Software Qual J (2008) 16:459–481
DOI 10.1007/s11219-008-9045-8

# Cost estimation for cross-organizational ERP projects: research perspectives

**Maya Daneva · Roel Wieringa**

Published online: 15 March 2008

**Abstract** There are many methods for estimating size, effort, schedule and other cost aspects of IS projects, but only one specifically developed for Enterprise Resource Planning (ERP) (Stensrud, Info Soft Technol 43(7):413–423, 2001) and none for simultaneous, interdependent ERP projects in a cross-organizational context. The objective of this paper is to sketch the problem domain of cross-organizational ERP cost estimation, to survey available solutions, and to propose a research program to improve those solutions. In it, we: (i) explain why knowledge in the cost estimation of cross-organizational ERP is fragmented, (ii) assess the need to integrate research perspectives, and (iii) propose research directions that an integrated view of estimating cross-organizational ERP project cost should include.

**Keywords** Enterprise resource planning · Cost estimation models · Cross-company estimation

## 1 Introduction

The integration strategies of today's networked organizations depend on process-oriented multi-component ERP solutions which, in practice, often do not meet critical requirements and cost more in terms of resources than originally planned. Many adopters of cross-organizational ERP systems agree, therefore, on the need to include a number of tasks in the requirements engineering (RE) process, namely: (i) assessing whether or not the ERP implementation will meet schedule and budget targets; (ii) assessing customization risk; and (iii) mitigating risk. Moreover, benchmarking experts in the software industry have determined that project cost estimation is the second most important in a set of 12 major factors

M. Daneva (✉) · R. Wieringa
Department of Computer Science, University of Twente, Drienerlolaan 5,
7500 AE Enschede, The Netherlands
e-mail: m.daneva@utwente.nl

R. Wieringa
e-mail: roelw@cs.utwente.nl

associated with both success and failure (Jones 2004, 2005). Empirical studies also indicate that both ERP vendors and ERP clients are starting to team up to identify, model, and evaluate cost factors contributing to the ERP return-on-investment equation (Davenport 2000; SAP SI 2004; Stensrud and Myrtveit 2003). Awareness of the current need for ERP cost estimation models has also been raised by some professional bodies, like Management Accountants of Canada and US Controllers, who are now offering systematic help to ERP adopters in business case analysis of their ERP spending (Epstein and Rejic 2005).

Despite significant discussions in both the academic and management literature concerning the importance of ERP cost factors, we as an RE community have not adopted an integrated approach to modeling and evaluating the relationships among size, effort, duration, productivity, and other cost aspects of cross-organizational ERP implementation projects. By failing to do so, we are missing an important opportunity to improve the existing level of knowledge in the field and to make a major contribution in an area of pressing business need. Clearly, for cross-organizational ERP implementation projects, there is a shortage of both relevant metrics and historical project datasets, and an absence of proper methodologies to evaluate size, effort, productivity, and other cost factors for such projects. Even approaches such as the COCOMO family of models (Abts et al. 2000; Boehm et al. 2000b), which do provide frameworks for analysis and management, are not suitable for cross-organizational projects. This is because any cross-organizational ERP project is always part of a business transformation initiative, which implies that the project not only deliver business functionality, but also bring with it changes in organizational structures, data, and business processes. Therefore, effort, productivity, and cost modeling have to be multidimensional, just like the project outputs (Stensrud 2001).

The objective of this paper is to sketch the problem domain of cross-organizational ERP cost estimation, to survey available solutions, and to propose a research program to improve those solutions. In Sect. 2, we provide background on what cross-organizational ERP systems are and how ERP projects are different from other types of business information systems and custom software projects in regard to cost estimation. In Sect. 3, we explain why we think that existing approaches to modeling relationships among size, effort, and other cost factors in cross-organizational ERP implementation have been quite fragmented. We also make the case that both researchers and business people need a more integrated view of this topic. Section 4 brings together some issues that we have selected to build an agenda for research in ERP sizing and cost modeling in cross-organizational project settings. In Sect. 5, we examine some efforts which have been made at conceptual integration and show where they need bolstering to create an integrated framework for cross-organizational ERP size/cost modeling. Section 6 summarizes our contributions.

## 2 Cross-organizational enterprise resource planning systems

This section could be considered a sidebar for readers who are less familiar with cross-organizational ERP systems and ERP projects. Our motivation for including a discussion on ERP systems and what makes ERP projects unique is to help them understand Sects. 3–5 and to avoid misunderstandings.

ERP systems are packaged software solutions, the key function of which is to coordinate work in a business. They are the vehicles modern organizations use to achieve true business connectivity, a state in which everyone knows what everyone else is doing in the business all over the world and at the same time. In the past ten years, the importance of ERP systems to companies has dramatically increased as companies have begun to realize how decisive the impact of ERP is on their future: organizations and their business partners and customers have started developing ‘value webs’, and ERP systems have become the tool of choice for obtaining the coordination support that companies need and for staying interconnected (Benchmarking Partners & IBM 2000; Daneva and Wieringa 2006b; Davenport 2000; Holland et al. 2005). By ‘value web’, we mean a set of different, independent (or nearly independent) businesses forming a business network—for example, the business value web of WalMart Stores, Inc., which collaborates with a large number of non-U.S. companies and gives them direct access to the American market (Daneva and Wieringa 2006a). A value web can also be any large company which has restructured as a set of nearly independent business units, each responsible for its own profit and loss. For example, British Petroleum retooled its asset management division through an SAP solution to link all its 1,500 suppliers involved in the maintenance, operation, and repair of specialized production equipment (Holland et al. 2005).

We define a cross-organizational ERP system as an information system that consists of one or more standard ERP software packages, and automates cross-organizational process work flows and data control flows composed of flow fragments owned by, or shared among, multiple companies in a value web. The packages in a cross-organizational ERP system may or may not all be shared by the participating companies; and they may each be provided by the same vendor or by different vendors, and each has its own application logic, data formats, and data semantics.

An ERP implementation project is the customization and introduction of a cross-organizational ERP system in a value web. Our research effort is focused on investigating measurement models which can be used for ERP project cost estimation at the requirements engineering stage of the project; for example, estimating effort at the ERP bidding stage, at which point requirements are not yet fully known. Following (Kitchenham et al. 2003), we consider a project quote to consist of three components: estimated cost, profit, and contingency. Here, however, we look at the measurement models used to estimate cost in particular, and, for this reason, we leave aside profit and contingency.

Literature sources (Arnesen and Thompson 2005; Daneva and Wieringa 2006a; Daneva and Wieringa 2006b; Davenport 2000; Eschinger 2004; Holland et al. 2005; O’Neil 2002; Rohde 2005; Stamelos et al. 2003; Stensrud 2001; Stensrud and Myrtveit 2003; Vogelesang 2006) comparing ERP projects to other projects indicate that, unlike business information systems projects (e.g. data warehousing or workflow management systems) or custom software projects, ERP projects:

(i) are broad in terms of functionality, covering thousands of business activities;

(ii) treat the cross-organizational business processes in a value web as the fundamental building blocks of the system;

(iiii) deliver a shared system which lets the business activities of one company become an integral part of the business of its partners;

(vi) create system capabilities far beyond the sum of the ERP components’ individual capabilities, which, allows the resulting system to qualitatively acquire new properties as result of its configuration;

(v) may well include diverse configurations, each of which matches the needs of a unique stakeholder group, which, in turn, implies the presence of cost drivers unique to each configuration;

(vi) deliver a system which is far from complete once the ERP project is over, because an ERP solution must mirror rapidly-changing business requirements, and so be adjusted regularly to accommodate current business needs;

(vii) do not have an identified owner at cross-organizational system level, as the system is shared;

(viii) may well have a low level of organizational awareness of what new project activities (e.g. identifying and analyzing capability gaps, investigation and mapping of configuration options (Kelly and Holland 2002; Rolland and Prakash 2000)) are to be added in order to plan and manage the ERP project, and what the factors are that drive effort for these new activities.

(ix) are not "built" in the sense that a master architect envisions the parts and their relationships; rather they evolve into existence and change through their life cycles as new shared pieces of functionality are built, existing intra-organizational systems connect to become shared, and shared parts of the system are disintegrated as soon as needs of sharing processes and data disappear.

The analyses by the above authors suggest that these characteristics pose cost estimation challenges which are well beyond those encountered in ordinary business information systems or custom projects. The authors provide evidence that these characteristics make it almost impossible for ERP adopting organizations to determine a level of trust in any estimate. Examples of some specific barriers to trust, which researchers (Arnesen and Thompson 2005; Davenport 2000; O'Neil 2002; Seaver 2005; Stamelos et al. 2003; Stensrud 2001; Stensrud and Myrtveit 2003) have found to be traceable to the above ERP project characteristics, include: lack of consensus on the objectives of the estimates, no known steps to ensure the integrity of the estimation process, no historical evidence at the ERP adopter's site supporting a reliable estimate, or the inability to clearly see whether or not estimates are consistent with consultants' demonstrated accomplishments on other projects in comparable organizations in the same business sector.

The point made by these authors is not simply that ERP projects are different; it is that the assumptions which we usually make when we estimate project cost do not apply. Clearly, a project cost analyst needs to know both the cost estimation vehicles at his disposal (or the "cost calculation machines" (Stensrud 2001) as Stensrud calls them) and whether or not the implicit and explicit assumptions about the use of these vehicles match the project environment. We will see in the next sections that most of the assumptions we typically make do not apply to a shared ERP solution of systems which is produced by vendors with independent governance, for existing business missions and missions yet unformulated, to interoperate with legacy applications and applications not yet conceived, with boundaries not entirely agreed to and process and data flows that may not be knowable in advance.

## 3 Building ERP cost estimation models in the literature and in business practice

This section provides an overview of what is contained in state-of-the art publications and in practice in the field of modeling the relationships between size and cost factors. Then, we use an example to: (i) illustrate some impacts that poor project resource planning might have on the organization's bottom line; and (ii) emphasize the need to increase our knowledge of the relationships between size and cost factors in cross-organizational ERP implementation.

### 3.1 Observations from the RE, IS, and ERP literature

Throughout the past 40 years, interest in size and cost estimation has generated considerable activity and many IS, and software engineering researchers have established stepping-stones for cost estimation models (Boehm et al. 2000a). There is a sizeable literature on size, effort, and cost estimation models (Boehm et al. 2000b; Jones 1997; Putnam and Myers 1992), each of which has at least passed a graded test for value by remaining economically viable over a minimum of a decade. One observation which appears to be consistent across these models is that the predominant paradigm used for accounting costs in IS projects is the manufacturing one, in which costs are relative to the size of the product built. The main cost driver assumed in a variety of models is the size of the product, whatever the measure may be (e.g. function points, future points, or lines of code). However, empirical studies indicate that in the ERP context the manufacturing paradigm is probably not the most adequate metaphor for portraying modern—specifically cross-organizational—ERP solutions, and that the results of this paradigm cannot ensure a specific form of the cost function (Stamelos et al. 2003; Stensrud and Myrtveit 1998). This is partly because none of the existing estimation methods has been specifically devised for the ERP setting, and because very few studies have created and used ERP datasets for validation (Stensrud 2001). Consequently, the literature leaves ERP estimators with limited confidence in the empirical results. The only non-empirical evaluation of both commercial and non-commercial estimation techniques of which we are aware was published by Stensrud (Stensrud 2001), who performed an initial screening based on theoretical analysis and delivered a short list of candidate approaches which it made sense to consider for inclusion in an integrated framework for ERP cost estimation.

### 3.2 Observations from ERP implementation practice

The software metrics discipline (Boehm and Sullivan 2000; Boehm et al. 2000b; Jones 1997) does offer a few recognized methods for modeling size and resources, but, although these have the potential to fit the ERP RE practice (Stensrud 2001), they do not seem appealing from the standpoint of senior management responsible for approving investments in ERP technology. In networked settings, key decision-makers tend to get on board only when approving a new partnership or a new development or enhancement project. They rarely become involved in cross-organizational ERP operational issues beyond their own organization, unless there have been significant problems, like a major operational failure or a serious security breach. Thus, senior managers tend to be detached from making decisions regarding operational ERP costs. Moreover, evidence from ERP case studies suggests that organizations often overlook rationality in justifying ERP spending (Davenport 2000; Epstein and Rejic 2005; SAP SI 2004). Many companies tend to accept ERP expenses, however much they are, as the operational cost of running a business in the 21st century.

There is agreement in the ERP literature (Davenport 2000; Epstein and Rejic 2005) that the most common analysis that companies make prior to ERP RE is *cost versus benefits*, the purpose of which is to determine whether it would cost an organization more to implement an ERP system than the value of the benefits that would ensue. Typically, this starts with a business case focused on the business value to be achieved and the associated business changes. It may also include both financial and non-financial factors. Many ERP adopters see the business case analysis as a one-time exercise performed even before package selection, when most of the requirements are unknown. However, the experience of successful ERP projects indicates that, for the business case analysis to be effective, it needs: (i) to be re-done in ERP RE; (ii) to take place at each stage of the implementation; and (iii) to involve decisions about how, and whether or not, to proceed with the ERP solution (Davenport 2000). The latter does not necessarily amount to an all-or-nothing proposition.

With big budgets invested, it will usually make sense to continue the project. However, from senior management's standpoint, it may also make sense to change the scope or the scale of the project based on ongoing business case analysis.

One approach which has been used in practice and has proven its worth in ongoing business case analysis (Bardhan et al. 2004; Benaroch and Kauffman 1999; Davenport 2000; Taudes et al. 2000) is the real-options approach developed by Amram (Amram and Kulatilaka 2000). It views ERP projects as investments and aims at capturing, in a disciplined way, the value created by those ERP investments that deliver platforms of business flexibility. For example, an investment in a data warehouse creates value through the ability to quickly develop new applications, and an investment in an open architecture creates sourcing options for all the partners in the business network. The challenge, however, is to move beyond these intuitions to find an efficient and practical way to apply real-options thinking to ERP investment (Amram and Kulatilaka 2000).

Amram's method is appealing to networked businesses because it makes it possible to: (i) assess opportunities which ERP systems offer to flexibly adjust the scale of the business over time; and (ii) easily and quickly make follow-on investments which could not have been achieved if the ERP business platform had not been put in place. Although the merits of the real-options approach have been proven in some real-life projects (Davenport 2000), in practice, the majority of today's ERP adopters still rely on intuition and rules of thumb which define relationships between cost factors. These rules are typically based on two commonsense assumptions: (i) cost factors can be split into three categories, namely, hardware costs, license costs, and solution implementation costs (Arnesen and Thompson 2005; Benchmarking Partners & IBM 2000); and (ii) while the first two can be viewed as well understood and could be kept within manageable limits by most ERP adopters, the solution implementation costs can severely impact the project and turn it into a success or a failure.

Furthermore, most ERP adopters define implementation costs as the costs required to: (i) customize the system to the organization; (ii) install it; and, most importantly and expensively, (iii) bring about the process, coordination, and strategic changes required to really make the solution a worthwhile business investment. Empirical studies agree that these costs are a function of the business change envisioned, and they can well exceed ten times the cost of hardware and software licenses (Benchmarking Partners & IBM 2000). If the ERP adopter does not include process change or does not introduce new ways of working, the costs would only double (Davenport 2000).

## 3.3 Need for an integrated framework

Our review of literature sources and state-of-the-art practices indicates that, with very few exceptions, most ERP adopters lack a way to: (i) communicate size, effort, duration, and other cost aspects of their projects; (ii) identify the most important risks for business managers; and (iii) easily determine the monetary consequences of those risks.

The case of Hershey Foods (Carr 2002; Collett 1999; Koch 2002; Sarrat 2002), a Fortune 500 company, is an illustration of how a business network can underestimate costs and sub-optimize total business risk while trying to manage project metrics like effort, productivity, cost, and schedule. When Hershey's business units set themselves the goal of moving to an ERP solution, senior management were aware that it was a risky proposition. Yet the big and complex SAP project in the areas of customer service, warehouse management, and order fulfillment was complemented by planning and transportation management software from Manugistics Group Inc. and a customer relationship management module by Siebel. During the implementation process, the company had to switch to a new set of SAP requirements and opted for a compressed schedule to complete the project in 39 months instead of 48 (Carr 2002). The project ran behind the aggressive schedule and the full solution did not come online until 6 months or so later. Included in the delayed conversion were SAP's critical order processing and billing systems, along with the Siebel and Manugistics application components. The timing meant that Hershey would be facing a major problem because peak season orders were already arriving and could not be processed. IT management chose to convert all these new systems using the direct cutover strategy in which the systems of all three vendors go live all at once. While this strategy is generally considered to be the most risky one, were it to be successful it could save the company time and money while enabling Hershey to fill their peak season orders on schedule. Problems arose for Hershey when the cutover strategy did not work because of immediately emerging serious issues with the customer service, warehouse, and order fulfillment components of the new solution which did not perform as planned. Business users found that critical requirements of how SAP and Manugistics components should work in integration had been consistently overlooked since the beginning of the project. Hershey had been trying to accomplish too much in the allotted time frame, and had been rushing through conversion requirements, change impact assessment, and testing, and ultimately, had been cutting corners on training which left users with wrong assumptions about how they had to change the way they worked (Koch 2002). Direct cutover to the new system resulted in disaster. As a consequence, Hershey found themselves unable to shift stock through their distribution channels, and many Hershey customers found their shelves empty as the peak season approached. Hershey reported a 19% decline in the 3rd quarter of 1999 and their stock price went into a sharp slide, its price falling to $47.50, down 35% from $74 one year earlier. The company sustained unrecoverable costs, blaming the drops in sales on the new SAP solution (Collett 1999).

As in any example of ERP failure, multiple factors conspired to prevent Hershey from succeeding. Some of these were in the control of senior managers, others were not. But the Hershey case clearly illustrates that size, effort, productivity, and other project cost factors can turn into issues with much broader consequences. One may wonder if Hershey's strategic leadership team would have been so eager to advance the project schedule if they had taken a serious look at the size, effort, cost, and customization risk factors, as well as the likelihood and consequences of operational failure owing to poor project resource estimation.

Our review of the literature and of practice has helped us identify four aspects that make size, effort, and cost modeling difficult in cases like Hershey's:

1. A cross-organizational ERP solution is never a stand-alone investment. It could: (i) be part of a product development initiative; (ii) enable financial controls; or (iii) require investment in some non-ERP solution. For these reasons, putting numbers on the size, effort, productivity, and other cost factors of an cross-organizational ERP project can be very difficult: the project does not always contain a set of well-defined requirements.
2. As an ERP-project spans technology and business issues, fragmented concepts cannot connect the multifaceted estimation needs and the many audiences together into a single vision.
3. In a networked ERP context, the existing estimation models do not necessarily yield accurate results, as staff-members from each of the network partners can incorporate their own intuition (and biases) into the estimate.

4. Effort and schedule seem to be driven not by the requirements that are known to the network and those that the network partners know are unknown, but by those that the partners do not know are unknown (Armour 2004). For example, crucial cross-organizational requirements about coordination and integration are easily ignored throughout the project, and the partners are unaware of this fact and do not account for it in their cost estimation activity.

This situation could be improved if flexible and integrated frameworks for size-effort-cost modeling were available to requirements engineers to help solve the following problems:

- allow estimation of implementation costs by linking multiple measures of size and ERP context factors together from a networked organization's perspective,
- deal with the inherent uncertainty of ERP cost drivers,
- allow application of real-option analysis (Amram and Kulatilaka 2000),
- allow integration of investment analysis of different but related projects, and
- deal with the incompatibility of data from different companies by establishing a measurement standard that each of the partners in the network can share and agree upon.

Such an integrated approach to ERP project cost modeling is especially important at the current stage for two reasons. First, the exposure of the partners to operational failure risks due to poor cross-organizational ERP implementation is increased because each of them may well participate in multiple virtual organizations, and, also, may provide capabilities across many value networks. Second, networked businesses increasingly rely on outside parties for the deployment and operation of cross-organizational ERP systems. One could argue that cost and benefits estimation as part of ERP requirements engineering is the *only* RE job a networked organization has left in an environment of total outsourcing.

## 4 Framing the discussion of ERP size and cost modeling

This section defines what is important to consider in building cost models for cross-organizational ERP. As an introduction to the section, we first present a high-level summary of the cost estimation problem space in Fig. 1 (which we derived as an abstraction of what follows in this section).

According to Fig. 1, developing an approach to ERP cost modeling means finding a mapping between (i) what problems we observe in the literature and in practice, and (ii) what possible solutions can be used to confront the observed problems. If we first collect facts and observations about the current difficulties with cross-organizational ERP sizing and cost

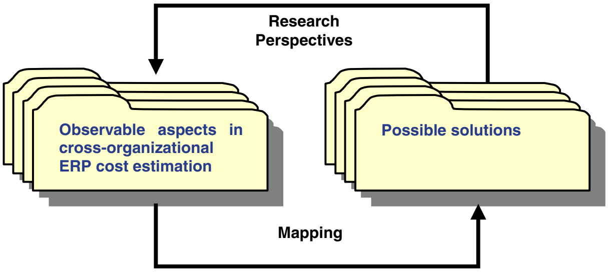
Research Perspectives
Observable aspects in cross-organizational ERP cost estimation
Possible solutions
Mapping

**Fig. 1** Cross-organizational ERP cost estimation: a high-level view

estimation, and, second, assess what the existing possible solutions offer to resolve those difficulties, then we should be able to establish such a mapping. Clearly, for a mapping to reflect the multifaceted nature of cross-organizational ERP projects, it needs to be approached from multiple research perspectives.

### 4.1 What makes up a cost estimation model for cross-organizational ERP projects?

A significant obstacle in the way of an integrated approach to modeling size/cost relationships in ERP RE is definitions, which must specify clearly what the appropriate level of analysis of ERP project aspects is and what these aspects are. The perspective we present in this paper suggests that there is a need for an integrated modeling of ERP project aspects at both the organizational and cross-organizational levels of analysis. At the organizational level, the knowledge and skills of business users and the amount of change in their working practices due to the new ERP systems are as important to understanding costs as is the customization of the package itself. At the cross-organizational level, the architecture decisions, such as where the place of the ERP system in the value network, e.g. whether it is placed on the buyer side, on the seller side, or in the middle of the network, can be of paramount importance to estimators, as such decisions determine which cost and risk factors to consider and what opportunities exist for cascading operational failure.

Typically, the prediction of the effort to be expended in a project is the most sought-after variable (Abran et al. 2001; Boehm et al. 2000a; Dolado 2001; Kitchenham and Mendes 2004; Molokken and Jorgensen 2003; Myrtveit et al. 2005; Stamelos et al. 2003; Stensrud 2001), that is, the size of the solution is related to the person-months employed to implement it. Therefore, the term “cost” is frequently used to represent the *human effort* required for solution implementation. When the relationship between cost/human effort and one or more variables, e.g. size, is expressed as an equation, it is called a cost function. The resulting equation may range from linear models to quadratic, Cobb-Douglas, and translog models. It is typically adjusted for a variety of intra-organizational factors, such as complexity of the solution and the use of standards and tools. Such an equation is also supposed to be calibrated to fit the actual data points of the organization’s projects. Although research in the past 40 years has introduced many equations, and most of them were good enough for the purposes for which they were created, none has produced sufficient evidence to be considered the definitive cost function for software (Hu 1997).

Furthermore, the business requirements document—a business artifact—is often considered to be the appropriate level of analysis for studying ERP size (Daneva 1999a, b). Certainly, this choice is related to important technical issues, such as the way applications would share data with the ERP solution or the vulnerability of much customized code which must be addressed in any complete estimation model. Different definitions of size can be found in the IS literature. According to one common definition, size is measured in lines of code (Cote et al. 1998). A second wave of definitions considers size to be measured in function points (Abran et al. 2001; Netherlands Software Measurement Association 1997; Symonds 1991), object points, future points (Jones 1997), case points (Mohagheghi et al. 2005), or class points (Costagliola and Tortora 2005) to determine the amount of functionality that would be embedded in the solution. A third characterization of size, specifically for ERP solutions, is provided by Accenture’s Global SAP Service and includes counts of the following ten aspects: users, sites, business units, software interfaces, EDI interfaces, data conversion, custom-developed reports, modified screens, and ERP modules (Stensrud and Myrtveit 1998; Stensrud and Myrtveit 2003). A proposal for an ERP size definition is also currently being developed by SAP Information Systems (2004) and includes counts of: (i) the number of clients, and (ii) the number of transactions included in an SAP solution. While the first two types of size definition (Abran et al. 2001; Costagliola and Tortora 2005; Cote et al. 1998; Jones 1997; Mohagheghi et al. 2005; Netherlands Software Measurement Association 1997; Symonds 1991) suggest that effort is related to a single size measure, Accenture (Stensrud and Myrtveit 1998; Stensrud and Myrtveit 2003) and SAP Information Systems (2004) relate effort to several different size measures.

Building on the above authors' efforts and on our own experience (Daneva 1999a, b), we will use an approach to size/effort modeling in the research proposed below in which effort is related to a set of different size measures. That is, in order to size cross-organizational ERP solutions, we will devise a size counting process: (i) based on the International Function Points User Group (IFPUG) FPA standard (Garmus and Herron 2001); and (ii) enriched with the Accenture measures of size (Stensrud and Myrtveit 1998; Stensrud and Myrtveit 2003). Our decision to use an FP-based measure is pragmatic and motivated by the following: (i) it is the first method (Abran et al. 2001) to conform to ISO/IEC14143-1, which specifies a set of generic mandatory requirements for a method to be referred to as a functional size measurement method; (ii) it is the only sizing approach applicable early in a project, e.g. in the RE process (Dekkers 2003; Garmus and Herron 2001; Horgan et al. 1998); (iii) it has seen some use in ERP projects (Daneva 1999a, b; Morrison and Seaver 2007; O'Neil 2002); (iv) it has proven its worth in building cost estimation models (Jones 1997); and (v) a considerable number of projects measured in FP are becoming available, for example through the International Software Benchmarking Standards Group (ISBSG) (Stamelos et al. 2003), providing options for performing statistical analysis on them. Furthermore, our decision to complement the FP counts with other requirements-based measures of size (e.g. the ones in the Accenture proposal (Stensrud and Myrtveit 1998)) is motivated by the need to achieve a multidimensional functional size measurement model (Stensrud 2001). In Sect. 5, we will see that at least four research perspectives could, and should, be taken into account in order to find the elements that might be good candidates for inclusion in a multidimensional size measure. In this section, we will discuss how the application of the IFPUG FP standard (Garmus and Herron 2001) to ERP projects is different from its application in custom software projects and other types of business information system projects (which might deploy new technologies, e.g. Internet, Intranet).

As per the IFPUG standard (Garmus and Herron 2001), a FP-based counting method involves counting and weighting the data and process logical components of a system with an adjustment thrown in for some aspects of the environment in which the system will operate. However, the IFPUG community warns that the standard FP counting manual fails to provide sufficient detail with respect to new and emerging technologies, for example, object-oriented systems (Abrahao and Poels 2007; Jones 1997), Internet and Intranet applications, and cross-organizational ERP, among others. For the IFPUG counting process to be applied to projects implementing new technologies, the standard definitions of the IFPUG data and process counting components must be rethought in the contexts of these projects and adapted to them (Abrahao and Poels 2007; Daneva 1999b, Jones 1997). For example, researchers in the field of functional size measurement have already advanced in terms of adapting FP counting definitions and validating their application in object-oriented development projects (Abrahao and Poels 2007). What is unique about the application of the IFPUG standard to ERP is that a set of rules must be established for mapping the data and process counting components of the IFPUG standard onto the components of the data and process requirements models in the ERP business requirements document (Daneva 1999b). These rules stipulate: (i) how to identify the boundary of the ERP project to be counted; (ii) how to identify the components to be counted within the boundary of the project; (iii) how to classify these components; and (iv) how to assess the complexity of each component that is counted. Attempts to define and use such mapping rules have been made since 1998 by the first author (Daneva 1999b), as well as by other practitioners in ERP size estimation (Morrison and Seaver 2007; O'Neil 2002; Vogelesang 2006). However, their combined experience indicates that this mapping exercise is far from straightforward. ERP size analysts encounter three major difficulties, for which, to the best of the authors' knowledge at the time of writing this paper, no solutions have yet been proposed. The first issue refers to the definition of the counting boundary for an ERP project. It stems from the fact that an ERP project does not implement merely application components, but cross-organizational business processes based on these components. Because cross-organizational processes are the fundamental building blocks of the ERP system (Holland et al. 2005), the boundary of an ERP project is drawn around these processes. Custom software projects differ in a major respect, which is that the boundary is drawn around the application itself. In the early project stages of ERP projects, Vogelesang (Vogelesang 2006) found that a lack of knowledge about how the ERP adopter's business processes relate to ERP application functions and how pieces of data interact means that an ERP project cannot be estimated like a custom software project by means of some "early FP counting method" (Jones 1997).

The second issue is that ERP packages use vendor-provided methodologies for engineering the business requirements and the solution architecture, which implies that different requirement modeling notations and requirement architecture frameworks are used for each package (Kelly and Holland 2002; Rolland and Prakash 2000). This calls for mapping rules to establish semantic equivalence between the modeling components of each of the package-specific requirement modeling languages and the counting elements of the IFPUG standard. Only then can size analysts be sure that they are referring to, and acknowledging, the same elements when applying a FP counting process based on the IFPUG standard. A fuller description of some of the mapping rules devised for the ERP project context of one specific package, namely SAP appears in (Daneva 1999b).

The third major issue arises from the fact that an ERP system consists of multiple configurations which let the system acquire new quality attributes as result of these configurations, each of which is a set of changes to (or so-called customizations of) the original functionality of the ERP package. These changes may well be of different types (Brehm et al. 2001; Daneva 1999b, Vogelesang 2006), each varying in terms of: (i) level of reuse (Daneva 1999b, Vogelesang 2006), (ii) the extent to which it affects other module configurations, and (iii) customization risk. As Brehm et al. have pointed out (Brehm et al. 2001), at least ten types of changes could be applied to an ERP system. Applying the IFPUG standard rules for counting the changed functionality in an ERP project is difficult, because the IFPUG counting method treats all changes equally and cannot capture categorical information about the types of complexity inherent in the various types of changes that can be applied to the package.

### 4.2 What kinds of cost estimation models are there?

Schools of thought in software economics (Boehm et al. 2000a; Boehm and Sullivan 2000) have been developing models of size/effort relationships that fall into three categories: expert judgments, algorithmic estimation, and estimation by analogy. The first class of models relies on the accumulated experience of a team of experts. Typically, judgments about effort and schedule are offered by ERP vendors based on their implementation experiences. For example, the Accelerated SAP process for rapid SAP implementation, marketed by SAP and their consulting partners, suggests a six-month time frame for project completion (SAP AG 1997). The second class of models relies on the application of a set of mathematical formulas which, typically, rests on statistical data analysis. This class is further structured in the form of parametric and non-parametric regression models, depending on the assumptions made about the form of the cost function. The third class of models compares a new project with a set of similar past projects (i.e. projects with known context characteristics, size, effort and duration) by using case-based reasoning techniques (Shepperd and Schofield 1997). To the best of our knowledge, however, the suitability of these approaches to cross-organizational ERP projects has not been empirically investigated. A few authors (Stamelos et al. 2003; Stensrud 2001; Stensrud and Myrtveit 2003; Daneva 2007) have investigated the applicability of these groups of methods to ERP projects. What Stensrud and Myrtveit found works well in the ERP context is to first use parametric regression models as an effort prediction system, and then to complement this with an analogy-based method for the purpose of exploratory data analysis (Stensrud 2001). Moreover, these authors built a productivity model (Stensrud and Myrtveit 2003) based on Data Envelopment Analysis and used the Accenture project datasets. They assessed the productivity of ERP implementations relative to the most productive project with similar output values (i.e. size measures). However, the selection of the most productive ERP project as a benchmark rested on the assumption rooted in economics according to which the most productive project is the one that deploys the best technology. This may not necessarily be the case for cross-organizational ERP projects, where the most productive project can be the easiest one, or the one deploying an RE process, standards and tools with which the RE team is familiar and which fit with the culture of the network partner who would architecturally host the ERP system.

Furthermore, the first author of this paper (Daneva) experimented with an approach (2007) combining COCOMO II (Boehm et al. 2000b), a Monte Carlo simulation, and a portfolio management concept. Unlike Stensrud and Myrtveit (Stensrud and Myrtveit 2003), this author treated an ERP implementation as a portfolio of interdependent ERP projects. In this approach (Daneva 2007), the strengths of the three models were complementary: COCOMO II was used to capture some ERP-specific cost drivers, the Monte Carlo simulation was the vehicle selected to handle the degree of uncertainty of these cost drivers, and an effort-and-deadline-probability-based portfolio management model was used to quantify the chance of success with the proposed interdependent deadlines for a set of related ERP projects. Daneva wanted to see if such a combined approach would improve the ability of effort analysts to make sense of the uncertainties in ERP projects that arise due to volatile requirements. The experiment indicated that there was an advantage to bundling ERP projects as a portfolio over managing them separately. Specifically, the results suggested that, when managed as a portfolio, highly uncertain ERP projects have a greater chance of succeeding under time and effort constraints.

Drawing on the early assessments of the authors of (Daneva 2007; Stamelos et al. 2003; Stensrud 2001; Stensrud and Myrtveit 2003) and on a wide range of literature, we will, in the research proposed in Sect. 5, experiment with an integrated conceptual framework deploying a variety of models capable of assisting estimators in reflecting on why evaluations provided by one model might differ significantly from evaluations provided by another. Good candidates for inclusion in our framework are models which respond to the following requirements:

- the flexibility to choose among a variety of predefined sets of inputs (Stensrud 2001);
- the ability to provide adequate results when the amount of related empirical data is initially minimal (Abts et al. 2000) and datasets are incomplete (Myrtveit and Stensrud 2001) or include extreme cases (e.g. missing data elements, outliers);
- the ability to interpret predictor variables, e.g. the method should rest on a valid assumption about the form of the size/cost relationship.

### 4.3 Research case for an integrated approach to size and cost modeling

Although IS researchers have indicated the importance of investigating size/cost relationships in ERP settings, their published works (Daneva 2007; Morrison and Seaver 2007; Seaver 2005; Stensrud and Myrtveit 1998; Stensrud 2001; Stensrud and Myrtveit 2003; Vogelesang 2006) cite business needs in isolation and offer "solution islands" requiring theoretical integration in a number of areas. Our review of the literature and practice indicates that it is possible to find arguments which, when assembled, call for an integrated approach to sizing and estimation efforts that ERP adopters should consider in cross-organizational ERP projects. The purpose of such an approach is not to argue that one size/cost relationship model is superior to another, but rather that a single model is not sufficient for all the concerns that senior management might have about an ERP project.

## 5 Research perspectives

We propose that the topic of size/cost modeling be approached in an integrated fashion from at least four perspectives: (i) the business change dimension of any cross-organizational ERP project; (ii) its knowledge management dimension; (iii) the diversity of the customization options to be handled; and (iv) the dynamics of the size and effort factors to be modeled. We decided to consider these perspectives because authors in the ERP, RE and cost estimation fields (Armour 2002; Brehm et al. 2001; Costello and Liu 1995; Davenport 2000; Ebert and De Neve 2001; Herbsleb et al. 2005; Ko et al. 2005; Ramasubbu et al. 2005; SAP SI 2004; Stamelos 2001; Stensrud and Myrtveit 2003; Summer 2000) found them useful in understanding the multifaceted nature of the cross-organizational ERP projects and the determinants of how project resources are likely to be consumed.

### 5.1 The business change dimension

While the majority of existing cost estimation models capture the functionality aspects of a project fairly well, they fall far short of capturing the amount of business change (or business transformation) set in motion by the ERP solution. In our research, we make a conscious effort to fill a noticeable gap in the current discussion on ERP cost functions. Specifically, this means that our integrated approach will: (i) account for the possible changes in terms of coordination processes, business processes, and data handling procedures, and (ii) express them as complexity factors that are to be linked to the size/cost relationship (Daneva 2007; Vogelesang 2006).

### 5.2 The knowledge management dimension

Our second perspective involves a view of ERP projects as knowledge transfer initiatives. Previously published empirical studies by (Davenport 2000; Ko et al. 2005) indicate that all ERP adopters have to acquire new ERP implementation knowledge, operational knowledge, and maintenance and training knowledge, and that ERP-related knowledge acquisition adds up to projects which are longer and more expensive. In cross-organizational settings, each partner in the network has to deal with their own knowledge barriers and should consider their own speed and depth of learning. As this is an important cost parameter (Ko et al. 2005), an integrated conceptual framework should be capable of expressing these project aspects in the form of factors of the ERP cost model.

### 5.3 The customization options

This perspective reflects the fact that ERP customization is the riskiest endeavor a company can undertake, and rests on the thorough study by Brehm, Markus and Tanis (Brehm et al. 2001) on customization options in ERP systems. The authors characterize nine ERP tailoring types which may vary widely in terms of how extensively a particular type is applied, how well it is applied, how much it changes either the data stored or the data structures, how much it has to be changed when the ERP solution is upgraded, and how it depends on other tailoring types. The authors discuss the customization option dimension as a major source of risk, and point out that it is indeed the most expensive portion of an ERP project, but they say nothing about how to incorporate assessments of customization requirements in size/cost relationship models.

In our research, we augment the Brehm, Markus and Tanis approach (Brehm et al. 2001) by providing a vehicle through which to understand and estimate the costs associated with the various modes of customization. For example, we are interested in knowing what the costs are of constructing a single/atomic customization requirement.

### 5.4 The dynamics of size and effort factors

The last dimension in our planned approach to cost modeling is that of the dynamics of the factors characterizing a cross-organization ERP project and its context. In networked settings, it is not possible to know whether the early or the final products are inherently “complete”. Therefore, metrics on products, resources, and processes are to be seen as moving targets. The fast-changing nature of networked businesses and of their ERP requirements makes it very difficult to develop parametric models which yield high accuracy. In cross-organizational ERP settings, legacy applications developed at different times are inherited and new global software processes (Ebert and De Neve 2001; Herbsleb et al. 2005; Ramasubbu et al. 2005) are adopted. As a result, the nature of the ERP solutions and the resources used to create them are evolving and the requirements that serve as input to the cost estimation model become highly volatile (Costello and Liu 1995), as do the estimates that depend on them. If an integrated framework is to address these difficulties, it should offer dynamics-oriented functionality-based estimation models which acknowledge changes in the ERP adopter’s business context.

### 5.5 Translating needs into research questions

This section discusses what we think makes up relevant research towards an integrated framework for building size/effort relationship models for cross-organizational ERP projects. We have attempted to derive research questions (Table 1) from the business and research needs we have identified in previous sections. We arrived at three major research questions that we have broken them down into sub-questions based on the specific needs to which they respond. Each of our research questions and sub-questions is designed to address the need for multidimensionality in our integrated approach to cross-organizational ERP cost estimation. Therefore, they cannot be specific to one dimension only.

## 5.6 Operationalization

This section describes our experience with operationalizing parts of the research agenda (Table 1) by using a set of research project proposals. Clearly, we think that it will be possible in the future for other researchers from the IS and RE community to come up with a number of alternative operationalizations. So, in no way can we claim that the experience we share in this section reflects the only path that researchers can adopt. Indeed, only by using a variety of operationalizations, can we as a community advance in collecting the empirical evidence (Hevner et al. 2004; Seaman 2001) that will allow us to encourage the inclusion of new research perspectives in the integrated approach to ERP size/effort estimation.

Below, we summarize how the proposed research agenda (Table 1) was used by researchers at the University of Twente, The Netherlands, and ten ERP cost estimation practitioners from ten member companies of the Dutch Software Measurement Association (NESMA, http://www.nesma.nl), who formed a consortium for preparing a set of research project proposals. These research proposals operationalized the agenda in terms of (i) specific actionable plans, (ii) research methods to be deployed, and (iii) specific research study contexts. The process used for this purpose followed the guidelines of design-science research (Hevner et al. 2004) whenever applicable. Each research proposal (i) was a four-year research effort for one PhD researcher and a number of business partners among the ten NESMA companies, (ii) was founded on a detailed research plan that further elaborated the research questions in the specific organizational contexts of the participating business partners, and (iii) responded to one or two of the needs documented in Table 1.

The budgets of two research projects have already been approved, and they have been launched at the University of Twente and in the ten companies. The overall aim of the first project, entitled Cost Estimation of Cross-organizational ERP systems (COSMOS), is to develop a method for quantitative ERP requirements engineering (RE) which integrates software product and process metrics. Its work packages are about investigating, comparing, and improving existing approaches to the problem of functional size measurement for cross-organizational ERP implementations. It is mainly linked to needs A1, A2, B1 and B2 presented in the second column of Table 1. The second project, Quality-Driven Requirements Engineering and Architectural Design (QuadREAD) has as one of its objectives the development of approaches to using knowledge about project cost drivers, in support of decision-making processes at the requirements engineering stage. A major part of its scope is the research on interval-based size/effort models that account for volatile requirements. This project focuses on research questions primarily addressing needs B3 and B4 presented in the second column of Table 1. Both projects deploy a process for industry-relevant research (Hevner et al. 2004) including five main clusters of activities (Gorschek and Wohlin 2004), as shown in Fig. 2:

**Table 1** List of relevant research questions

| Research question | Needs | Research sub-questions |
|---|---|---|
| A. What are the essential cost drivers to be considered for inclusion in cost estimation models for interdependent ERP projects in cross-organizational settings? | A.1. To identify ERP-specific cost drivers in cost models | A.1.1. What are the typical cost drivers that impact cost estimation in cross-organizational ERP projects? |
| | | A.1.2. What are the variables worth including in a model? |
| | | A.1.3. How do we know we included the right variables and the most important ones? |
| | A.2. To estimate the costs of interdependent ERP projects | A.2.1. How can a cost estimation model be extended to meet the estimation requirements of multiple projects? |
| | | A.2.2. Are there existing techniques that may be applicable to multiple cross-organizational ERP projects? |
| | | A.2.3. What is the value of using parametric versus non-parametric models in cross-organizational ERP projects? |
| | A.3. To establish a basis for comparison of ERP cost functions | A.3.1. How can cost estimation models be compared? |
| | | A.3.2. How do we know if the behavior of a model is reasonable outside the range of the pre-existing datasets? |
| | | A.3.3. What are the assumptions that ERP-adopting organizations use or can use to define size/cost equations? |
| B. What is a good functional size measure for cross-organizational ERP projects? | B.1. To define a multidimensional size measure to provide adequate assistance for ERP effort estimation | B.1.1. How can size be regarded as having multiple dimensions, instead of trying to capture it in a singe number? |
| | | B.1.2. What is a valid value adjustment factor for a FP-based measurement? |
| | B.2. To revise/extend the existing vehicles to capture categorical information about the project, in FP-based models | B.2.1 What characterizes the context of a cross-organizational ERP project? |
| | | B.2.2. How can characterizing information be captured in a size model? |
| | B.3. To respond to the fast changing nature of cross-organizational ERP integration | B.3.1. What functionality-based estimation models suit the dynamics of the cross-organizational ERP project context? |
| | | B.3.2. How can volatile cost drivers be accounted for in early estimates? |

**Table 1** continued

| Research question | Needs | Research sub-questions |
|---|---|---|
| | | B.3.3. How can volatile requirements be dealt with when used as input to ERP effort/cost estimation model? |
| | B.4. To cope with uncertainty due to ERP requirement errors | B.4.1.What are the common errors in ERP requirements? |
| | | B.4.2. How can the effects of ERP requirement errors be mitigated? |
| | | B.4.3. What interval-based models make sense? |
| C. How can we judge the fit of cost models in real-life cross-organizational ERP settings? | C.1. To know the goodness of fit of an ERP estimation model | C.1.1. What makes up an adequate criterion of goodness-of-fit in cross-organizational ERP settings? |
| | | C.1.2. How do we know whether a size/cost relationship is stable? |
| | | C.1.3. How can the accuracy of ERP models be improved? |
| | C.2. To explain why the results from individual studies differ | C.2.1. How can inconsistent results from previous studies be reconciled? |
| | | C.2.2. How can meta-analysis (Mair and Shepperd 2004) be applied or extended to cross-organizational ERP settings? |

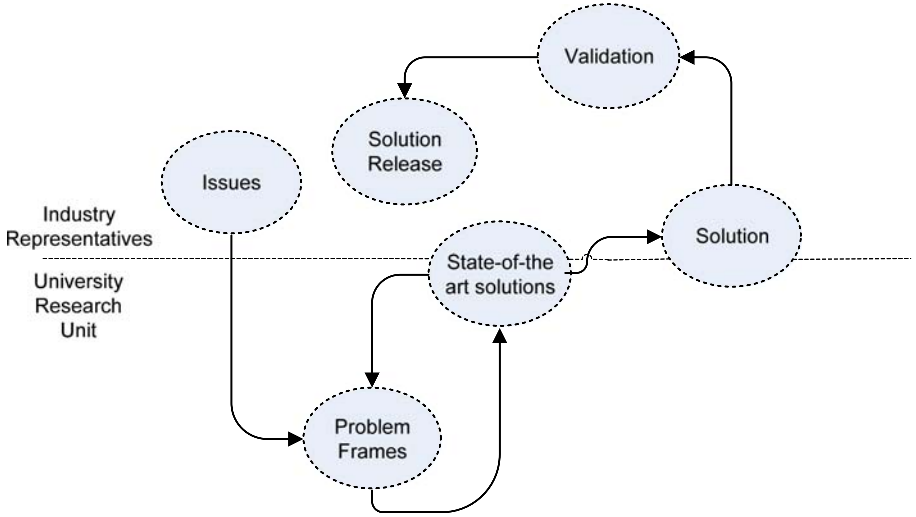

**Fig. 2** The research process founded on (Gorschek and Wohlin 2004)

### *5.6.1 Issues*

We suggest that observable and identifiable issues arising at our partners' ERP project sites constitute the trigger for formulating research hypotheses and setting up research questions.

### *5.6.2 Problem frames and state-of-the-art solutions cycle*

Once researchers document, categorize, and catalogue the issues, the project team defines the search space for existing solution options from both the literature and practice. The expected outcome of this activity is a catalogue of existing solutions and an understanding of why they fall short of solving the problems identified.

### *5.6.3 Solution*

Capitalizing on the knowledge gained, the project team plans to use (i) the companies' ability to innovate and (ii) the researchers' expertise in designing a solution for quantitative ERP RE which includes functional size measurement and effort estimation at its core.

### *5.6.4 Validation*

The elements of the solution are subjected to validation in industrial settings; for example, assessments of internal, external, constructive and concluding validity concerns (Yin 2002) related to the solution design. Incremental changes will be implemented in the solution design to address issues revealed in the validation case studies.

### *5.6.5 Solution release cycle*

We suggest that validated parts of the solution be made available to partner companies to incrementally implement them in real-life settings. Based on some lessons we expect to learn in this process, enhancements to the solution approach are to be introduced.

The key actors in this research process are the following:

- The principle investigating research unit, namely the University of Twente, and three collaborating parties, the University of Valencia in Spain, the Université du Québec, Montreal and Concordia University in Canada.
- Industry representatives from both the ERP consulting sector and the Dutch community of ERP clients.

The business partners will be involved in both the COSMOS and the QuadREAD projects in a variety of ways:

- as participants in action research and case studies to (i) make sure that the definitions of our measures are based on their project goals, and (ii) define procedures for packaging, cataloguing, publishing, and reporting measurement data.
- all partners will have access to the research results and will receive researchers' support in using the results to improve current practice,
- master's degree student's projects on size/effort estimation will be hosted by business partners.

Following the process in Fig. 2 means that the project research teams commit to five work principles (Gorschek and Wohlin 2004; Yin 2002):

1. Industry settings are used as a laboratory.
2. In each iteration, short case studies are included to ensure better analytical results.
3. Short feedback loops are part of the interactions between industry partners and researchers to ensure the relevancy of the researched subjects.
4. Whenever a measurement or prediction model is to be developed, it will include: (i) identification of key variables affecting the product or process attribute of interest; (ii) the formulation of a theory on how the variables relate to each other and to the attribute of interest; and (iii) the construction of a model that reflects this relation.
5. Refinements of our solution approach and improvements to it are to be performed incrementally on a step-by-step basis.

## 6 Conclusions

This paper surveyed the state-of-the-art literature and practice in modeling and evaluating size and cost factors in cross-organizational ERP projects. It identified shortfalls in existing knowledge and made a proposal for what we think is a viable approach to investigating size/cost relationships in networked settings. Networked businesses are beginning to see the value of an integrated approach to building size and cost estimation models for their ERP projects, and so the time is right for the RE field to begin developing an integrated approach to modeling relationships among size, effort, and other cost aspects, and, ultimately, using such models for managing cross-organizational ERP customization-related risks. Not only will such an approach be useful to businesses in their attempts to gain deeper insights into the size and cost aspects of their ERP projects, it will also help bring together a large part of the ERP literature under a common conceptual umbrella. By viewing cross-organizational ERP solutions in the light of business change, knowledge transfer, diversity of customization choices and dynamics along with business case analysis as part of the ERP RE process, management of size, efforts, and costs become the focus of attention. By viewing operational system failure, security breaches, and competitive threats as different types of the unitary phenomenon of ERP RE risk, it becomes possible for a business network to make intelligent customization tradeoff decisions about their cross-organizational ERP solution.

## References

Abrahao, S., & Poels, G. (2007). Experimental evaluation of an object-oriented function point measurement procedure. *Information and Software Technology, 49*(4), 366–380.

Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., & Symonds, C. (2001). *COSMIC-FFP measurement manual, version 2.0, Software Engineering Management Research Laboratory*. Montreal, Canada: Université du Quebéc à Montréal.

Abts, C., Boehm, B., & Clark, E. (2000). *COCOTS: A COTS software integration lifecycle cost model—model overview and preliminary data collection findings, technical report USC-CSE-2000-501, USC center for software engineering*. Los Angeles: USA University of Southern California.

Amram, M., & Kulatilaka N. (2000). *Real options: Managing strategic investment in an uncertain world*. Boston: Harvard Business School Press.

Armour, P. (2002). Ten unmyths of project estimation. *Communications of the ACM, 45*(11), 15–18.

Armour, P. (2004). Real work, necessary friction, optional chaos. *Communications of the ACM, 47*(6), 15–18.

Arnesen, S., & Thompson, J. (2005). How to budget for enterprise software. *Strategic Finance, January*, 43–47.

Bardhan, I., Bafgci, S., & Sougstad R. (2004). A real options approach for prioritization of a portfolio of information technology projects: A Case study of a utility company. *Proceedings of the 37th Hawaii International Conference on Systems Sciences*, IEEE CS Press, Los Alamitos, p. 80215b.

Benaroch, M., & Kauffman, R. J. (1999). A case for using real options pricing analysis to evaluate information technology investments. *Information Systems Research, 10*(1), 70–86.

Benchmarking Partners & IBM. (2000). Beyond ERP collaboration and value networks, ERP and e-business transformation.

Boehm, B., & Sullivan, K. J. (2000). Software economics: A roadmap. *8th International Conference on Software Engineering, Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland, pp. 311–343.

Boehm, B., Abts, C., & Chulani, S. (2000a). Software development cost estimation approaches—a survey. *Annals of Software Engineering, 10*, 177–205.

Boehm, B., Horowitz, E., Madachy, R., Reifer, D., Clark, B. K., Steece, B., Brown, A. W., Chulani, S., & Abts, C. (2000b). *Software cost estimation with Cocomo II*. Upper Saddle River, NJ: Prentice Hall PTR.

Brehm, L., Heinzl, A., & Markus, M. L. (2001). Tailoring ERP systems: A spectrum of choices and their implications. *Proceedings of the 34th Hawaii International Congress on System Sciences*, IEEE Computer Society Press, Los Alamitos.

Collett, S. (1999). Hershey earnings drop as new warehouse, order systems falter. *Computer world*, October 27.

Costagliola, G., & Tortora, G. (2005). Class points: An approach for the size estimation of object-oriented systems. *IEEE Transactions on Software Engineering, 31*(1), 999–1013.

Costello, R., & Liu, D.-B. (1995). Metrics for requirements engineering. *Journal of Systems Software, 29*(1), 52–74.

Cote, V., Bourque, P., Oligny, S., & Rivard, N. (1998). Software metrics: An overview of resent results. *Journal of Systems and Software, 8*(2), 121–131.

Daneva, M. (1999a). Function point counts derived from SAP business scenario requirements. In R. Dumke & A. Abran (Eds.), *Software measurement: Current trends in research and practice*. Wiesbaden: Gabler Verlag.

Daneva, M. (1999b). Measuring reuse of SAP requirements: A model-based approach. *Proceedings of the 5th International Symposium on Software Reusability (SSR'99)*, Los Angeles, CA, USA, ACM Press, pp. 141–150.

Daneva, M. (2007). Approaching the ERP project cost estimation problem: An experiment. *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, IEEE Computer Society Press, p. 500.

Daneva, M., & Wieringa, R. J. (2006a). A requirements engineering framework for cross-organizational ERP Systems. *Requirements Engineering Journal, 11*, 194–204.

Daneva, M., & Wieringa, R. J. (2006b). A coordination complexity model to support requirements engineering in cross-organizational ERP. *Proceedings of the International Conference on Requirements Engineering*, IEEE Computer Society Press, Minneapolis, USA.

Davenport, T. (2000). *Mission critical: Realizing the promise of enterprise systems*, Boston, MA: HBS Press.

Dekkers, T. (2003). Beter meten met Cffp, Informatie (Dutch IT magazine).

Dolado, J. J. (2001). On the problems of the software cost function. *Information and Software Technology, 43*, 61–72.

Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE Software, 18*(2), 62–69.

Epstein, M. J., & Rejic, A. (2005). Measuring the payoffs of IT investments, CMA Management.

Eschinger, C. (2004). Forecast: ERP software worldwide 2004–2009, Gartner Report G00127420, Stamford, CT, USA.

Garmus, D., & Herron, D. (2001). *Function point analysis: Measurement practices for successful software projects*, Boston, MA: Addison Wesley.

Gorschek, T., & Wohlin, C. (2004). Packaging software process improvement issues: A method and a case study. *Software Practice and Experience, 34*(14), 1311–1344.

Herbsleb, J. D., Paulish, D., & Bass, M. (2005). Global software development at siemens: Experience from nine projects. *Proceedings of International Conference on Software Engineering*, St. Louis, Missouri, pp. 524–533.

Carr, D. (2002). Hershey's Sweet Victory, Baseline Magazine, The Project Management Center, December 16, 2002.

Hevner, A. R., Salvatore, T., March, J. P., & Sudha, R. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.

Holland, C. P., Shaw, D. R., & Kawalek, P. (2005). BP's multi-enterprise asset management system. *Information and Software Technology, 47*(15), 999–1007.

Horgan G., Khaddaj, S., & Forte, P. (1998). Construction of an FPA-type metric for early lifecycle estimation. *Information and Software Technology, 40*(8), 409–415.

Hu, Q. (1997). Evaluating alternative software production functions. *IEEE Transactions on Software Engineering, 23*(6), 379–387.

Jones, C. (1997). *Applied software measurement*. New York: McGraw-Hill.

Jones, C. (2004). Software project management practices: Failure versus success, *CrossTalk—the Journal of Defense Software Engineering, October*, pp. 5–9.

Jones, C. (2005). Software engineering: The state of the art in 2005, Software productivity research, Burlington, Massachusetts. http://www.spr.com.

Kelly, S., & Holland, C. P. (2002). The ERP systems development approach to achieving an adaptive enterprise: The impact of enterprise process modelling tools. In P. Henderson (Ed.), *Systems engineering for business process change* (pp. 241–252). London: Springer.

Kitchenham, B., & Mendes, E. (2004). Software productivity measurement using multiple size measures. *IEEE Transactions on Software Engineering, 30*(12), 1023–1035.

Kitchenham, B. A., Pickard, L., Linkman, S., & Jones, P. (2003). Modelling software bidding risks. *IEEE Transactions on Software Engineering, 29*(6), 542–554.

Ko, D.-G., Kirsch, L., & King, W. (2005). Antecedents of knowledge transfer from consultants to clients in enterprise system implementations. *MIS Quarterly, 29*(1), 59–85.

Koch, C. (2002). Hershey's bittersweet lesson. *CIO Magazine*.

Mair, C., & Shepperd, M. (2004). Making software cost data available for meta-analysis. *Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE, 2004)*. Scotland, UK: Edinburgh, pp. 37–45.

Molokken, K., & Jorgensen, M. (2003). A review of software surveys on software effort estimation. *Proceedings of International Symposium on Empirical Software Engineering*. IEEE Computer Society Press, pp. 223–230.

Mohagheghi, P., Anda, B., & Conradi, R. (2005). Effort estimation of use cases for incremental large-scale software development. *Proceedings of International Conference on Software Engineering*, St. Louis, Missouri, pp. 303–311.

Morrison, N., & Seaver, D. (2007). Utilizing functional size on enhancement projects, PSM Users Group, PRICE Systems Research Inc., Golden CO, July 23–27, http://www.pricesystems.com.

Myrtveit, I., & Stensrud, E. (2001). Analysing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods. *IEEE Transactions on Software Engineering, 27*(11), 999–1013.

Myrtveit, I., Stensrud, E., & Shepperd, M. (2005). Reliability and validity in comparative studies of software prediction models. *IEEE Transactions on Software Engineering, 31*(5), 380–391.

Netherlands Software Measurement Association. (1997). *Definitions and counting guidelines for the application of function points, Version 2.0*. The Netherlands: NESMA.

O'Neil, M. (2002). Measuring SAP or other ERP applications. *Proceedings of the Annual Conference of the Japanese Function Points User Group*.

Putnam, L., & Myers, W. (1992). *Measures for excellence: Reliable software on time within budget*. Yourdon Press Computing Series.

Ramasubbu, N., Krishnan, M. S., & Kompalli P. (2005). Leveraging global resources: A process maturity framework for managing distributed development. IEEE Software, May/June, pp. 80–86.

Rohde, J. (2005). *Conquering the challenge of global SAP implementations: SAP roll-out strategy & best practices*. CIBER Inc, Greenwood Village, CO.

Rolland, C., & Prakash, N. (2000). Bridging the gap between organizational needs and ERP functionality. *Requirements Engineering Journal, 5*, 180–193.

SAP AG. (1997). ASAP methodology for rapid R/3 implementation: User manual, Waldorf, Germany.

SAP SI. (2004). Total cost of ownership in the ERP environment, SAP Systems Integration AG. Dresden, Germany.

Sarrat, C. (2002). Hershey stays the course with mySAP.com Upgrade, Computer Weekly, September 2.

Seaman, C. (1999). Qualitative methods in empirical software engineering. *IEEE Transactions on Software Engineering, 25*(4), 557–572.

Seaver, D. P. (2005). Estimating ERP systems. *Annual European ERP Symposium*, Paris, France.

Shepperd, M., & Schofield, C. (1997). Estimating software project effort using analogies. *IEEE Transactions on Software Engineering, 23*(11), 736–743.

Stamelos, I. (2001). Managing uncertainty in project portfolio cost estimation. *Information and Software Technology, 43*(13), 759–768.

Stamelos, I., Angelis, L., Morosio, M., Sakellaris, E., & Bleris, G. (2003). Estimating the development cost of custom software. *Information & Management, 40*(8), 729–741.

Stensrud, E. (2001). Alternative approaches to effort prediction of ERP projects. *Information & Software Technology, 43*(7), 413–423.

Stensrud, E., & Myrtveit, I. (1998). Human performance estimating with analogy and regression models: An empirical validation, *Proceedings of the 5th International Symposium on Software Metrics, METRICS'98*, pp. 205–213, Bethesda, MD.

Stensrud, E., & Myrtveit, I. (2003). Identifying high performance ERP projects. *IEEE Trans. Software Eng. 29*(5), 398–416.

Summer, M. (2000). Risk factors in enterprise wide information systems projects. *Special Interest Group on Computer Personnel Research Annual Conference Chicago*, Illinois, pp. 180–187.

Symonds, C. R. (1991). *Software size and estimating—MkII FPA*. Chichester, UK: Wiley, pp. 47–91.

Taudes, A., Feurstein, M., & Mild, A. (2000). Options analysis of software platform decisions: A case study. *MIS Quarterly, 24*(2), 227–243.

Vogelesang, F. (2006). Using COSMIC FFP for sizing, estimating and planning in an ERP environment. *International Workshop on Software Measurement*. Shaker Publ: Potsdam, pp. 327–342.

Yin, R. K. (2002). *Case study research, design and methods* (3rd ed). Sage Publications: Newbury Park.

## Author Biographies

**Maya Daneva** is currently an assistant professor with the Information Systems Department at the University of Twente. She is currently a leader of a research program on ERP size and cost estimation. Maya spent 9 years as a business process analyst in the Architecture Group of TELUS Corporation, Toronto, the second largest telecommunication company in Canada, where she consulted on ERP requirement engineering processes, requirements-based efffort estimation models, and function-point counting methods for SAP projects. Prior to this, she was a research scientist at the Institute fuer Wirtschaftsinformatik, at the University of the Saarlandes, Saarbruecken, Germany. Maya Daneva received her Ph.D. degree in Computer Science in 1996 from the St-Climent-Ochridsky-University of Sofia, Bulgaria.

**Roel Wieringa** is a full professor in the Computer Science Department at the University of Twente, The Netherlands, and head of the Information Systems Group there. His research focuses on the design of distributed information systems for organizations. He also chairs the Steering Committee of the IEEE International Requirements Engineering Conference. Roel Wieringa received his PhD degree in Computer Science from the Free University of Amsterdam, The Netherlands.