Converging Perceptions After a Crisis Leading to Successful Change - Dynamics of CSFs in a Post-Merger ERP Program

Completed Research Paper

Stefan K. Mueller
WU-Vienna University of Economics and Business
Welthandelsplatz 1, D2, A-1020 Vienna
stefan.k.mueller@ai.wu.ac.at

Jan Mendling
WU-Vienna University of Economics and Business
Welthandelsplatz 1, D2, A-1020 Vienna
jan.mendling@wu.ac.at

Edward W.N. Bernroider
WU-Vienna University of Economics and Business
Welthandelsplatz 1, D2, A-1020 Vienna
edward.bernroider@wu.ac.at

Abstract

Research on Critical Success Factors (CSFs) for ERP (Enterprise Resource Planning) implementations has paid so far little attention to (1) ERP programs and challenges stemming from interdependent, related projects, and (2) intergroup relations, which are particularly evident in these implementations. We approach this gap and identify CSFs through a single, interpretive, post-merger case study of a complex ERP program, using a grounded theory approach, particularly as a coding technique. We use the Social Identity Theory (SIT) as a lens for understanding the dynamics within the ERP program life-cycle. Our results suggest that, within the ERP program the perceptions of different groups were diverging, which contributed to the crisis within the main project. However, a positive reflection and converging views after the crisis eventually resulted in a successful program. Using the SIT perspective was valuable to interpret the identified CSFs, which offer important implications for research and practice.

Keywords: ERP Program, Program Management, Enterprise Resource Planning (ERP), ERP Implementation, Critical Success Factors (CSFs), Intergroup Relations, Enterprise Systems
Introduction

The current body of ERP literature agrees on the enormous potentials, e.g. in terms of cost savings, a properly implemented ERP system offers to a company (e.g. Holland and Light, 1999; Seddon et al., 2010). Through Enterprise Resource Planning (ERP) systems “multinational firms are able to integrate their geographically dispersed subsidiaries with headquarters resulting in a single uniform and coordinated information system and are, thus, able to coordinate and monitor their performance in real time” (Rajagopal, 2002, p. 89). However, potential benefits are associated with high risks of a complex implementation, or as Davenport puts it: “if you´re not careful, the dream of information integration can turn into a nightmare” (Davenport, 1998, p. 121). These potentials and risks are even higher when more than one project is subject to the ERP implementation. In practice, companies often define a structure around interrelated, individual projects. This structure is typically referred to as a program, being associated with a dedicated program management. Seddon et al. (2010) point out that a group of projects is often coordinated through some sort of overarching program and that normally enterprise systems projects go live in different geographic locations or with different functionality. Experts at Gartner use programs for Information Technology (IT) transformations (Gartner, 2014). Thus, we conclude that new governance structures, manifested as ERP programs, including multiple interdependent projects, are contemporary phenomena potentially reducing the risks of large scale ERP implementations.

Although program management is increasingly used in practice, the structuring of ERP implementations into programs is hardly explicitly covered in the current IS literature. Some of the more explicit contributions include Ribbers and Schoo (2002), who create a framework for critical success factors (CSFs) and their impact on the success of an ERP implementation, but focus only on certain “key elements” for program management. CSFs can be defined as "the underlying or guiding principles of an effort that must be regarded to ensure that it is successful" (Caralli et al., 2004, p. 27). Seidel (2009) develops a model, based on common and site specific CSFs, in order to predict the success of an ERP program but does not consider implementation phases. Programs offer various means and techniques to increase the likelihood of success for strategic, organizational changes (Cabinet Office, 2011; Pellegrinelli, 1997; PMI, 2008), which go along with the introduction of these systems. Even though program management and the interdependence of projects are partially covered in these works, a more explicit consideration is warranted. A few previous studies indicate that group memberships and intergroup relationships are especially important for these IT enabled changes (Schwartz and Watson, 2005). In the context of ERP, managers struggled to control the implementation of ERP projects against the backdrop of changing stakeholder perceptions (Besson and Rowe 2001; Markus et al. 2000) and strong influences of biased project teams (Bernroider, 2013).

This study bridges the gap, stemming from the lack of understanding CSFs across the different phases of ERP programs, and seeks to shed light on the changing challenges and perceptions of a complex ERP program that experienced a dynamic adoption process and a crisis situation. In particular, we investigate critical success factors, with an explicit focus on different program phases before, during, and after the crisis. Furthermore, we interpret our findings from a Social Identity Theory (SIT) (Tajfel and Turner, 1986) perspective, to broaden our understanding about the intergroup relationships, which are important for ERP implementations and particularly ERP programs. In this way, this research fills some blind spots in prior research, provides valuable insights and new avenues for future research. For practitioners it can be used as a point of reference, to ensure that CSFs and group memberships are considered, as well as changing dynamics.

This paper is structured as follows. In the next section we discuss the theoretical background of ERP and program management. Then we describe our research method, which is based on an interpretive case study approach and a variant of the “Straussian” grounded theory method (Sarker et al., 2001, Strauss and Corbin, 1998). Next we present the results of our single interpretive case study of an ERP program, before we discuss the most relevant CSFs and findings of our research. Finally, we close with the conclusion.
Theoretical Background

In this section we elaborate on the necessary research background of ERP and program management. We begin with the characteristics of ERP systems. Afterwards we highlight typical challenges of large scale ERP implementations and lead over to ERP programs and ERP implementation costs.

**Enterprise Resource Planning (ERP) Systems**

ERP systems are off-the-shelf systems standard solutions in contrast to custom applications (Scheer and Habermann, 2000). They are designed with a best practice approach to fit the needs of many organizations, supporting generic business processes (Markus and Tanis, 2000). ERP systems incorporate best practices to facilitate rapid decision-making, cost reductions, and greater managerial control (Bernroider and Hampel, 2005; Holland and Light, 1999).

ERP systems are often defined as a specific type of enterprise system. “Enterprise systems are large-scale, real-time, integrated application-software packages that use the computational, data storage, and data transmission power of modern information technology to support processes, information flows, reporting, and business analytics within and between complex organizations” (Seddon et al., 2010, p. 305). According to this view, the term Enterprise System (ES) includes, amongst ERP systems and other applications, Customer Relationship Management (CRM), Supply Chain Management and data warehousing. This is in line with Davenport et al. (2004) who stress the benefits of ES in terms of process integration. Great benefits stem from seamless information flows within a company and across the inter-organizational supply chain (Markus and Tanis, 2000). Apart from the integration of business functions within the company, ERP systems are therefore also increasingly used to share information beyond organizational boundaries or as Davenport et al. (2004, p. 19) state “Integration also does not stop within a company’s own four walls”.

Typically, ERP systems are of a huge size and costly (Bernroider, 2013). While ERP systems were initially designed for organizations reaching a certain size, the major ERP software vendors have been targeting the small to medium sized enterprises for over a decade (Bernroider, 2001). However, the efforts for implementing ERP solutions are even very high for smaller businesses, which more regularly experience an initial decline in organizational performance after going live (Bernroider and Hampel, 2005). Generally, business process modeling methods can help to reduce the cost of software implementation, and increase user acceptance (Scheer and Habermann, 2000; Dumas et al., 2013), while extensive business reorganization increases the time needed for ERP implementations (Bernroider, 2013).

**Challenges of Large Scale ERP Implementations**

Certainly the most important challenge of a large scale ERP implementation is the major business change, which is concurrently triggered by the introduction of such a system. Norms, underpinned by the stakeholders’ values and beliefs, can be violated, since the organizational environment is changed through the implementation of the ERP system. This is suggested to be the root cause of most ERP implementation problems (Krumpholz et al., 2001). In order to realize the full range of business benefits, an ERP implementation should be accompanied by business process redesign (Accenture, 2011). However, the higher the level of business process redesign (associated changes), the higher the implementation complexity (implementation challenge) (Ribbers and Schoo, 2002; Accenture, 2011), and, consequently, the more resources are expended for ERP implementation (Bernroider, 2013).

The definition of harmonized business processes is a huge challenge, especially when more than one business unit (and/or more than one site) is subject to a large scale ERP implementation. This is usually the case since scale effects lead to cost reductions and time savings during the configuration of the new system (Huber et al., 2000). The definition and the adoption of new harmonized business processes, the establishment of key information entities, the settlement of reporting and information aggregation structures, are time consuming activities (Davenport, 2000). Often, local sites are quite independent and strong, which may result in tensions between local sites and central management. A common understanding of the future business must be developed. This process may be blocked by political conflicts, prestige, communication problems and different priorities and habits. Changes can be forced by a strong management (Gulla and Mollan, 1999). Thus, with respect to the organizational changes and the
harmonization of business processes, large scale ERP implementations are strongly interrelated across business units and sites (Klaus et al., 2000) such that strong management attention is indispensable.

Beside changes how the daily business is conducted a large scale ERP implementation is always associated with interrelations between its elements which increase the implementation complexity. Ribbers and Schoo (2002) propose three measures for implementation complexity. Variety reflects the interrelations in a system and will increase with the number of sites affected or the functions of an implemented package. Variability is related to dynamics over time and the interrelations between the elements of a system. Examples are scope changes, lack of resources, and dependencies on other implementations that are competing for resources. Integration refers to the planned changes which will be realized, the innovation in IT and business processes (Ribbers and Schoo, 2002). In our research, we associate a large scale ERP implementation with interrelations between its elements (e.g. projects) and with a change in the business processes. „While ERP projects are focused on outputs (a functioning ERP system), ERP programmes are focused on outcomes (a change in how the organisation operates)” (Seidel, 2009, p. 18). Since interrelations and changes are key characteristics of large scale ERP implementations, we will use the term ERP programs for the remainder of this paper.

**ERP Programs and ERP Implementation Costs**

In some publications, research on program management acknowledges the use of programs in major IT- and ERP implementations. So does the PMI (2008), which refers to ERP implementations, major IT-implementations, and business process improvement initiatives, as programs which deliver incremental benefits during their life-cycle, which is typical for a phased approach. Pellegrinelli (2002) mentions an ERP implementation as an example, when he investigates missing capabilities of project managers in managing programs. The Cabinet Office (2011) uses ERP as an example for a specification led program. Within ERP research (Davenport et al., 2004; Ribbers and Schoo, 2002; Seddon et al., 2010; Seidel, 2009), the existence of programs as a means to implement an ERP system is acknowledged. Hence we can conclude that, as well from the theoretical position, the use of programs and program management within ERP implementations is recognized.

According to the 2014 ERP Report, the average cost of ERP implementations has been $6.5 million with an average duration of 16.1 months. 54% of the projects have exceeded their planned budgets, while 72% have exceeded the planned durations. Furthermore, 66% of the respondent organizations did realize less than 50% of the anticipated benefits (Panorama Consulting Solutions, 2014). We assume that for ERP programs the budgets and the durations might be higher, since the 2014 ERP Report includes also data from SMEs, while programs are typically in place within large organizations. We can underline that program management is actually applied to implement ERP systems, with budgets sometimes exceeding $100 million. Given the importance of this phenomenon, research has hardly discussed ERP programs within their life-cycle extensively, and we see the absolute need to close this gap.

In this chapter, we highlighted the knowledge areas of ERP and program management, and focused on the challenges of ERP programs, which warrant the central intention of our research. In the next section, we present our research method and discuss why the chosen approach is particularly useful for our study.

**Research Method**

In this section we present our research method. We start with the philosophical stance and our method choice. Then we discuss the quality aspects of our chosen method, before we turn to the data collection and the data analysis (coding procedure).

**Philosophical Stance and Method Choice**

For investigating ERP programs within organizations and to explain them, we took an interpretive philosophical stance. To capture the complex, dynamic, context- and time-dependent social phenomena, in our research setting an interpretive perspective is particularly useful (Orlikowski and Baroudi, 1991). Interpretive studies assume that people create their own subjective and inter-subjective meanings as they interact with the world around them. Our interpretive lens allowed us to explore phenomena by accessing these meanings and thereby better capture human thoughts and actions in social and organizational
contexts (Klein and Myers, 1999). We believe that it is essential to understand these thoughts and actions, to draw conclusions embedded into well considered organizational contexts. As research on ERP programs is still in its infancy, our exploratory study is particularly useful to discover not anticipated features, factors and issues which may also apply to other similar situations (Myers, 1999). Thus, we relied on an in-depth, interpretive case study as the best means for our purposes.

We do not intend to generate or to test universal laws as proposed by positivist researchers (Eisenhardt, 1989; Paré, 2004; Yin, 2003). The design and use of IS systems in organizations are intrinsically embedded in social contexts, which are marked by time, locale, politics and culture. Thus, by neglecting these influences one would only get an incomplete picture of the IS phenomena (Orlikowski and Baroudi, 1991). ERP systems as a subset of IS systems certainly meet these criteria. Furthermore, historical and contextual conditions might trigger events or influence human action (Orlikowski and Baroudi, 1991). Therefore, we follow the line of thought of leading interpretive researchers as Klein and Myers (1999) or Walsham (1995; 2006), who consider contextual and historical conditions.

We aim to meet our main research intentions based on the rich and the specific settings of the investigated case. Our concepts are grounded in data, and we therefore follow the grounded theory approach, for constant comparison, iterative conceptualization and theoretical sampling of our study. In particular, we follow the approach of Strauss and Corbin (1998), and the guidelines proposed by Urquhart et al. (2010). For the coding (analysis) part we use a variant of the “Straussian” approach, which is proposed by Sarker et al. (2001). A related recent account, also drawing on a grounded theory approach for an interpretive single case study, is given by Berente and Yoo (2012), who investigated how different forms of loose coupling are satisfying the demands of contradicting, institutional logics, after the introduction of an ERP system. Instead of generalizing from the specific case to a target population, we rather aim at gaining a shared understanding of the phenomenon and its deeper structure. This can be used to inform other settings (Orlikowski and Baroudi, 1991). Therefore, only thoughtful and cautious inferences to other cases are possible.

**Quality Assurance**

Interpretive research does not use the traditional quality criteria as we know from positivism. Whereas proponents of positivist case study research suggest ensuring quality in terms of validity and reliability (Eisenhardt, 1989; Paré, 2004; Yin, 2003), these criteria are not appropriate for interpretive research (Klein and Myers, 1999). Interpretive case studies define quality in terms of plausibility of the story and the argument (Myers, 2009). Walsham (1995; 2006) provides guidelines how to conduct fieldwork in interpretive research. We concluded that a consistent, shared view, how to conduct interpretive research, in particular, interpretive case studies, exists, and we tried to adhere to these standards.

**Data Collection and Data Analysis**

In our research the concepts emerge from data and there is a continuous interplay between data collection and analysis, referred to as constant comparison (Urquhart et al., 2010). The primary sources of evidence are interviews, which best allow to access the interpretations of the participants regarding the actions and events (Walsham, 1995). In particular, we conducted semi-structured interviews with a special focus on open questions, but also on the context, program design and demographic information. The interview guide was adapted regularly, based on intermediary results. Furthermore, the questions were posed selectively (For a version of the interview guide, please contact the authors). In total we completed 12 interviews with 11 key players (Table 1), with an average duration of 1 hour. We selected the interviewees after consultation with the program manager, who was our single point of contact with the intention to gain rich insights. Furthermore, the interviewed persons represent different groups within the program, and therefore different perspectives. As suggested for interpretive studies, we supplemented the interviews with data from other sources (Walsham, 2006). In our research, we used informal talks, public media and documents as additional sources. Thus, the results are grounded in data and built upon multiple sources of evidence (data triangulation).
As proposed by Strauss and Corbin (1998), we in principle used three coding procedures, including open coding, axial coding and selective coding. The usage of this original three-tier coding scheme has been criticized for being too rigid, for forcing of data, and for hindering emergence. Seidel and Urquhart (2013) showed that while grounded theory is adapted frequently, the usage of the scheme can be varied and depends on the studied phenomena and the intent of a research (Seidel and Urquhart, 2013). One approach to modify the paradigm is well exemplified by Sarker et al. (2001), who related categories hierarchically to their subcategories (concepts). Furthermore, they wrote integrative memos on each major category, including as many subcategories as possible, to accomplish the goals of axial coding. For selective coding, Sarker et al. (2001) used two meta-theories to develop the story line, and to relate the core (central) category to the other categories.

We also used parts of this adapted approach, which can be seen as being less rigid as the paradigm of Strauss and Corbin (1998), but is still based on the basic principles of the “Straussian” grounded theory method. In particular, we started with open coding, where we labeled data chunks with open codes. Then, in step 1 of axial coding, we grouped the concepts into categories and related them hierarchically to their subcategories (i.e. we created network views reflecting the relations). In step 2, of axial coding, integrative memos were written, detailing the relations of step 1, to meet the targets of axial coding. Finally, we chose the meta-theory to interpret the results gained through open and axial coding. Unlike Sarker et al. (2001), we did not define explicitly a core (central) category, since our main aim in this paper was, to interpret the phase specific CSFs from a valuable theoretical perspective, to receive a better understanding of the dynamics within the case. Since many different groups were evident in our case, we found the Social Identity Theory (SIT) (Tajfel and Turner, 1986) particularly useful, to interpret our findings. Based on SIT, a group is a collection of individuals who perceive themselves (and are perceived from others) as members of the same social category (a group). The members share some emotional involvement and social consensus about the evaluation of their group. As members of social groups the individuals achieve an identification of themselves in social terms, their social identities (Tajfel and Turner, 1986). Below we list the three main principles of SIT (Tajfel and Turner, 1986, p. 16).

1. Individuals strive to achieve or to maintain a positive social identity.

2. Positive social identity is based to a large extent on favorable comparisons that can be made between the in-group and some relevant out-groups: the in-group must be perceived as positively differentiated or distinct from the relevant out-groups.

3. When social identity is unsatisfactory, individuals will strive either to leave their existing group and join some more positively distinct group and/or to make their existing group more positively distinct.

We used SIT as lens to interpret the data, but not as means for a more formal integration, at this stage.

Table 1. Interview Partners

<table>
<thead>
<tr>
<th>Role within the Program</th>
<th>Time of the Interview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program manager (2 interviews)</td>
<td>July 2013, April 2014</td>
</tr>
<tr>
<td>Technical coordinator (Rep-SAP)</td>
<td>September 2013</td>
</tr>
<tr>
<td>Work package owner, member of IT-core team</td>
<td>September 2013</td>
</tr>
<tr>
<td>Member of IT-core team, member of program management office</td>
<td>September 2013</td>
</tr>
<tr>
<td>Stream lead, stream 1, functional area 1</td>
<td>September 2013</td>
</tr>
<tr>
<td>Director, functional area 2</td>
<td>October 2013</td>
</tr>
<tr>
<td>Project manager (Change-SAP)</td>
<td>October 2013</td>
</tr>
<tr>
<td>Director, functional area 3</td>
<td>November 2013</td>
</tr>
<tr>
<td>Technical coordinator of the program, project manager (Core-SAP)</td>
<td>December 2013</td>
</tr>
<tr>
<td>Consultant (Core-SAP)</td>
<td>February 2014</td>
</tr>
<tr>
<td>Consultant (Rep-SAP)</td>
<td>March 2014</td>
</tr>
</tbody>
</table>

As proposed by Strauss and Corbin (1998), we in principle used three coding procedures, including open coding, axial coding and selective coding. The usage of this original three-tier coding scheme has been criticized for being too rigid, for forcing of data, and for hindering emergence. Seidel and Urquhart (2013) showed that while grounded theory is adapted frequently, the usage of the scheme can be varied and depends on the studied phenomena and the intent of a research (Seidel and Urquhart, 2013). One approach to modify the paradigm is well exemplified by Sarker et al. (2001), who related categories hierarchically to their subcategories (concepts). Furthermore, they wrote integrative memos on each major category, including as many subcategories as possible, to accomplish the goals of axial coding. For selective coding, Sarker et al. (2001) used two meta-theories to develop the story line, and to relate the core (central) category to the other categories.
A Post-Merger Case Study of the ERP Program at Pegasus

In this section we present our case study. As we deem the settings of the case especially important, we start with the contextual information, then we introduce the different groups within the program, the organization and the environment. Next we present our results assigned to program phases, and we discuss some aspects from a SIT perspective and depict relevant CSFs and corrective actions during our timeline.

**Contextual Information**

The case study was conducted between July 2013 and April 2014 within Pegasus (pseudonym). Pegasus is a European service providing company operating in approximately 10 countries. The industry and markets where Pegasus is operating are highly competitive and technology-orientated. At the time the program started, Pegasus was in a post-merger phase, since the two formerly independent companies PEGA and SUS (operating in the main market) merged previously. The two different organizational cultures were highly visible at the beginning of the ERP program. Therefore, the ERP program was also a high priority post-merger integration initiative, which was sponsored by the board.

During a time period of two and a half years, this large ERP program consisting of several projects was conducted. Table 2 depicts how the program was structured within Pegasus, and which projects were included. Furthermore, Table 2 describes the different projects and lists the major goals. The program team comprised 600 persons (including the testing staff) and more than 100 external consultants. More than 700 interfaces needed to be considered and the number of end users exceeded 5000 persons. The scope included, depending on the functional area, technically consolidated and harmonized processes. E.g., the number of accounting processes was reduced from 1200 to 600 harmonized processes. The numbers reflect the huge challenge faced by Pegasus in terms of the ERP program.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Core-SAP (Core SAP ERP System)</td>
<td>Two legacy systems were merged into one new system, with nearly all modules. The new system includes partially new processes. Depending on the functional area, legacy processes were either harmonized or technically consolidated. The project was subdivided into streams.</td>
<td>Harmonized and consolidated processes were implemented, tested, approved and ready-to-use, data cleansed, and migration completed, back to standard processes, IT-Trainings, documentation, securing operations and support.</td>
</tr>
<tr>
<td>Project Rep-SAP (BI &amp; Reporting)</td>
<td>Two legacy reporting systems were merged into one new reporting system (BI/BW). Additionally, a data warehouse is part of the reporting infrastructure. The project was subdivided into slots.</td>
<td>Meeting all reporting requirements, appropriate reporting architecture under consideration of future requirements, avoiding redundant data.</td>
</tr>
<tr>
<td>Project Change-SAP (Change &amp; Communication)</td>
<td>Since processes are affected a change management project was created.</td>
<td>Stakeholder engagement, communication, change enablement, end user training.</td>
</tr>
<tr>
<td>Project One-SAP+ (Release 2: Core-SAP &amp; Rep-SAP)</td>
<td>A conception project for a 2nd release, for topics which were not in included in the projects Core-SAP and Rep-SAP.</td>
<td>Further harmonization, cover functional gaps and change requests.</td>
</tr>
<tr>
<td>Project Doc-SAP (Documentation)</td>
<td>This documentation project was finally included in the program. Mainly included financial processes.</td>
<td>Complete documentation, meeting the requirements of auditors.</td>
</tr>
</tbody>
</table>

Table 2. The Program Structure at Pegasus
Social Identities and Groups Affecting the ERP program

Within the ERP program of Pegasus every individual belonged to a different group, and each group had intergroup relations (see Figure 1). The amount of groups was increased due to the integrated nature of ERP software, and by the fact that the program comprised several related projects. Furthermore, the post-merger phase increased the number of groups, since all project members were once either employees of PEGA or SUS. A hierarchical order between the groups existed and one individual might resolve intergroup-conflicts by defining him- or herself in terms of the most salient social identity (Ashfort and Mael, 1989), and acting accordingly. In our case that means that e.g. in the beginning, for some individuals, the membership to the group PEGA (or SUS) was more salient than the membership to the new company, and they represented the interests of this dominant group.

Figure 1. Different Groups Affecting the ERP program

Figure 1 depicts a simplified overview of the salient groups and important factors impacting these groups within the program. On the highest level is the environment, which has an influence on the entire organization and the ERP program. Different groups, including shareholders, customers, competitors, implementation partners or the public (appearance and reputation) play an important role. On the next level the internal view is considered, where internal stakeholder groups, e.g. groups defining the strategy and enterprise architecture, and the program sponsors have an impact on the program. On the next level, we have the program itself, including the program management group (at Pegasus it includes, among a dedicated program team also the project managers) and the steering committee. Next, there is the project level, which consists of different streams/slots (depending on the functional area), the internal and external IT. Furthermore, due to the post-merger phase the organizational cultures of PEGA and SUS still played a role, mainly on the stream level. Depicted as dotted circle we have the informal compound group “Basis”, consisting of the stream, internal and external IT. The informal group “Basis” was especially important to understand the dynamics during the ERP program at Pegasus. We referred to an informal group for “Basis”, as this group is not represented in the governance structure, as opposed to formal groups like “stream” or “project”. We also want to mention here that not all groups can be clearly assigned to one level, e.g. “External IT” (depicted as dotted box) is on the one hand part of the project/program,
but on the other hand not part of the environment, or the program sponsors which interface the program and the organizational level.

The social identity of an individual “is an amalgam of loosely coupled identities” (Ashfort and Mael, 1989, p. 30). For example, an individual of one business unit located in stream 3, project Core-SAP, is within the program of the organization Pegasus. Pegasus is surrounded by its environment. Furthermore, the individual was either employee of PEGA or SUS, before the two companies merged. This means that the individual is a member of different in-groups. As a myriad of out-groups exist there is a lot of room for intergroup relations between salient in-groups and out-groups, which might have different views (perceptions) about certain circumstances and interpret situations differently. These intergroup relations will affect the program, as we will see during the listing of events, assigned to certain program phases, in the next section.

**Timeline and Results**

In Table 3 we depict the timeline of the project Core-SAP. The program and the other projects were very strongly influenced by this project, since it was the main project with the highest priority. Therefore, in this paper, we apply a single case study design with multiple, embedded levels of analysis (Yin, 2003). The timeline spans 31 months, which we subdivided into 9 phases.

<table>
<thead>
<tr>
<th>T</th>
<th>No.</th>
<th>Time</th>
<th>Phase</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Oct/ Year 1</td>
<td>Blueprint (Conception)</td>
<td>Pre-Crisis</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Jun/Year 2</td>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Feb/Year 3</td>
<td>Replanning</td>
<td>Crisis &amp; Reflection</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Apr/Year 3</td>
<td>Realization, Different Migration Concepts</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Jul/Year 3</td>
<td>Migration Runs (Test cycle X), Realization, Functional Tests</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Sep/Year 3</td>
<td>Migration Runs (Test cycle Y), Realization, Integration &amp; End to End Tests</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Nov/Year 3</td>
<td>Final Tests, Final Migration Test Runs (Test cycle Z)</td>
<td>Post Crisis</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Dec/Year 3</td>
<td>Deployment/Cut over/Go-live</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Jan/Year 4 – Apr/Year 4</td>
<td>Post Go-live</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3. Timeline of the Project Core-SAP**

**Pre-Crisis**

During the blueprint phase (No. 1 in Table 3), not all the design documents were finished sufficiently. The different teams were formed according to their functional areas within Pegasus. In many cases the process spanned more than one functional area, and different harmonized and consolidated processes needed to fit together. The situation was worsened through the cultural and process differences of the formerly independent companies PEGA and SUS (see Quotation 1, Q1, Table 4). Prototyping was conducted only in a few cases, also due to time constraints. The streams and their IT-counterparts had often difficulties to imagine the complete end to end process without sufficient data. Furthermore, the new external implementation partner (Pegasus selected one general vendor for the blueprint phase) did not meet the expectations. Cultural barriers and knowledge gaps (team setup of the external partner) were the main reasons why, in particular, the streams at Pegasus were not satisfied with the performance of the implementation partner. Due to these reasons the blueprint phase did not reach the anticipated goals.

During the implementation phase (No. 2 in Table 3) the groups within the program tried to catch up. After a new invitation of tenders the program management at Pegasus decided to proceed with a new general implementation partner, who should also involve former subcontractors (with a history as vendor either for PEGA or SUS) more strongly. This decision was appreciated by all program groups (apart from the original vendor). Furthermore, already during the tender, the subcontractors were asked to document
all the existing processes which were outside of the ERP system’s standard. Not only for harmonized processes, but also for consolidated processes, adoptions needed to be made. Conjoint decisions, about the new adopted design, were often not made in a common meeting of all stream members. Instead the functional areas of PEGA and SUS sat together with their IT-counterparts. Given the strict time plan, the latter often finalized the design in a separate meeting and informed their functional areas later. After all, most processes were implemented, tested separately within the respective functional areas, but the integrated end to end view was still not evident for the single streams (see Q2, Table 4).

In general, the cooperation with the new partner was much better in phase 2 than in phase 1, but still different views existed (program and project management vs. external IT; streams vs. internal/external IT, program and project management), concerning the test cycles for the data migration. The migration tools and the external management of the migration-team did not meet the expectations of Pegasus. Moreover, the huge amount of data led to long execution times. Already the first test cycles clearly did not achieve the targets. Thus, the data migration became the major problem of this phase (see Q3, Table 4).

In addition to the problems of the data migration test cycles, not all the problems in regard to the definition of new processes were resolved. The strict time plan (proposed by the formal program sponsor the CFO, and the CTO), to which the program management (including project managers) committed, was seen as very ambitious by the basis (stream level and below, see Figure 1). Some team members even used the term “unrealistic” (see Q4, Table 4), and with regard to the persisting problems, the doubts concerning the planned go-live date in 05/year 3 increased. Before Christmas, in 12/ year 2, almost nobody within the basis believed in the feasibility of the original plan. Since the situation did not change for the better, early in the year 3, also the program management started to consider different scenarios and a rescheduling of the go-live date. In the meantime, Rep-SAP performed well (see Q5, Table 4).

<table>
<thead>
<tr>
<th>Representative Quotations (translated)</th>
<th>Interpretation from a SIT Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Q1:</strong> When is the blueprint detailed and good enough to start with the implementation? This was something where we did not really meet the target. It was a huge challenge, especially for an endeavor of this size. A second major issue during the blueprint was the post-merger-phase of the company. On the one hand the people from PEGA, on the other hand the people from SUS... before they were not really confronted with each other. (Program manager)</td>
<td>In the beginning the “new” post-merger group Pegasus did still not exist in many heads. Instead, the most salient group for individuals in functional areas was still the respective pre-merger company, either PEGA or SUS. Therefore, the strategy to positively distinct the in-group from the out-group was social competition (Tajfel and Turner, 1986), which made an agreement more difficult.</td>
</tr>
</tbody>
</table>

**Open Codes:** Team-Composition; Post-Merger Phase to consider; Blueprint Phase not really finished sufficiently

**Axial Codes:** Efficient Human Capital Management; Ensuring Business Process Management; Case-Specific Context

| **Q2:** The majority of the users is outside the FUNCTAREA2 area.... a lack of awareness that the ERP system is highly integrated and plays an essential role in the entire company. Apart from the FUNCTAREA2 most functional areas did not know that it will also be a change for them. (Director functional area 2) | Every functional area (apart from FUNCTAREA2) is one in-group. They compare themselves with the out-group FUNCTAREA2 on the dimension “relevance”. In the early phases of the program, different perceptions concerning the business impact were perceived. |

**Open Code:** Expandable Transportation of Awareness concerning Importance and Impact

**Axial Code:** Secure Change Management

| **Q3:** We were always one data migration test cycle behind. The vendor expected a technical migration, we expected already a first test migration... these two perceptions did not fit each other... we expected that they provide technically mature migration tools, which was not the case.... The functional areas realized that | Obviously, different groups shared different perceptions. Whereas the program and project expected that the migration tools are technically mature and a first test cycle with plausible data, the ERP vendor saw this test run pure technically. Furthermore, the functional realized that the data will be migrated only partially. This led necessarily |

| **Open Code:** | |
| **Axial Code:** | |
not all the data will be migrated.... and they asked themselves: “how can I work with that data”. (Technical coordinator program, project manager Core-SAP)

to tensions between the groups. (program and project management vs. external IT; streams vs. internal/external IT, program and project management).

Open Codes: Emerging Indicators for Rescheduling; Negative Perceptions concerning Data Migration test Cycles; fail to fully manage expectations  
Axial Codes: Ensure Data Migration/ Accuracy; Define Stakeholder & Communication Management

Q4: The program management and the project managers approved the first go live date, but the basis did not believe that this date is feasible right from the start. With basis, I mean stream leaders, work package owners and project staff. I mean you need to convey this plausibly to the basis, otherwise the date won't be accepted. And we did not manage to transport this properly. Therefore, we lost the basis for a certain time. That was one reason why we had to skip the first go live date. (Technical coordinator program, project manager Core-SAP)

The program management and the project managers affiliated with the higher-status group program sponsors. In other words, the most salient group for the program management and the project managers was the in-group organization. The most salient group for the “basis” remained their informal lower-status group, since the higher status group did not get them on board. Thus, different perceptions regarding the feasibility of the time plan persisted.

Open Codes: Loosing Trustworthiness; Unrealistic Timeframe  
Axial Code: Ensure Realistic Planning of Time Schedule

Q5: We have people who push things forward.....that’s why we said we make our own documents....but we remained lean....This was only possible because we have been our own project....and the program management was happy with our performance right from the start. (Technical coordinator Rep-SAP)

The group-members of Rep-SAP created positive distinctiveness by comparing themselves with the out-group (e.g. Core-SAP) on a new dimension. They had no need to deliver the same amount of documents, and were more flexible. Tajfel and Turner (1986) call that social creativity.

Open Codes: Rep-SAP with separate Methodology Requirements; Project Empowerment is seen positively  
Axial Code: Flexibility of Program Components

Table 4. Representative Quotations (Pre-Crisis)

<table>
<thead>
<tr>
<th>CSFs</th>
<th>Description</th>
<th>Groups at Pegasus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensure Business Process Management</td>
<td>Redesigning business processes in accordance with the ERP strategy and envisioned target business environment.</td>
<td>Streams, Internal and External IT (with Subgroups PEGA &amp; SUS)</td>
</tr>
<tr>
<td>Ensure Realistic Planning of Time Schedule</td>
<td>Defining an ambitious but realistic time frame.</td>
<td>Program Sponsors, Program and Project Management, Basis</td>
</tr>
<tr>
<td>Ensure Appropriateness of ERP Vendor</td>
<td>Choosing the appropriate ERP vendor, ensuring ongoing vendor support.</td>
<td>ERP Vendor 1, ERP Vendor 2, Subcontractors, Internal IT, Streams</td>
</tr>
<tr>
<td>Ensure Data Migration/Accuracy</td>
<td>Ensuring that data is migrated accurately to the ERP system with appropriate tools.</td>
<td>ERP Vendor 2, Basis</td>
</tr>
<tr>
<td>Secure Change Management</td>
<td>Creating awareness of business impact and relevance.</td>
<td>Change-SAP, Streams</td>
</tr>
<tr>
<td>Flexibility of Program Components</td>
<td>Not all of the components needed to meet the same strict requirements, which was particularly good for Rep-SAP.</td>
<td>Project Rep-SAP</td>
</tr>
</tbody>
</table>

Table 5. Most Important CSFs and Involved Groups (Pre-Crisis)
In Table 5 the CSFs (and the particular properties/dimension in the description), which were not really met are depicted. Other relevant CSFs ("Secure Top Management Support", "Establish Governance Structure", etc.) were addressed appropriately. Additionally, in the last row we report on the program-specific CSF “Flexibility of Program Components”.

Crisis and Reflection

Within the replanning phase (No. 3 in Table 3), the original go-live date was cancelled. This decision was appreciated by the basis (stream level and below, see Figure 1). The project Core-SAP was responsible for the cancellation, but likewise all projects were affected (For example, Rep-SAP was performing well all the time). Furthermore, one stream was split into two streams, which reflected strongly the interest of both new groups. The perceived workload decreased heavily for the basis, but at the same time increased significantly, for the program management and the project managers, who started to refine the new schedule and to prepare different scenarios.

Within phase 4 (Realization, Different Migration Concepts, Table 3), the program management prepared a detailed plan for the new go-live date in 01/year 4, which was presented in a decision workshop in 06/year 3. As the data migration was considered as the most serious issue, five different alternatives were proposed. Furthermore, five different deployment scenarios were presented. Additionally, issues with a critical business impact were highlighted. The decision workshop served as a large buy-in of all relevant stakeholders, since beside the CFO, the CTO, the program management and the project managers, also directors and stream leaders (in total 26 persons) participated. Although the new schedule was still ambitious, it was considered as realistic and therefore all participants committed to the new go-live date, and agreed in accordance on a specific combination of the migration/deployment scenario. The consequence was a strong overall commitment to the new envisioned go-live date (see Q6, Table 6).

<table>
<thead>
<tr>
<th>Representative Quotation (translated)</th>
<th>Interpretation from a SIT perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Q6:</strong> The crisis had a positive effect on the complete program. We used the time to reflect what went wrong, what do we need to improve, where do we need to change the structure, how must we change our collaboration... A new quality of collaboration... The team found together... (Project manager Change-SAP)</td>
<td>During phase 3 and 4 the group formation (group &quot;program&quot;) was strengthened. The extent to which an individual identifies with a group may be affected by interpersonal interaction, common history, shared goals or threat (Ashfort and Mael, 1989). We assume that pressure might also have been a condition.</td>
</tr>
</tbody>
</table>

**Open Codes:** Positive Reflection of the Past; New Quality of Collaboration  
**Axial Codes:** Establish Collaboration and Decision Making; Secure Lessons Learned

### Table 6. Representative Quotation (Crisis – Replanning and Reflection)

<table>
<thead>
<tr>
<th>Actions</th>
<th>Description</th>
<th>CSFs influenced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy-in Meeting</td>
<td>Securing the commitment of stakeholders. Decisions made on data migration and deployment scenarios.</td>
<td>Top Management Support, Establish Commitment of Key Players, Ensure Data Migration/Accuracy, Define Program Methodology</td>
</tr>
<tr>
<td>Reinforcement of Data Migration Team</td>
<td>The data migration team was reinforced with additional workforce, most notably at the management level. The program management realized that the collaboration between functional areas, IT, and the implementation partner needed to be improved.</td>
<td>Ensure Data Migration/Accuracy</td>
</tr>
<tr>
<td>Quality Gates introduced</td>
<td>For data migration, end to end tests, Integration Tests Percentage of executed test cases, percentage of priority 1 errors.</td>
<td>Define Program Methodology</td>
</tr>
</tbody>
</table>
An issue management tool was introduced. To prioritize tickets and generate reports. Secure Issue Management, Defining Stakeholder & Communication Management

Convey management decisions to Basis, (decisions made in decision workshop). Secure Change Management, Define Stakeholder & Communication Management

Separate project rooms were provided to strengthen the communication and collaboration, and to improve decision making. Establish Collaboration and Decision Making, Ensure Business Process Management

Table 7. Most Important Corrective Actions During the Crisis

Table 7 depicts the corrective actions, which we deemed most important during the crisis. These actions influenced certain CSFs, as we show in column 3.

Post-Crisis

The data migration test cycle X (No. 5 in Table 3) resulted in additional errors. The bug-fixing, together with the open issues in the implementation (development) of processes, was a challenge for the implementation team and tickets needed to be prioritized. This was done with a dedicated issue tracking tool, which was introduced at that time. The introduction of the tool was generally appreciated and allowed a detailed reporting and communication of the current status (see Q7, Table 8).

The data migration test cycles X and Y (No. 6 in Table 3) continued with the goal to meet the target values of the quality gate. This was the prerequisite to start with the final test cycle Z in 11/year 3. One realized still room for improvement concerning the data preparation for the upcoming data migration runs. Finally, very late, integration- and end to end tests started and new issues popped up. This resulted in new change requests, which needed to be prioritized. Especially in the CS (Customer Service) -stream some critical integration-tests failed. High priority was given to the issues concerning customer oriented processes, and further downtimes and tests were arranged. Additional awareness (at least informally) was given to this topic, since a service oriented company got bad press because of their bad deployment of a new system. By that time at the latest, all groups within the program realized the strong impact an ERP system might have on public appearance, and that the link to the organizational and environmental context is especially important (see Q8, Table 8).

In this phase (Final Tests, Final Migration Test Runs, Test cycle Z, No. 7 in Table 3) the progress slowly approximated to the desired state, as a prerequisite for the formal approval of the go-live scenario. Initially, the end to end tests (with data of data migration test cycle 2) did not meet the envisioned targets and the test phase needed to be prolonged. 19 processes were classified as business critical, and needed to pass the final tests. Finally, with a lot of commitment, and increased professionalism, of the whole program team (especially from the project Core-SAP), all the critical business processes met the acceptance criteria, and data migration test cycle Z passed the target values of the quality gate (see Q9, Table 8). Hence, one day after the final end to end tests, the steering committee formally approved the go-live scenario.

The deployment and go-live phase (No. 8 in Table 3) was seen as a success by all stakeholder groups. No business critical business issues remained and the steering committee approved the formal go-live date. The project Rep-SAP also went live successfully a few weeks later, followed by the project Doc-SAP, which successfully went live in 06/year 4. The project One-SAP+ (conception release 2) was delayed and successfully concluded later, i.e., it was rescheduled due to the higher priority of Core-SAP.
In the post go live-phase (No. 9 in Table 3), all the open issues were addressed and prioritized. Again, also within this phase, the issue tracking tool proved to be very valuable and offered high transparency to the relevant stakeholders (see Q7, Table 8). All issues have either been solved or postponed to release 2. It is important to note that the project Change-SAP was strongly involved in communication and training issues during the post go-live phase. Finally, a last workshop was conducted to reflect upon the entire program. The program was evaluated as a success in most dimensions. While the aspect of collaboration was rated unanimously as excellent in phases 8 and 9, it was assessed as very bad for phase 2 (start of the implementation). Open issues were handed over to operations (including roles, tools, processes), and lessons learned were saved for future programs.

<table>
<thead>
<tr>
<th>Representative Quotations (translated)</th>
<th>Interpretation from a SIT Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Q7</strong>: The open issues were managed with an issue tracking tool. We used it heavily, also for reports depending on the priority and the business impact. (Program manager)</td>
<td>The introduction of the issue tracking tool was appreciated by all groups, since it positively affected the program. Therefore, it strengthened the identification with the in-group “program”, and group formation through shared goals (Ashfort and Mael, 1989). The tool made communication with the out-groups on the organizational level easier.</td>
</tr>
</tbody>
</table>

**Open Codes**: Positive Perceptions concerning the Usage of an Issue Tracking tool; Intensive Communication of Issues and Errors  
**Axial Codes**: Secure Issue Management; Define Stakeholder & Communication Management

| **Q8**: We had many concerns. That’s why we prepared ourselves for the situation. The worst thing is to get bad press, how it happened to the company WILL NOT MENTION... this would have been the worst case. (Stream leader, stream 1, functional area 1) | Employees of functional area 1 belong to different in-groups (e.g. their stream, the project, the program, the organization). The out-group is the company (environment), who failed in implementing an IS properly. Given this situation, it is clear, that one wanted to avoid a negative public appearance, by any means. |

**Open Codes**: No bad Press and no negative Public Appearance; Detailed Cut-Over (Deployment) Planning  
**Axial Codes**: Emphasize Vision and Business Case; Dimensions for Success; Define Program Methodology, Case-Specific Context

| **Q9**: The essential point was the increased professionalism in the migration team, we also had some changes in the team composition...collaboration and the interplay between development, test migrations, tests, integration-tests, end to end tests, data cleansing... was much better. (Program manager) | The relevant in-group formation “Data Migration” was strengthened. Additionally, the interplay with salient out-groups “streams” improved. Also the identification with higher-status groups (project, program, organization) increased. |

**Open Codes**: Strengthening and Restructuring of the Migration Team; Perceptions concerning the Collaboration improved steadily; Increasing Professionalism within the Migration Team  
**Axial Codes**: Ensure Data Migration/ Accuracy; Efficient Human Capital Management; Establish Collaboration and Decision Making

**Table 8. Representative Quotations (Post-Crisis)**

Table 9 (below) depicts the CSFs, which we deemed most important during the Post-Crisis. In the description the properties/dimensions, which were of particular importance at Pegasus are shown.

In this section we presented our interpretive case study. We started with the program structure, followed by the introduction of salient groups within the program at Pegasus. Next, we presented the timeline and continued with our storyline, which is based on social identity theory (SIT). We interpreted some aspects from a SIT perspective and depicted relevant CSFs and corrective actions during our timeline (Pre-Crisis, Crisis and Reflection, Post-Crisis). Furthermore, we considered the salient in-groups and out-groups. Next, we discuss the most relevant findings of our study, its limitations and avenues for future research.
Table 9. Most Important CSFs and Involved Groups (Post-Crisis)

<table>
<thead>
<tr>
<th>CSFs</th>
<th>Description</th>
<th>Groups at Pegasus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensure Data Migration/Accuracy</td>
<td>Collaboration and lessons learned in the migration team, additional workforce for the migration team, committed migration scenario, collaboration with other teams, ensure data freeze.</td>
<td>Streams, Internal IT, ERP Vendor</td>
</tr>
<tr>
<td>Emphasize Vision and Business Case</td>
<td>No negative public appearance, customers should not recognize the change, adapted business case secured, meeting the defined success dimensions.</td>
<td>All Groups within the Organization, especially the Program Groups</td>
</tr>
<tr>
<td>Facilitate Collaboration and Decision Making</td>
<td>Trusting each other and other groups, small teams which are enforced to make fast decisions, interplay between IT and streams.</td>
<td>All Program Groups</td>
</tr>
<tr>
<td>Enforce Program Methodology</td>
<td>Strict quality gates for data migration cycles, end to end testing, integration tests.</td>
<td>All Program Groups</td>
</tr>
<tr>
<td>Establish Commitment of Key players</td>
<td>Absolute commitment to 2nd go-live date.</td>
<td>All Program Groups</td>
</tr>
<tr>
<td>Realize Benefits</td>
<td>Transition into operations, lessons learned, reflecting the project and reviewing the targets.</td>
<td>Streams, Internal IT, Program and Project Management</td>
</tr>
</tbody>
</table>

Discussion, Limitations and Future Research

This research examines the challenges and CSFs of ERP implementation, and the changing dynamics, which seem to unfold particularly in an ERP program environment, where multiple related projects and groups are involved. The results of our interpretive case study may be interpreted in two different ways. On the one hand, the results are confirmatory in terms of some of our identified CSFs were already discussed in prior research. On the other hand, we provide findings which seem to be new, and therefore complementary, in this specific ERP program context. Table 10 summarizes these CSFs and in terms of our confirmatory findings provides extensive references to related literature. Next, we focus on briefly discussing the shaded three CSFs, which we deem as most novel in the ERP program context considered in this paper.

First, we identified "Flexibility of Program Components" as a new CSF, which is clearly program-specific and would not be applicable to a more traditional project level setup. In particular, it relates to separate methodological requirements for different projects as exemplified in this study. The project Rep-SAP performed well from the beginning, and the program management granted Rep-SAP certain tolerances (see Q5, Table 4). Presumably, the ability to account for specific setups and requirements contributed strongly to the final success of the ERP program. This finding is of particular importance for practitioners, but also future ERP research should build more on this insight and its placement within the program management literature. The CSF also refers to exercising program management not by the micro-management of individual projects, which is the independent domain of project managers given certain tolerances set by program management. The program management must create mechanisms to assess the performance of its processes and projects (Cabinet Office, 2011) within these tolerances. “The effective use of tolerances can directly enable the efficient execution of a program” (PMI, 2008, p. 82). Thus, we conclude that this finding is congruent with well accepted standards of program management, and future ERP research should more closely integrate this and other program management concepts.

Second, as a very good example of the changing dynamics of a CSF, we want to highlight "Establish the Commitment of Key Players”. While prior research has confirmed "Top Management Support” as CSF for
ERP projects (Bernroider, 2008; Markus and Tanis, 2000; Ribbers and Schoo, 2002; Seidel, 2009), we showed that this statement needs to be extended to account also for the commitment of other key players in the context of ERP programs. The strong commitment of all program groups (e.g. the basis, and the groups on project-, program-, organizational-, environmental level) which was established in the later phases, including the converging perceptions (e.g. Pegasus replacing PEGA and SUS as most salient group), was a major determinant for the success of our case. Similarly, Besson and Rowe (2001) stress the commitment of the ERP project’s leading actors. We also believe that a contextual factor within the environment, that is the failure of the IS implementation (see Q8, Table 8), had a certain impact on this CSF. The PMI (2008) refers to these outside influences as program-external factors. Thus, since particularly in an ERP program context many different groups (in- and outside the program) are involved, strong attention should be paid to this CSF.

<table>
<thead>
<tr>
<th>CSFs</th>
<th>References (Program Management Literature first)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish Commitment of Key Players</td>
<td>Besson and Rowe (2001), Bernroider (2008). Further ERP program research warranted, since more key players (and different perceptions) might be involved than in a traditional project setup.</td>
</tr>
<tr>
<td>Establish Collaboration and Decision Making</td>
<td>(Markus and Tanis, 2000). Further ERP program research warranted, to consider the mass of different groups.</td>
</tr>
</tbody>
</table>

Table 10. CSFs related to ERP & Program Management Literature (Most Novel CSFs shaded)
Third, we highlight "Establish Collaboration and Decision Making" as another CSF, which unfolded over the different ERP program phases. This CSF significantly improved in the post-crisis phase, also through the corrective actions taken during the crisis. The corrective actions in later phases had been necessary, since the outcomes of earlier phases became starting conditions for the next phase, and thereby either increased or decreased the likelihood of success (Markus and Tanis, 2000). The role of different groups of people is also partially tackled by Markus and Tanis (2000), and they mention the communication difficulties that accompany the “handoffs” between phases. We believe that in an ERP program, where many salient groups are involved, it is of major importance to establish an environment, where effective "Collaboration and Decision Making" can take place.

We already mentioned that we did not exactly define a core category, although, considering the use of SIT as a lens for interpreting our findings, we believe that especially "Defining Stakeholder- and Communication Management" might be an appropriate candidate category. We believe that many CSFs are related to different perceptions, and effective stakeholder- and communication management appears to be necessary to influence them properly. Of course, there are other possible core categories, but this will be a part of our future research, where we will integrate our concepts into a more formal theory.

Our findings rely on a single interpretive case study, and are grounded in data. Although we emphasize the context, additional research is necessary to broaden our understanding and to establish more foundations. Also other settings, where programs include more than one site are of great interest. Furthermore, due to the space constraints of this paper, we were not able to cover all CSFs in full detail.

Some CSFs are context related. For example, the post-merger setting of the case and the failure of the IS implementation in the environment (Cabinet Office, 2011; PMI, 2008) are highly relevant. Since an ERP program deals with change and is of strategic nature, the context appears to be even more important than in a traditional project setup. Future research would be useful to establish a more detailed understanding and a broader set of accounts regarding these different contexts.

Conclusions

In our research we conducted an interpretive in-depth case study, and investigated a complex post-merger ERP program, with a special focus on the contextual information and the salient groups involved in the ERP program. For the interpretive case study, we followed well accepted guidelines of leading interpretive researchers, and we did the same for the coding part, where we used parts of an adapted grounded theory approach, which can be seen as being less rigid as the paradigm of Strauss and Corbin (1998), but is still based on the basic principles of the “Straussian” grounded theory method.

We uncovered a range of CSFs of ERP programs, which are not explicitly covered in the current body of ERP literature in the context of program management. Our analysis has a special focus on phases and shows that the influence of a single CSF (see Tables 5 and 9, reference to literature in Table 10) is dynamic and changes in the course of these phases. We also detected novel program-specific CSFs, in particular "Flexibility of Program Components". Future research can build on this foundation.

The SIT perspective is valuable, and relevant, to broaden our understanding why ERP implementations are successful. This is particularly true for ERP programs where the number of different salient groups is generally higher than in a traditional project setup. The perceptions of the groups contributed to the crisis. Only to mention a few: the basis and the program management had different views concerning the feasibility of the go-live date. Next, for many individuals PEGA and SUS were still the most salient groups. Finally, in regard to the data migration different views existed between Pegasus and the 2nd ERP vendor. A positive reflection and corrective actions, during the crisis, contributed to converging perceptions, stronger commitment, faster decision making and eventually to a successful go-live. We believe the application of SIT taking into account perceptions and intergroup relations is a fruitful theoretical approach to explain the dynamics of an ERP program.

Our findings suggest that practitioners should consider the different perceptions and intergroup relations, within their ERP implementation (and particularly programs, to increase the likelihood of a successful change). Securing shared goals and the unity of the program is of high importance. Therefore, change management and stakeholder- and communication management should be addressed properly, especially in settings where many salient groups are involved.
References


