practitioners. We use a template questionnaire to facilitate the comparison of results across industries by ensuring consistency with respect to the data gathered. The questionnaire contains both open and close-ended questions. We frame the close-ended questions in four styles: ordinal, categorical, numerical and multiple-choice questions. The open-ended questions give the participants a chance to express their ideas without being forced to select options that do not represent their opinions. This combination of questions reduces the length of the questionnaire and ensures that the participants answer comprehensively, while simultaneously covering a wide range of information. We divide the questionnaire into three sections: (1) the background details of the project, the company, the product and the market, (2) the RE process of the project in detail, including the tasks, products and roles identified in the requirement engineering and (3) the practices and methods used as RE techniques.

For each project, the questionnaire captures data about the participants, the companies and the project contexts. The participants' positions, the number of years they worked for the company and their overall professional work experiences are examined. Information about the size of the companies, the size of their IT departments, the types of products they produced and their markets are examined also. The questionnaire also captures information about the project size, the project priorities and the customer type (internal or external). Furthermore, the questionnaire probes whether the company's documented RE process is based on a standard RE process model, and if the practiced RE process (for each project) deviates from the documented process.

To study the relationship between practice and research, the participants are asked to identify the RE phases, assign the RE activities to each phase, and identify the roles and efforts in each activity. We investigate whether the identified RE activities are explicitly, implicitly or not at all performed. It is also examined how the RE process fits into the project lifecycle: is it at the beginning of the project, at the beginning of each increment or is it a continuous activity during the whole project life cycle. Based on the data collected from the participants, high-level process models are constructed for each project and then compared with models in the existing literature.

The participants are selected based on their responsibilities and involvement in the RE process of a software development project. Only those individuals who are highly involved in RE process are selected to participate. Most of our participants are in a project manager or business analyst role with business or technical backgrounds. In some cases, we interview multiple participants to provide varying perspectives. Each participant completes the written questionnaire and adds additional information verbally to the researcher privately. All interviews are taped to prevent inaccurate reporting of the verbal input of the participants. The private and confidential interview sessions allow the participants to clarify the meanings of the terms and questions used and ensure that the participants have a clear understanding of what is being examined. To maintain a quiet and undisrupted environment, interviews are held in private meeting rooms on the companies' premises and at convenient times for the participants during work hours. The duration of each interview is approximately 1 h. This time fluctuates slightly, depending on the amount of additional verbal information provided by the participant.

3.2 Project cases description

We examine 28 successful customer-specific software projects in 16 different companies from the following industries: finance, pharmaceutical, health services, telecommunications, education and food. Most of the projects are thus in industries where software was a supporting product, not a main business. The software types in these projects consisted primarily of business systems, along with a small number of embedded systems. The project characteristics are summarized in Table 2. In our analysis we compare the characteristics of where the products include embedded systems with business systems [we view business systems and embedded systems as being mutually exclusive in the questionnaire]. For the majority of the projects, their customers are internal and have direct interactions with the developers. Only five projects have external customers and their customer-supplier relationships are consistent with the "*responding to a specific customer request*" scenario (Macaulay 1996).

3.3 RE process model

We use the following eight activities to describe RE process models for benchmarking purposes in the study:

1. *Project creation* This is initiating a project and deciding whether a modification should take place in the existing system, or whether a new product should be developed.
2. *Elicitation* This is identifying the high level goals of the target system, the requirements for different groups of users, and the tasks to be accomplished, along with system boundaries [32].
3. *Analysis* This is analyzing the requirements in order to uncover any conflicts, ambiguities, missing or duplicate requirements in order to identify alternatives and convert them into a structured and unambiguous representation.
4. *Negotiation* This is selecting the best trade-offs that receive agreement from all parties including the requirements engineers and stakeholders [53].
5. *Verification and validation* This is examining the requirements to find any deficiencies in their consistency, accuracy and adequacy [48]. It also includes the investigation of their feasibility, verifying the cost of development and the consumption of resources.
6. *Change management* This is recognizing changes through continuous requirements elicitation, reevaluation of risk and evaluation of the system in its operational environment [35], to assure that all relevant information for each change is collected, and that a cost/benefit analysis is carried out [48].
7. *Requirement tracing* This is managing the evolution of requirements; recording and maintaining traces about the history of each requirement to track the origins, so that if a change has to be made to a design component, the original requirement can be located [13].
8. *Documentation* This is documenting the RE process and the actual requirements specification document.

4 Data analysis and results

The following section presents the results of this study and shows the relationships between their characteristics. The data reveals various characteristics of the performed RE activities, how they fit into the project life cycle and how the effort is distributed amongst them. It also reveals the degree of awareness about RE and any associated organizational roles.

4.1 Effort distribution

Our findings indicate that the nature of the project plays an important role in the distribution of the RE effort amongst activities. A trade-off between RE activities becomes obvious. When developers spend little time on a certain activity in a project; they make

Table 2 Project characteristics

Project names	Customer type	Industry type	Software type	Project description
Ph1	Internal	Manufacturing/pharmaceutical	Embedded system	Upgrading current dispensary management system
Ph2[a]	Internal	Manufacturing/pharmaceutical	Business system	Upgrading current barcode system for package processing
Ph3[a]	Internal	Manufacturing/pharmaceutical	Embedded system	Developing a control system for production line
Ph4[a]	Internal	Manufacturing/pharmaceutical	Business system	Developing a key part of an integrated manufacturing solution
Ph5[a]	Internal	Manufacturing/pharmaceutical	Embedded system	Developing a line management system in manufacturing division
Ph6[a]	Internal	Manufacturing/pharmaceutical	Business system	Implementation of an enterprise resource planning system
Tel1[a]	Internal	Telecommunications	Business system	Merge of two databases
Tel2[a]	Internal	Telecommunications	Business system	Developing a decision support system
Tel3[a]	Internal	Telecommunications	Business system	Customization an existing business system
Tel4[a]	Internal	Telecommunications	Business system	Customization an existing business system
Hs1[a]	Internal	Health service	Business system	Upgrading existing administrative system
Hs2[a]	Internal	Health service	Business system	Developing a decision Support System
Fd1	Internal	Manufacturing/food	Business system	Implementation of an enterprise resource planning system
Fd2	Internal	Manufacturing/food	Business system	Upgrading all systems particularly ERP to introduce GST
Fd3	Internal	Manufacturing/food	Business system	Implementing enhancements to an existing sales promotions management system
Fin1	Internal	Finance	Business system	Developing a customized website
Fin2	Internal	Finance	Business system	Implementation and customization of CRM system
Fin3	Internal	Finance	Business system	Implementation a new trading system
Fin4	Internal	Finance	Business system	Implementation a new legacy system
Fin5	Internal	Finance	Business system	Developing system for asset taking, controlling and monitoring process
Fin6	Internal	Finance	Business system	Developing a HR system
Fin7	Internal	Finance	Business system	Developing a web-based application
IT1	Internal	Education	Business system	Developing a web-based application
IT2	External	Outsourcing (for local customers)	Business system	Developing a vast portal (intranet)
IT3	External	Outsourcing (for local customers)	Business system	Developing a standard operating environment for customer service department
IT4	External	Outsourcing (for local customers)	Business system	Building a web-based purchasing system
IT5	External	Outsourcing (for local customers)	Business system	Automation of broadband activation process
IT6	External	Outsourcing (for local customers)	Business system	Implementation of a portable business system to support medical process

[a] Mission critical projects

up for this by spending more time on the following phase. Moreover, there is a great deal of variation between the team members' perceptions of the RE effort distribution (even amongst those working on the same project) within the same company, complementing MacDonell and Shepperd [27] findings. Figure 1 illustrates the average effort invested for each RE activity in internal as well as external projects.

The aggregate of average effort and resources spent on the first three RE activities for internal and external projects are 53.5 and 48%, respectively. There is no notable difference between the two but the distribution of effort across these activities differs markedly. The effort invested in the project creation of external projects is almost twice the effort invested in internal projects. Conversely, the effort devoted to the elicitation for external projects is a mere third of the effort devoted to internal projects. The effort spent on requirements analysis, on the other hand, is roughly equal for both internal and external projects. The average number of resources and amount of effort invested in the negotiation, validation and change management is also roughly the same across external and internal projects. In contrast, a drastic difference is observed between internal and external projects for requirements tracing, as illustrated in Fig. 1. That

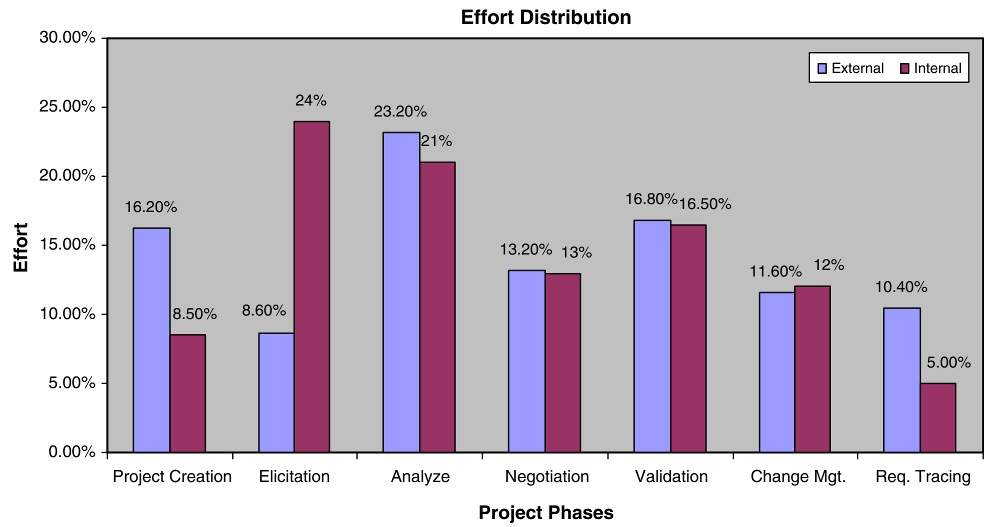


Fig. 1 Average efforts invested in RE activities among all projects

means, in both types of projects about half of the RE effort is invested in the final stages of the RE process. This is probably due to the lack of effort in earlier stages of the RE process, which is concerning as it leads to higher demand of financial resources.

We also compare our results with Hoffmann and Lehner's study [20] conducted within six COTS and nine customized application development projects in the telecommunication and banking industries. We group our RE activities in approximate accordance with Hoffmann and Lehner's RE activities. Table 3 depicts the results of this comparison. While the types of customers in Hoffmann and Lehner's study are unknown; it can be observed that there are significant similarities between the results of our study and that of Hoffmann and Lehner, reinforcing their findings. Notably, with reference to the internal projects, the pharmaceutical industry exhibits a balanced distribution of the effort spent on RE activities with 35% of the effort spent on elicitation, 34% on modeling and 31% on validation. However, for external projects our results depart from those of Hoffmann and Lehner. We do not observe that a balanced effort is spent on the RE process in successful projects (as they do).

Table 4 shows the highest and the lowest effort invested in each activity for both internal and external types of projects. Internal projects are shown in smaller groups of different industries.

Several observations are made when analyzing the results at an industry level: In pharmaceutical projects, most of the efforts are invested in the elicitation, validation and change management. The effort on change management in pharmaceutical projects is comparatively higher than in other industries. This is most likely due to the requirement of adhering to external industry regulations. Among projects in the food industry, the elicitation and analysis are the largest proportions of effort. In telecommunication projects, the highest effort lies in the elicitation and analysis. In finance, project creation and requirements tracing have the lowest amount of effort. Total effort invested in the other five RE activities, i.e. elicitation, analysis, negotiation, validation, change management and requirements tracing in different projects is about the same for all. Companies providing health services invest greater effort into the elicitation and the change management and our only education service provider also invests higher effort in the elicitation. However, due to the small number of companies providing health and education services in our study, these results are not conclusive. In embedded systems projects, most of the RE effort is invested in the elicitation and validation. The pattern of distribution of efforts for these smaller groups within internal projects is consistent with what is shown in Fig. 1 (for internal projects). Hence, for internal projects, regardless of industry or technique employed, the lowest effort is on the project creation and the requirements tracing, while the greatest effort is devoted to the elicitation (Table 4).

4.2 Distribution of RE activities

We studied the distribution of the RE effort to the different RE activities and asked the participants to state which activities were explicit and which were implicit or not performed at all. We collected data not only on the explicit effort but on the unrecorded effort exerted "implicitly" by project team members on their own accord.

Table 3 Comparison of RE effort to a previous study

		Our study Internal (%)	External (%)	Hoffmann and Lehner [20] (%)
Elicitation	Project creation + elicitation	32.5	24.8	39
Modelling	Analysis + change management + tracing	38	45.2	36
Validation	Verification and validation + negotiation	29.5	30	25

Table 4 Highest (√) and lowest (X) rate of efforts invested on RE activities

		Project creation	Elicitation	Analysis	Validation	Change management	Requirements tracing
Internal customers	Pharmaceutical industries		√		√	√	X
	Food industries		√	√			X
	Financial industries	X			√		X
	Telecommunication		√				X
External customers			X	√			

Figure 2 a, b summarizes our findings on explicit and implicit RE activities. In most cases, we observe a trade-off between the RE activities for both internal and external projects, i.e. if an activity is disregarded in one phase of project, it gets covered in the following phases. These findings are similar to our previous studies [38].

In all the cases involving internal customers, the number of explicit RE activities is considerably higher than that of implicit activities. While the effort for requirements elicitation and validation, 87 and 70%, respectively, is performed explicitly, the effort for requirements tracing is only 35%. As for the other activities, 35% for requirements tracing, 65% for project creation and 57% for negotiation are performed explicitly across all projects. Elicitation and validation involve the highest rate of the explicit effort while requirements tracing is mostly performed in an implicit manner with the smallest amount of resources devoted to it.

The distribution of RE activities in projects with external customers is significantly different from that in internal projects. Differences are markedly observed for the elicitation and tracing activities. For internal projects, the requirements elicitation is a major activity and is performed highly explicitly, for external projects this activity is considered a minor implicit activity. However, the tracing of requirements in external projects absorbs twice the resources and effort of that in internal projects and most of this effort is explicit.

In some projects, the performance of both explicit and implicit RE activities follows a recurring pattern, depending on the nature of the project. For instance, in the case of embedded system projects, the elicitation, negotiation and validation are always performed explicitly. This holds benefits for the gathering of accurate information for effective software design. No recurring patterns are uncovered in the explicit or implicit performance of RE activities at neither company nor industry levels. Our findings indicate that still many RE activities are performed implicitly.

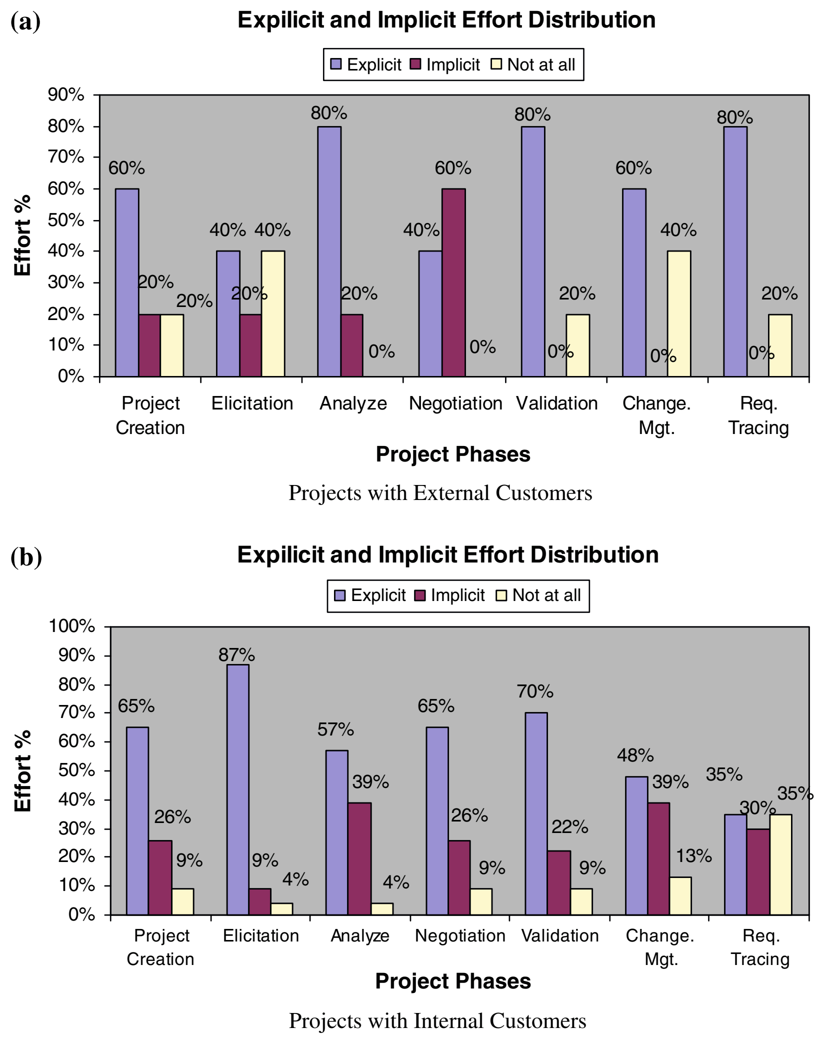
We believe that RE activities should be made as explicit as possible. This not only assists the project team in developing a clear idea of the process and the tasks, but this also supports organizational learning by providing the possibility of building an organization-specific repository of the know-how. Project teams can reuse the accumulated experiences from previous projects. This in turn increases the possibility of success of the software development project [20].

4.3 RE awareness

We identify five characteristics for each project with respect to RE: *Project Size, Activity Awareness, Role Awareness, Documentation Awareness and Process Awareness*. While the project size can be large, medium or small, as illustrated in Table 5, we only measure the remaining four characteristics in terms of High (H), Medium (M) or Low (L).

To study the relationship between practice and research, the participants are asked to identify the RE phases, to assign the RE activities to each phase, and to identify the roles and efforts in each activity. We investigate whether the identified RE activities are performed explicitly, implicitly or not at all. It is also examined how the RE process fits into the project lifecycle, if it is at the beginning of the project, at the

Fig. 2 Explicit and implicit efforts allocated to RE activities among all projects



(a) Projects with External Customers

(b) Projects with Internal Customers

beginning of each increment or a continuous activity during the whole project life cycle. Based on the data collected from the participants, high-level process models are constructed for each project and then compared to models in the existing literature.

- Project size is defined by the project's time span and the number of staff involved as illustrated in Table 5.
- **Activity awareness** is primarily determined by the number of explicit RE activities performed in the project. We classify the number of explicit and implicit RE activities for each project as *activity awareness*. Activity awareness is considered to be High, if there are >5 explicit but<3 implicit activities performed, Medium, if there are <5 explicit but >3 implicit activities performed and Low, if there are <3 any explicit or implicit activities performed during the project.
- **Role awareness** refers to the awareness of the project managers and is evident in the creation of RE roles within the project team. It is considered High, if it is assigned to at least three activities, Medium, if it is assigned to 1 or 2; otherwise it is considered Low.
- **Documentation awareness** relates to whether the necessary documentation is produced and exchanged between the team members throughout the software cycle. It is Low, if customers and developers have only a limited documented record of the project. It is High, if the documentation includes a requirements specification, legal documents, a risk list, a

Table 5 Criteria for project size

Project size	Small	Medium	Large	
Person month	<50	>50	<150	>150
Staff number	1–10	<25	>10	>25

glossary, a feasibility assessment, a description of supported business processes and a project/concept proposal.
- **Process awareness** means that there is a RE process in place and that this process has been documented. It is defined as a combination of explicit documentation of the RE process and of the project. We consider process awareness High, if the requirements tasks are defined explicitly, documented well and reflect the actual process (i.e. what actually happened during the project) without any deviation. We consider it Medium, if the process tasks are defined explicitly and documented well, but the process itself is not described in the documentation. Finally, it is considered Low, if there is an explicit process without documentation.

4.3.1 RE Process awareness

We ask whether the company has a standard process and/or if they have a structured process for RE. In most cases, the companies do not have a standard RE process definition and the structure of the RE process depends upon the project. One of the software developers remarks about her company's previous attempts at defining a formal RE process:

> ''We went through a phase where we tried to actually set something up. We had a team in IS and representatives from other groups, and started gathering information, but the timing wasn't good and people just didn't have time.... It's something that we saw, and still do see, as a need but it actually requires a lot of time and space to develop it and when you're in a time of great change in the department then it's not a good time to put it up, where we were at the time. Anyway, we attempted it and failed!''

There seems to be a general skepticism towards the need for a formal RE process, as is reflected by another developer's comments:

> ''It's a lot less structured now. Some of the design processes in terms of requirements were just too cumbersome, too structured and too lengthy. They required people to just fill in boxes without getting to the very meaning of what was in those boxes.''

In some cases developers from the same company show various level of awareness regarding standard documentation for the RE process. For instance, in one of the companies, developers indicate that there is a high-level, two-page RE process document based on Rational at the time of the project. Another from the same company points out that there is a little used standard RE process document of approximately 60 pages created by an external consultant. It is evident that it is rarely used, as the remaining participants are not aware of a standard RE process document. For example, one developer comments:

> ''Is there a requirements engineering process? No. Because each [project] is different, that's the point. It's very much a reflection of the actual customer''

The software developers understand the need for a structured approach to RE, but the result of standardizing the documentation and the process means to them that more time is spent complying with the process than understanding the underlying needs of the customer, as evident by the following comments:

> Comment 1: ''The design and build phase has got to be structured. It's got to be at least aligned with an overall picture of where the business wants to go and the product that you're putting in. But then in the post go-live phase it can be less structured but there is still that need to document what the requirements are.''
> Comment 2: ''I think increasingly people are getting more and more suspicious of grand projects and things like that. It's very hard to deliver that way, very hard to deliver to peoples' needs. Requirements are changing, they change all the time in business, if you're not flexible... and that's part of the problem with the current system, in that it's not flexible and there's no where to go''.

Time restrictions affect the RE process. Few developers note that since moving towards shorter development cycles with more releases, there has not been enough time to follow an extensive RE process with thorough documentation.

In general, RE is generally performed in a partially ad hoc manner, regardless of the industry. However, companies with mission critical projects showed a more structured approach towards RE than others. These companies followed a pre-defined RE process where the documentation contains a statement of the requirements and refers to further supporting documents. RE roles are explicitly defined for existing employees, who are actively encouraged to understand and document problems. Generally, past experiences with software projects have a relatively higher impact on such companies' approaches to the RE process than other factors, like company or project size. For example in one company, large projects are given a more structured approach than small ones, while in another company there is a strong RE oriented approach projects, regardless of their size.

In summary, in 50% of the overall projects, RE activities are run on an ongoing basis from the initiation of the project to its delivery. In the remaining 50% of the projects, RE is a dedicated phase either at the beginning of the project life cycle (43%) or at the beginning of each increment of the project (7%). In this half of the projects, later changes to requirements are handled through project change management.

4.3.2 Documentation awareness

We find that documentation awareness increase with project size, changing from low to high level. The findings showed that 36% of the projects are classified as having a High documentation awareness, 60% are classified as Medium and only 4% of the projects have a low documentation awareness. Some companies use highly developed templates for documenting their software projects. In particular, companies experienced in software development, with technical backgrounds demonstrate a more thorough and structured approach to documentation. The following results are obtained from the questionnaire:

- 56% of the projects document the business process
- 40% of the projects have a glossary
- 43% of the projects document stakeholder requests
- 56% of the projects document requirements specifications
- 35% of the projects document legal documents, e.g. contractual agreements
- 40% of the projects document a list of risks
- 36% of the projects generate a vision document

A short development life cycle is often regarded as an obstacle to the documentation by many participants e.g.

> ‘‘The requirements documentation is completely redundant, and for, I'd say, 8 out of 10 projects. Because we do such tight cycles.... there's no downtime ...They are good for reviews... but as actual specifications of applications they are not much use by the end of it.’’

Another developer articulates the negative affect of shorter development cycles on documentation as follows:

> ‘‘Because we do such tight cycles, you don't get a break between projects, there's no downtime. ...They are good for reviews and stuff, but as actual specifications of applications they are not much use by the end of it.’’

Shorter development cycles result in less time for documentation maintenance.

4.3.3 Role awareness

Given that RE is an activity in its own right requiring much coordination, it becomes increasingly important to assign the responsibility for overseeing the process to an individual/s. The role of the person who is responsible for the RE changes for each project. It is either a business systems analyst or a project manager, or no specific person is defined for the role. The number of team members in the project appears to affect whom the role is assigned to. When listing the roles responsible for the RE activities, the project manager of one of the projects examined comments:

> ‘‘A lot of this is the project manager because we were a very small team’’.

In other words, the project manager performs the RE activities. A number of projects, namely projects Ph1, Ph5, IT2, IT3, IT5 and Fin7 (see Table 2) do not have a dedicated RE role, despite an explicit RE process definition. The likelihood of responsibility for overseeing RE being assigned to a particular role increases as the number of team members increases. Only 30% of projects with less than ten team members have a dedicated role for the RE process. However, role awareness, defined as the number of roles assigned to RE activities, is not as strongly related to project size. We observe high role awareness regardless of the project size. About 70% of the projects with a small number of employees have a medium or low-level role awareness.

4.4 Discussion: RE practices in general

Despite some companies demonstrating an understanding of the importance of the RE process (often reflected in their explicit definition and documentation of the RE process), we question the degree of devotion these companies have to their process definitions. Our observations indicate loose adherence to these definitions. As discussed earlier, time, financial constraints and diversity of requirements are frequently the cause of deviation from documented RE processes.

Companies with more experience in software development projects or those who worked on mission critical projects generally follow a more structured approach to the RE process and RE documentation.

The number of employees working on a project is found to affect the existence and number of RE responsible roles in a project. In a considerable number of smaller projects, this role is merged with the project manager role, while we observe a greater tendency for a dedicated RE role in projects with a relatively large number of team members. Dedicated roles create visible accountability and responsibility and lead to the convergence of efforts from the earliest stages of the project and to the optimization of the use of resources. All of the projects observed share the common characteristic of having been successful (from the point of view of the developers involved). While evidence indicates that their ways to success vary, a dedicated role with responsibility for RE is one of the factors that improve product quality. In our opinion, the dedicated RE role generally increases the likelihood of the project's success.

We find that the degree to which RE activities are performed implicitly or explicitly relates to the degree of process awareness. The explicit definition of the requirements process and documentation decrease the likelihood of misunderstandings, mistakes and uncertainties related to the implicit RE process and positively affect the software quality. Moreover, the development of explicit knowledge from explicit activities will ideally facilitate learning about the RE process in general within companies.

Our study supports the finding that standardised RE activities are not essential to the project's success [32]. In some of the companies, we examine formal standards are not used and this is reported as not being a hindrance to the success of the projects. We do not identify communication or coordination as RE problems in any of the observed projects, despite these two factors having been introduced by Curtis et al. [12] as two out of three major RE problems, and despite the fact that such findings have been confirmed in other studies e.g. [16, 19, 30]. This may be because most of the projects we observed have internal customers on site with the developers. Direct and indirect involvements of the users in many of the projects facilitate a common understanding of the application and performance of the products. This ease of contact with the customer is reflected in medium and high levels of documentation in these projects. Nonetheless, regardless of the high volume of documentation, direct contact with customers in most projects reduces problems that can result from unclearly documented requirements/changes.

In most projects observed, the majority of those involved in the requirements process have worked for their companies for several years. A high degree of control over the goals of the proposed system and deep knowledge of the organizational procedures means that the projects enjoyed a high level of "*domain knowledge*" as one of the major sources for their success. This study could indicate that using domain experts, direct contact and ease of verbal communication with customers can highly compensate for the absence of a structured approach to the RE process and greatly increase a project's chances of success.

4.5 RE process models

Our final objective of this study is to construct high level descriptive RE process models for each company and then compare these models to frequently cited RE process models in the literature, selected for their different structures. The results are drawn from the questions in the questionnaire that asked the participants to describe the RE process and the activities in each phase of a specific project they played a part in. Examples of the process models identified by the participants are illustrated in Figs. 3, 4 and 5. The left hand side of the figures list the phases of the RE process (i.e. terminology used in the organization) in a particular project. The right hand sides of the figures display the RE activities (in rectangular boxes) performed at each phase (given in Sect. 3.3). Solid, straight and dashed lines are used to illustrate whether the activities are performed explicitly or implicitly. Note that the size of the boxes is of no importance. The order of the boxes shows the progression of the activities in each phase.

The RE process models allow the following useful observations:

- The developers have their own terminology, thus the questionnaire is redesigned to capture this information while allowing us to identify the common RE activities mentioned in Sect. 3.3.
- We find that a single process model for all projects for each company cannot not be produced, as they depend upon the nature of the project. Therefore, individual RE process models for each project are constructed. Each RE process is represented in a matrix, depicting the major steps in the RE process and the activities performed at each of those steps.
- When we compare the RE process models (of projects) to the process models from the literature, we find that none of the models can be universally

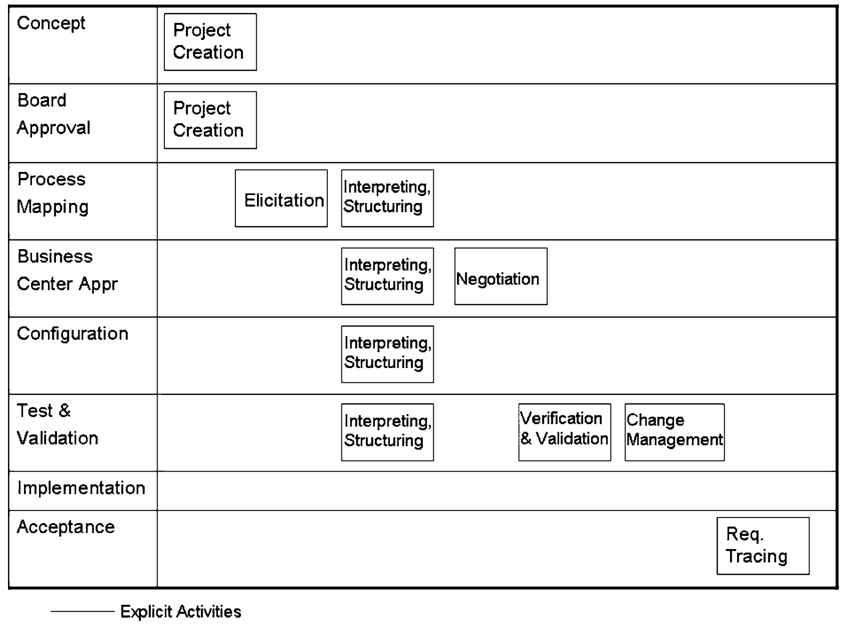


Fig. 3 Process for one pharmaceutical project

mapped to every RE process identified, however the characteristics of each model make them suitable to particular processes on a case-by-case basis only.

- Projects where RE is seen as an ongoing task throughout the project tend to have an iterative RE process with activities performed in multiple phases. In contrast, projects with RE performed at the start of the project follow a more linear process.

This study confirms the high degree of dependency between the RE process and its context that prevents the use of a single model for each company or across an industry group. For instance, despite the existence of a recommended framework for requirements specifications, design and testing for a manufacturer of drugs in Australia defined by GAMP (Good Automated Manufacturing Practice, http://www.ispe.org/gamp/) (a V-Model [8]), none of the pharmaceutical companies in our study employ this framework for the RE process. Figure 3 illustrates the constructed process model for one of the pharmaceutical projects. Notably the process is represented by a conceptual linear model [46] rather than a V-Model.

The structure of the RE process model differs across projects, even within a company. Figure 4a, b indicates the structure of the RE process for two different projects within one company. The structure of (a) shows similarities to a linear model while (b) can be better described by the nonlinear model introduced by Locoupoulos and Karakostas [25].

As mentioned above, none of the process models that are selected from the literature (i.e. the spiral model [7], nonlinear model [25], linear model [26], W-Model [8], standard waterfall [43] and conceptual linear model [46] can represent all of the RE processes undertaken in the observed projects. Moreover, none of the models is capable of completely representing a single actual RE process. In some cases, project size is an influential factor in determining how RE would be carried out, whilst in others it is not. In one of the IT outsourcing companies, two projects have the same structure despite differences in their size.

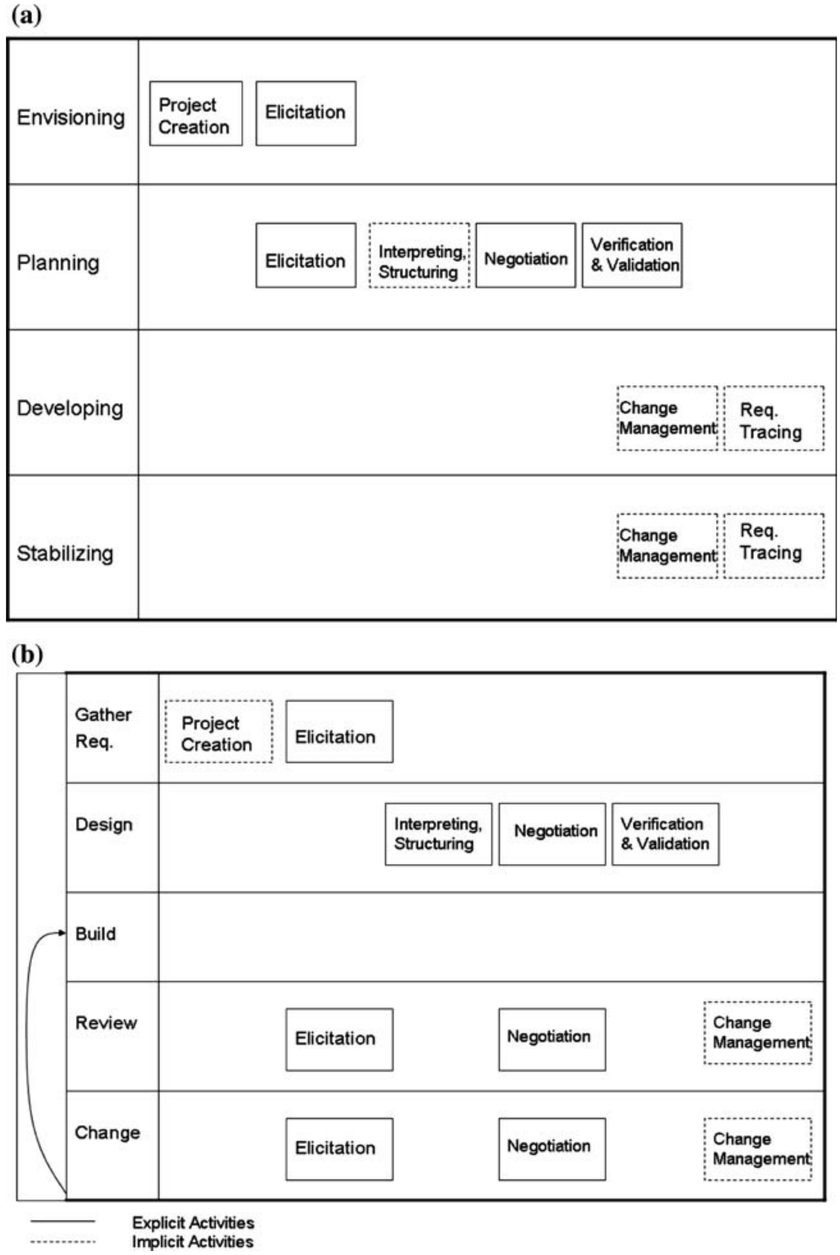
Some projects use a hybrid of two models. Figure 5 depicts a project where activities across the phases of the project have a linear structure; however, validation and verification have been performed starting from early phases which is common in a "V" model.

Despite the findings of previous studies [32] this study does not reveal any evidence of the popularity of the waterfall model. Even within a single company, it is not possible to construct a single model to represent RE practices and processes. Each model is developed separately for each project. That is, none of the selected RE literature process models show adequate flexibility and adaptability in their current state to meet industry software development project needs. This study supports the idea that RE processes are highly dependent on the context in which RE takes place and thus suggests the inapplicability of a universal approach to RE.

5 Validity threats

We consider the validity of our research in four perspectives:

Fig. 4 Processes for different projects within one company

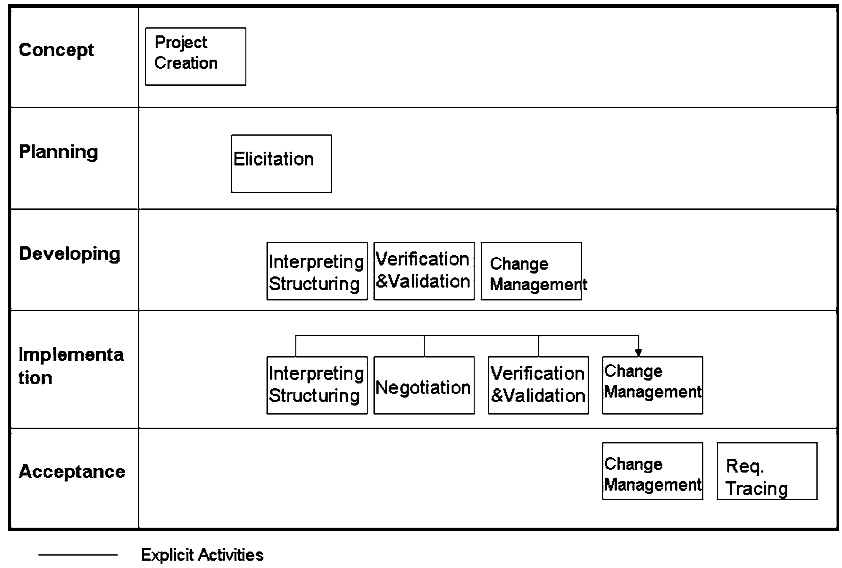


Construct validity This indicates the relation between theories or constructs behind the research, and the observations that are made. The variables in our research are measured through a questionnaire and interviews. While questions guide the interview process, there are open-ended aspects where the participants are asked to express their own opinions. Poor or inaccurate verbal reporting is avoided by recording the interviews. The collection of data from a wide range of sources on the topic of study assists with avoiding a ‘‘mono-operation bias’’ [55].

Internal validity This was not a major validity threat because it is not our primary aim to establish causal relationships. However, whenever possible, the study examines significant relationships between pairs of variables and investigates whether those relationships could be explained through causal links. Thus, while internal validity remains a minor concern, any threat of stating a non-realistic causal relationship is mitigated by exercising caution when selecting appropriate non-parametric statistical techniques as a complementary tool in the analysis phase.

Other threats to internal validity include instrumentation, maturity and selection threats. In our study, using a previously administered and validated questionnaire (in Germany) mitigate the instrumentation

Fig. 5 Process with a hybrid structure



threat. Maturity threats are handled by reducing the duration of interview sessions using close-ended questions, preventing the participants from losing interest or motivation. Selection threats are reduced by narrowing the choice of the participants to only those who are highly involved in RE process in the companies.

Conclusion validity This is the ability to draw accurate conclusions [55] Reliability of the measurement is an important contributor to this kind of validity. Hence, the research instrument is considered an important factor. The reliability of the questionnaire used in this study is confirmed in a previous German study [37, 38]. A standard environment is maintained for each interview to preserve consistency. Hence, our data is not adversely affected by an unreliable research method.

External validity This threat is concerned with the ability to generalize the findings beyond the actual study. In this study, selection bias and sample size are considered to have influence on the external validity. To avoid a selection bias, the participants of this study are selected according to their roles within the software team and based on their experience. However, the small sample size is a major threat to the external validity of this study. It may have affected the conclusions we have drawn and hence this research cannot, and should not, be taken as representative of the Australian software engineering industry as a whole. It highlights the need for replication of this type of study. Finally, skills and experience levels of team members and managers as well as differences in team cohesiveness and synergy from project to project are factors which have been ignored in this study. These can potentially confound our analysis and indeed present limitations to our study.

6 Summary, conclusion and future study

This article presents the results of an empirical study that examines RE practices in 16 Australian companies. Data is collected from 28 successful software projects at the companies. The majority of project customers are internal and the projects are aimed at the development of a software product which supports the daily business of the company. Data is collected from several industries including finance, telecommunication, pharmaceutical and healthcare services, the food industry and education. To the best of our knowledge, there are no other field studies on this scale that examine RE practices in Australian companies.

We study the distribution of effort within RE activities. The findings reveal that time, diversity of requirements and financial constraints are among the factors that diverted efforts to the entire project, instead of the RE process itself. The findings highlight two important issues: (a) the nature of the projects, whether they are internal or external, played an important role in the effort distribution, (b) there is a trade-off between RE activities for both internal and external projects. In other words, when an activity is avoided, or developers spend less time on a certain activity in a project; this gets covered in a following phase. In this study, companies with greater technical maturity demonstrate a more structured approach towards RE activities. Although we did not find any consistent patterns between the explicit or implicit RE activities at a company or industry level, in all internal projects, the number of explicit activities is considerably higher than that of implicit activities. We assert that a domain specific RE process model is needed, allowing for iterations, emphasizing the explicit undertaking of RE activities, which are significant features for RE in a specific domain. We argue that if change management is highly involved, that is more changes are being introduced, implying new requirements and from this different or new implementations of the system, then the level of effort allocated to requirements tracing should always be explicit and should reflect the effort allocated to change the management. It also becomes the responsibility of organizations to formalize their own best practice RE process, which should be used consistently when developing systems in a specific domain. An RE process model encouraging explicit (as opposed to implicit) undertaking of RE activities helps to maintain and spread knowledge through the phases of system development, allowing knowledge sharing among stakeholders directly involved in the development of manufacturing systems.

We investigate several characteristics of each project with respect to RE activities. These characteristics include activity awareness, role awareness, documentation awareness and process awareness. Our findings show that mission critical projects and external projects involved a more structured approach to the RE process than others. While RE occurred in all projects, the general feeling from the interviews is that the RE process is seen as one aspect of the entire software process, rather than a process in its own right. The formal term “Requirements Engineering” is generally not well known, but the importance of gathering requirements is. While document awareness increases with project size, where the descriptions of business processes and requirements specifications are the most common documents, role awareness, i.e. the number of the roles assigned to RE activities, is not related to project size.

We examine the RE process model used at each company for each project and we compare this with RE process models in the literature, namely linear, waterfall, spiral, V and W models. We find that even pharmaceutical companies who develop safety critical products, do not necessarily follow the recommended RE process model during software development. Furthermore, even for a single type of project, there is no consistent process model used across any of the companies for representing RE practices. In other words, none of the selected models from the literature is able to represent or explain exactly what happens in these projects. Our findings indicate that RE processes are highly dependent on their context (including the nature of the project and the industry). Our results support the method engineering endeavor proposed and developed in [9, 42]. This endeavor supports the creation of a complete methodology out of portions of the so called fragments [40] of existing methodologies according to the development context. Our work can contribute to this endeavor by identifying the suitable RE fragments towards the assembled methodology.

Providing a detailed analysis of RE activities in industry, the value of this paper lies in its (partially) anecdotal discovery of useful issues to be considered in a more thoroughly controlled study. We expect that this study will enhance our academic knowledge of the current state of the art of RE practices in general. Not only will this benefit RE practitioners to provide insight into the factors that influence their effectiveness (e.g. by facilitating better planning), but it will also assist the practitioners and researchers to share their views. Indeed, we have established a baseline for further research providing additional validation of the findings reported in this article.

References

1. Alexander IF, Stevens R (2002) Writing better requirements. Addison Wesley, UK
2. Arthur JD, Gröner MK (2005) An operational model for structuring the requirements generation process. Requir Eng J 10:45–62
3. Aurum A, Jeffery R, Wohlin C, Handzic M (2003) Managing software engineering knowledge. Springer, Berlin
4. Aurum A, Wohlin C (2005) Engineering and managing software requirements. Springer, Berlin
5. Berntsson-Svensson R, Aurum A (2006) Successful software projects and products. In: 5th international symposium on empirical software engineering (ISESE’06), 21–22 September, 2006, Rio-de Janeiro, Brazil
6. Boehm B (1987) Improving software productivity. IEEE Comput 20(9):43–57
7. Boehm B, Papaccio P (1988) Understanding and controlling software costs. IEEE Trans Softw Eng 14(10):1462–1477
8. Boehm B, Egyed A, Kwan J, Port D, Shah A (1998) Using the Win Win spiral model: a case study. IEEE Comput 33–44
9. Brinkkemper S, (1996) Method engineering: engineering of information systems development methods and tools. Inf Softw Technol 38(4):275–280
10. Chatzoglou P (1997) Factors affecting completion of the requirements capture stage of projects with different characteristics. Inf Softw Technol 39(9):627–640
11. Chatzoglou P, Macaulay L (1996) Requirement capture and analysis: a survey of current practice. Requir Eng J 1(2):75–87
12. Curtis B, Krasner H, Iscoe N (1998) A field study of the software design process for large systems. Commun ACM 31(11):1268–1287
13. Davis A (1993) Software requirements: objects, functions, and states. Prentice-Hall, Engelwood Cliffs
14. Davis A, Hickey A (2002) Requirements researchers: do we practice what we preach? Requir Eng J 7:107–111
15. Dourish P, Bellotti V (1992) Awareness and coordination in shared workspace. In: Proceedings of CSCW’92. ACM Press, New York, pp 104–114
16. El Emam K, Madhavji N (1995) A field study of requirements engineering practice in information systems development. In:

Proceedings of the 2nd international symposium on requirements engineering, York, England, pp 68–80
17. Endres A, Rombach D (2003) A handbook of software and systems engineering—empirical observations, laws and theories. Pearson Education, Boston
18. Gutwin C, Roseman M, Greenberg S (1996) An usability study of awareness widgets in a shared workspace groupware system. In: Proceedings of CSCW '96. ACM Press, New York, Cambridge, pp 258–267
19. Hall T, Beecham S, Rainer A (2002) Requirements problems in twelve companies: an empirical analysis. IEEE Proc Softw 149(5):153–160
20. Hoffmann H, Lehner F (2001) Requirements engineering as a success factor in software projects. IEEE Softw 18(4):58–66
21. Houdek F, Pohl K (2000) Analysing requirements engineering processes. In: Proceedings of 11th international workshop on database and expert systems applications, pp 983–987
22. Jorgensen M, Sjoberg D (2001) Impact of effort estimates on software project work. Inf Softw Technol43:939–948
23. Kaindl H, Brinkkemper S, Bubenko J (2002) Requirements engineering and technology transfer: obstacles, incentives and improvement agenda. Requir Eng J 7:113–123
24. Liechti O (2000) Awareness and WWW: an overview. Sig Group Bull 21(3):312
25. Locoupoulos P, Karakostas V (1995) System requirements engineering. Mc Graw-Hill, New York
26. Macaulay L (1996) Requirement engineering. Springer, Berlin
27. MacDonell S, Shepperd M (2003) Using prior-phase effort records for re-estimation during software projects. In: 9th international symposium on software metrics. Sydney, Australia, pp 73–86
28. Madhavji NH, Holtje D, Hong W, Bruckhaus T (1994) Elicit: a method for eliciting process models. In: CAS Conference CASCON'94. Toronto, Canada
29. Martin S, Aurum A, Jeffer R, Peach B (2002) Requirements engineering process models in practice. In: 7th Australian workshop on requirements engineering. Deakin University, Melbourne, Australia, pp 41–47
30. McPhee C, Eberlein A (2002) Requirements engineering for time -to-market projects. In: 9th annual IEEE international conference on the engineering of computer based systems. Lund, Sweden, pp 17–26
31. Morris P, Masera M, Wilikens M (1998) Requirements engineering and industrial uptake. In: 3rd international conference on requirements engineering. Colorado Spring, CO, USA, pp 130–137
32. Neil C, Laplante PA (2003) Requirements engineering: the state of the practice. IEEE Softw 20(6):40–45
33. Nikula U, Saieniemi J, Kalvianen H (2000) State-of-the-practice survey on requirements engineering: industry education and technology transfer. In: 5th international conference on software process improvement research, education and training (INSPIRE). The University of North London, London, Great Britain, pp 13–24
34. Nguyen L, Swatmann P (2001) Managing the requirements engineering process. In: 7th international workshop on requirement engineering: foundations for software quality. Interlocken, Switzerland
35. Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. ACM Press, New York
36. Ohlsson M, Wohlin C, Regnell B (1998) A project effort estimation study. Inf Softw Technol 40:831–839
37. Paech B Koenig T (2004) Charakterisierung und Probleme von Anforderungsprozessen deutscher Unternehmen-Auswertung einer Erhebung. In: 3rd HOOD requirements engineering conference (http://www.reconf.de), Munich, Germany, pp 86–90
38. Paech B Koenig T, Borner L, Aurum A (2005) An Analysis of empirical requirements engineering survey data. In: Aurum A, Wohlin C (eds) Engineering and management software requirements. Springer, Berlin, pp 427–450
39. Prakash S, Aurum A, Cox K (2004) Benchmarking a best practice requirements engineering process for pharmaceutical and healthcare manufacturing. In: 11th Asian-Pacific conference on software engineering, Busan, Korea
40. Ralyté J, Rolland C (2001) An assembly process model for method engineering. In: 13th conference on advanced information systems engineering (CAiSE). Springer, Berlin
41. Reifer D (2003) Is the software engineering state of the practice getting closer to the state of the art? IEEE Softw 20(6):78–83
42. Rolland C, Prakash N (1996) A proposal for context-specific method engineering. In: Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on method engineering: principles of method construction and tool support: principles of method construction and tool support, Atlanta, USA. Chapman and Hall, UK, pp 191–208
43. Royce W (1970) Managing the development of large software systems, concepts and techniques. In: 9th international conference on software engineering, ACM Press, Montery, USA, pp 328–339
44. Sawyer P, Viller S, Sommerville I (1998) Requirements process improvement through the phased introduction of good practice. Isoftw Process Improv Prac J 3(1):19–34
45. Smith JL, Bohner SA, McCrickard DS (2005) Project management for the 21st century: supporting collaborative design through risk analysis. In: Proceedings of the ACM southeast conference (ACMSE '05), Kennesaw GA, pp 2-300–2-305
46. Sommerville I, Sawyer P (1997) Requirements engineering: a good practice guide. Wiley, Chichester
47. Sommerville I, Kotonya G (1998) Requirement engineering-process and techniques. Wiley, Chichester
48. Sommerville I, Ransom J (2005) An empirical study of industrial requirements engineering process assessment and improvement, ACM Trans Softw Eng Methodol 14(1):85–117
49. Spillner A, Bremenn H (2000) The W model: strengthening the bond between development and test, Workshop on testing non-functional software requirements, Nuremberg, pp 221–231
50. Standish Group. CHAOS, http://www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf (Accessed Feb 16, 2006)
51. Theofanos M, Pfleeger S (1996) Wavefront: a goal-driven requirements process model, Inf Softw Technol 38:507–519
52. Tiwana A (2004) An empirical study of the effect of knowledge integration on software development performance. Inf Softw Technol 46(13):899–906
53. van Lamsweerde A (2000) Requirements engineering in the year 00: a research perspective. In: 22nd international conference on software engineering, Limerick, pp 5–18
54. Weisband S (2002) Maintaining awareness in distributed team collaboration: implications for leadership and performance. In: Hinds P, Kiesler S (eds) Distributed work. MIT, Cambridge, pp 311–333
55. Wohlin C, Runeson P, Host M, Ohlsson M, Regnell B, Wesslen A (2000) Experimentation in software engineering—an introduction. Kluwer, London