Understanding Requirements in Enterprise Systems Projects

Jon Atle Gulla
Norwegian University of Science and Technology
Trondheim, Norway
Jon.Atle.Gulla@idi.ntnu.no

Abstract

Due to the generality and complexity of enterprise systems, they are challenging to implement and deploy successfully in organizations. Many of these problems are rooted in the way requirements are analysed and negotiated among the stakeholders. Critical success factors have been used as guidelines for enterprise systems projects, but their applicability depends on the context and the goals and scopes of the projects. We propose a model that explains the success of the requirements engineering process in terms of the project’s ability to balance various concerns. This model defines a number of indicators that may expose unbalanced requirements and clarify the use of critical success factors in enterprise systems projects.

1. Introduction

Enterprise systems are large complex information systems that integrate and streamline the organization’s business processes across departmental and geographical borders. The systems are vital to the daily operations of the organization, and any deficiency or system failure can cause havoc to its business. Among the most well-known enterprise systems are ERP (Enterprise Resource Planning) systems, which provide pre-implemented modules for backbone operations like finance, sales and distribution, and materials management. Due to the customization nature of these systems, traditional requirements engineering activities have a less prominent place in these projects than in traditional software engineering projects.

However, experience tells us that enterprise systems projects suffer from a wide range of problems. One in four ERP projects is over budget. About 20% of the projects are terminated before the implementation is completed, and 40% of the companies confirm that their ERP project failed to achieve their business objectives [2]. In 1996 FoxMeyer Corporation filed bankruptcy after spending $64 million on their failed ERP project. The company sued the consulting company and the ERP vendor for $500 million each [21]. Other projects, like Hydro Agri, were reasonably successful at the end, but faced both delays and substantial changes of scope and requirements [10]. The importance of adequate and clearly stated requirements is emphasized in several recent studies of enterprise systems projects (e.g. [4]).

Even when projects are completed and a well-functioning enterprise system solution is delivered, there may be deeper requirements problems that only show up after some time. The reverse engineering tools of Businesscape and Intellicorp expose the real use of the operative ERP system, which is often quite different from what was envisaged at the requirements stages. In Daneva’s study of 67 SAP subprojects in the telecommunication industry, about 13.5% appeared successful from an engineering perspective, but did not meet the real needs of the organizations [4].

In an attempt to evaluate the results of ERP implementation projects and guide later projects, critical success factors (CSFs) for ERP implementations have been introduced [7]. These form guidelines that should ensure that the enterprise system is successfully implemented, deployed and received. Even though some of them address the requirements phase, the emphasis is more on project management and implementation approach. The same is also the case for ERP implementation methodologies like ASAP [13].

As noted by several studies [4,15], however, there is no one best way to develop enterprise systems. There are successful projects that have made vastly different implementation choices with respect to customization, process harmonization, and stakeholder involvement. Although the CSFs reflect comprehensive studies of successful and failed ERP projects, they must be used with caution and interpreted in a particular context.
In this paper we develop a deeper understanding of how requirements are handled in enterprise systems projects. We show how the requirements engineering process can be regarded as a process of balancing requirements along three dimensions. Firstly, there has to be a balance between technological requirements, workflow requirements and management requirements. Secondly, this balance has to be preserved when the company leaves the overall strategic requirements and start digging into the detailed tactical requirements. And lastly, the company needs to weigh the level of customization against the level of organizational change. Using this model, we can start explaining project problems and provide more insight into the ways the CSFs should be applied.

Section 2 discusses some of the particularities of enterprise systems that complicate the requirements engineering process. In Section 3 we present the requirements engineering model for enterprise systems that will afterwards be used to interpret early indicators of project problems and expose the background and applicability of critical success factors. Section 4 through 6 are devoted to the requirements issues related to customization, organizational tiers and planning horizon. Section 7 uses an SAP project to illustrate the use of the model, and the discussion of the model as well as our conclusions are found in Section 7.

2. The Challenges of Enterprise Systems

Enterprise systems are today among the most complex systems to develop and deploy in organizations. SAP R/3, the market leader among ERP systems, offers several thousand transactions that potentially penetrate about every business process of the company. This complexity combined with the generality of customizable modules makes ERP projects very large and difficult to control. A survey among Australian companies showed that it took between 6 and 7 years to complete an ERP project from early design to a successful company transformation [1]. In the Hydro Agri project, 120 applications were replaced by an SAP R/3 solution that integrated 47 legal companies [10]. The complexity is further increased by cultural and legal differences among the sites to be supported by the enterprise system [14,19].

A consequence of the wide coverage of enterprise systems is the wide range of knowledge needed to implement them. In [6] this knowledge can be categorized as business knowledge, technical knowledge, product ERP knowledge, company-specific knowledge, and project management knowledge. Successful ERP projects are truly collaborative projects, where the quality of the final solution depends on the knowledge of a large number of domain experts and ERP consultants.

What often disturbs this collaboration is the organizational resistance to change. Employees may feel threatened by the new system, or they may not want to change their current tasks or routines. Overcoming this resistance is difficult, but may be crucial to the involvement of the users and the subsequent success of the project.

The issue of wicked problems is also getting more apparent in the enterprise system sector. A system engineering problem is wicked if the system requirements and the implemented system mutually affect each other. As soon as the ERP system is in operation, thus, new requirements tend to pop up, leading to ever new cycles of requirements engineering and system implementation.

Finally, an enterprise system project cannot afford to fail, and an enterprise system cannot afford to break down. Since these systems are needed for the business to run, their stability and reliability are immensely important. The systems must be able to deal with any kind of business situation that may emerge, also those that have never occurred in the past.

What these challenges reveal, is that the organizational make-up and needs must be carefully analyzed if the enterprise systems project is to succeed. There are no fixed answers as to what the requirements should be, and different requirements and different implementation approaches tend to work in different circumstances [15,21]. We cannot determine if the requirements in an enterprise system project are correct or not. At the most we can work out principles that indicate if the requirements are unbalanced or wrong.

3. A Requirements Engineering Model

The organization’s requirements reflect the often conflicting needs of three different parts of the organization (see Figure 1). These parts involve the management itself, the process organization and the IT staff, and their requirements need to be justified and prioritized across the tiers. The emphasis of each of these three tiers may vary from one project to another, though success requires that the balancing of these tiers is in accordance with the emphasis chosen. In radical BPR projects, for example, the workflow requirements are deemed more central than some of the technology requirements. In the following, we briefly describe the concerns at each tier.
3.1 Organizational tiers

Management tier. Management tools like the Balanced Scorecard, Value-Based Management and EFQM stress the importance of performance measurements systems [12]. Establishing key performance indicators for the most important business drivers is considered vital for the continuous monitoring and improvement of the business. In the Balanced Scorecard framework, these indicators address the perspectives of finance, customers, internal processes and innovation & learning. The EFQM framework emphasizes the performance related to employees, customers, and society at large. Many of these performance indicators can be related to data available from the enterprise system.

Defining the desired business processes for the enterprise system, the organization needs to make sure both that the performance indicators capture the relevant aspects of the process carried out and that the process as a whole supports the management goals underlying the indicators. This last issue has been a considerable concern in many ERP projects. A strict approval procedure for requisitions and purchase orders in the system, for example, allows the company to control costs at a very detailed level and enforce managerial powers in purchasing. On the other hand, if these procedures are too constraining, they hamper the efficiency of the process and can be counter-productive to the motivation and development of employees. The functionality of the ERP system, thus, has to be adopted and adapted in accordance with the organization’s strategies and goals.

Workflow tier. Harmonized and streamlined business processes are the goals of most ERP projects. The projects need to analyze how organizational boundaries can be torn down and people work more efficiently together. This process optimization requires that people change their departmental focus and evaluate the efficiency of their operations in terms of total process efficiency. Moving from resource optimization within the department to process optimization for the whole company is considered an important part of most enterprise systems project. This requires that the system requirements include a proper analysis of current work routines and propose new and more efficient ways to run the business.

Technology tier. The new enterprise system supports the backbone operations of the organization and must be interfaced with a number of other applications. This implies that new interfaces must be written and the technical infrastructure for linking applications has to be determined. Hardware, additional software, networks and support organizations have to be analyzed and specified as part of the requirements. Also, routines for converting data from the old legacy systems have to be developed. The new system must fit the organization’s enterprise architecture plans and work seamlessly with the organization’s other applications and technical constraints.

3.2 Planning Horizon

Whereas the management tiers reflect the needs of different parts of the organization, there is also a distinction between strategic requirements and tactical requirements at each tier. Strategic requirements concern long-term goals and policies that set a direction for years to come. The tactical requirements address issues that need to be resolved to put the system into operation. Many of the strategic requirements are given by the management, like the enterprise system’s support of key performance indicators, but there are also strategic workflow requirements and strategic technology requirements. If the company has an enterprise architecture plan, many
of its elements will have to be incorporated into the strategic technology requirements. One of Hydro Agri’s strategic workflow requirements was to create a pan-European business flow, where products were sourced from any of the 17 production sites to the correct sales office all over Europe.

The tactical management requirements include the retrieval and aggregation of performance data from the systems. At the workflow tier, there are tactical requirements that decide for example how documents flow between departments and what authorization profiles should be assigned to the employees.

3.3 Customization

Finally, the requirements tend to be anchored to existing structures and routines in the organization or the standard functionality of the enterprise system. Some of the requirements reflect the particularities of the organization and cannot be adapted to the best practice functionality offered by the ERP system. Other requirements come as the result of redesigning old processes into more efficient ones. However, there are also constraints in the ERP system that forces the organization to accept some ERP-inspired requirements. Many organizations, for example, have restructured their organization to fit the organizational structures implemented in the ERP system. The requirements specification must decide to what extent the organization should adapt to the ERP functionality and vice versa.

The balance of organizational change and system customization is often decided at the top management level and is closely aligned with the objectives of the whole project. Whereas some enterprise systems projects just replace existing applications with new and more efficient ones, other projects use the enterprise system to fundamentally change the whole organization.

3.4 Model Structure

The requirements in enterprise systems projects reflect the needs at different organizational tiers, the balance between strategic and tactical concerns, and the desired adoption of standard ERP functionality (see Figure 1). Central to our requirements engineering model is the assumption that the project’s focus and overall objectives decide how requirements should be balanced and reconciled. In a successful project, the requirements for each of the three areas are balanced in accordance with the priorities of the project. This applies to both the specification of the requirements and the actual realization of requirements in the implemented solution. If the balance is disturbed in the course of the project, we expect some dissatisfaction to emerge. The signs of this dissatisfaction, the indicators, warn us of potential project problems that may lower the quality of the final system solution if they are not acted upon. All the indicators included in the model are well-known from other studies, even though they have not previously been related to such an explanatory model.

The model also establishes a link from unbalanced requirements to critical success factors that may be relevant to deal with these problems. Our CSFs come from [7], which incorporates and classifies factors from a number of earlier studies on ERP systems.

4. Balance of Organizational Tiers

The requirements must take into account the needs of the management, the process organization, and the technical staff. There has to be a balance between these tiers that is consistent with the scope and goals of the project and is acceptable to the stakeholders of the project. The important thing is not necessarily that all tiers should have the same weight in the project, but that the project has made conscious choices about which requirements are the most important ones.

If the influence of the three organizational tiers change in an unhealthy manner, the project risks developing a solution that is more adapted to the needs of some stakeholders than others. Indicators that this may happen are listed in Table 1 below. The corresponding potential problems for the project are also included. For example, if the management does not exercise its authority in favor of the new enterprise system, the project may not be able to use the new solution to improve the organization’s business processes.

Included in the table are also the critical success factors that may prevent these problems from emerging or can be used to control a situation where they have already emerged. Trust between partners, for example, is important to ensure that users get involved in the projects and communicate well with each other.

If these potential problems are not addressed in time, the project will typically not address properly the needs of all the stakeholders. Often new projects are defined right afterwards to solve the outstanding issues and calm the organization. The organization is not able to integrate its operations because groups or departments gradually get dissatisfied with the solution.
Projects with successful cooperation at all organizational levels tend to opt for and succeed at implementing radical ERP solutions [15]. In the terminology of [21], these systems have adopted a comprehensive success model. However, projects focusing on a particular tier may also lead to a satisfactory result. Adopting a technological success model, for example, means that the technology tier is given the most weight and management needs and workflow needs are considered less critical to the success of the project. What will typically cause serious problems is to emphasize requirements that are not consistent with the chosen success model.

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Problem situation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management support weak or absent</td>
<td>The project does not get the authority to change the organization’s processes and policies</td>
</tr>
<tr>
<td>User involvement weak</td>
<td>All user needs are not known to the project</td>
</tr>
<tr>
<td>No trust in organization</td>
<td>Users do not pay attention to the needs of the whole organization</td>
</tr>
<tr>
<td>No prioritization of requirements</td>
<td>Wrong requirements are given priority</td>
</tr>
<tr>
<td>Reporting needs ignored</td>
<td>Reporting information may not be available with the customization chosen</td>
</tr>
<tr>
<td>Project considered an IT project rather than a business project</td>
<td>The solution does not meet the business objectives</td>
</tr>
<tr>
<td>Lack of communication inside organization</td>
<td>Departments are not able to collaborate and agree on set of consistent requirements</td>
</tr>
</tbody>
</table>

Table 1. Organizational tier problems

5. Balance of Planning Horizon

Most enterprise systems projects start out by analyzing important strategic needs of the organization. This is a process that is usually led by the management or is heavily supported by the top management team. Even though management needs tend to dominate these discussions, also workflow strategies and IT strategies are taken into account. The result is a specification of overall requirements that govern the subsequent work in the project.

Gradually, though, the project needs to start working out more detailed tactical requirements. Many of these are not specified up-front, but are rather introduced by process owners and IT staff as the project proceeds and prototypes are tested and evaluated. At this stage it is important that the tactical issues do not overshadow the strategic requirements from the early phases. There will be tactical constraints that are not unifiable with the original strategies, and the project needs to find ways of resolving these inconsistencies in a balanced and well-founded manner.

Projects that have suffered from an undesirable focus on tactical requirements have often experienced that project structures and project members have changed in the later phases. The indicators for this problem are listed in Table 2 below. Normal project routines tend to break down, domain experts disengage in the development or leave important decisions to external consultants, or the project is faced with difficult and unanticipated technical challenges. The result is a situation, in which abstract requirements are replaced by directly verifiable requirements.

Many of the problems here could have been avoided if the project had been better planned from the very beginning. This also applies to the way requirements are assessed and specified. A typical problem in many ERP projects is the lack of precise requirements when the implementation is started. The assumption is that these requirements will be uncovered as prototypes are tested and the project goes through the parameters for each module of the system. However, the experience in most projects is that this shifts the emphasis too much onto the tactical requirements. These requirements can be directly related to parameters or technical issues and are easy for both domain experts and ERP implementers to understand.
6. Balance of Customization

Excessive customization leads to a solution that is expensive to implement and maintain. Add-ons and other non-standard functionality have to be tested heavily during upgrading and may often need to be reprogrammed. The whole solution is more error-prone and unstable than what a standard enterprise system would offer.

On the other hand, it is not unproblematic to stick to standard functionality and adapt the organization correspondingly. If end-users are not properly prepared, they may not understand the system and use it correctly. And imposing standard business processes that may be highly inefficient for a particular organization may in the end be more costly than customizing the system in the first place.

The balance between system customization and organizational change is often a political or strategic one. It depends on the degree of business process reengineering involved, the enthusiasm of the management team, and the ability to drive changes through the organization. It is not obvious whether an organization should go for extensive business process reengineering or just customize the software to the current needs of the organization. Whereas a comprehensive reengineering was recommended in many earlier publications, it was often unclear what that should mean for the degree of customization. A common recommendation today is to adapt the organization to the standard functionality of the ERP system, independently of that means extensive or moderate process reengineering [7]. However, there may be features of the organization’s structures or business that make it impossible to adopt the standard functionality of the ERP system. Legal documentation requirements, for example, tend to vary somewhat from one country to another, and not all these requirements are currently supported by the standard software. Another area is functionality that is currently outside the scope of current ERP systems, but need to be used by the organization as an integral part of their new ERP system.

Projects that are not able to create the appropriate balance between process redesign and system customization are characterized by the indicators in Table 3. Many of them refer to the human effects of introducing large enterprise systems and reflect an understanding that enterprise systems projects do not succeed unless the employees want them to succeed.

### Table 2. Planning horizon problems

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Problem situation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements engineering replaced by prototyping</td>
<td>Strategic issues are ignored.</td>
</tr>
<tr>
<td>IT focus too high</td>
<td>No deeper understanding of organizational needs</td>
</tr>
<tr>
<td>Strategic issues and workflow issues ignored in tactical requirements</td>
<td></td>
</tr>
<tr>
<td>Management too passive throughout project</td>
<td>Lack of commitment and ability to change the organization successfully</td>
</tr>
<tr>
<td>User involvement fades in later phases</td>
<td>Surprising end result.</td>
</tr>
<tr>
<td>External consultants too influential</td>
<td>Requirements inconsistent</td>
</tr>
<tr>
<td>Over-emphasis on time and costs</td>
<td>Too much customization</td>
</tr>
<tr>
<td>Formal project plans break down in later phases</td>
<td>Requirements ignored in implementation phase</td>
</tr>
<tr>
<td>Complex structure of legacy systems</td>
<td>Delays, scope creeps and budget problems</td>
</tr>
<tr>
<td>Technology issues over-emphasized</td>
<td></td>
</tr>
</tbody>
</table>

Relevant critical success factors

- Legacy system knowledge
- Preventive trouble shooting
- Formalized project plan/schedule
- Sustained management support
- Adequate ERP implementation strategy
- Dedicated staff and consultants
- Strong communication inward and outwards
- Empowered decision makers

If there is no proper balance between strategic and tactical requirements, the solution tends to be unsatisfactory over time. Some strategic requirements tend to be left out, and new projects will later need to be started to address these requirements. The solution may however appear as a success when it is put into operation. It fulfills the most important tactical requirements and does not violate many obvious requirements. These systems are referred to as invisible failures in [4].

The critical success factors here concern the way the project is set up and organized. A project that is well composed and well supported by the organization is better prepared to deal with the stress and tactical problems that show up in the later phases of the project.
**Indicators** | **Problem situation**
---|---
Employee resistance to change | Customizing system is politically easier than changing work processes
Organization with unrealistic feeling of uniqueness | Misconception that “everything” has to be customized
Weak project management | Line organization enforces unnecessary customization
Modest BPR component | Customization only used to preserve existing processes
End-user training is weak | Standard functionality used with limited adaptation to current processes
No proper assessment of system’s impact on organization | Organization not prepared to changed work processes

**Relevant critical success factors**

- Avoid customization
- Comprehensive business process reengineering
- Adequate training program
- Effective organizational change program
- Appropriate usage of consultants

**Table 3. Customization problems**

The balance between system customization and organizational change must be chosen carefully in each individual project. Numerous enterprise systems projects have failed terribly because of excessive customization, but there are also failures due to a lack of interest in business process optimization and an overrated trust in standard functionality. Danova refers to these two types of project failures as catastrophic failures and visible failures, respectively [4]. On the other hand, it may in many cases be reasonable to choose a point of balance that implies very little customization. These companies have a good chance at succeeding and are described as frugal ERP adopters in [15].

Most critical success factors advice the projects to avoid customizing the system and rather spend the time and resources on preparing the organization for improved business processes. This reflects the fact that many projects find it easier to customize than to change the way the organization works. If the organization is to change its business processes, all users need extensive training and support. They also need to be convinced that the new system makes it worthwhile to change their habits and routines. The new processes may also have unanticipated effects on the organization that need to be uncovered and dealt with before the system goes live. All these tasks involve human cooperation and approval, which is a problematic issue if the users fear that the new system will lead to layoffs and outsourcing.

### 7. The Hydro Agri Case

We can apply the whole model to discuss and visualize the characteristics of ongoing or completed enterprise systems projects. Take for example the Hydro Agri project, which lasted from 1995 to 1999 and had a total budget of $126 million. Hydro Agri (now called Yara) is the world’s leading producer of mineral fertilizers, with revenues of around $6 billion at the end of the 90s. During the late 80s and early 90s it acquired a number of companies in the UK, Holland, France, Germany and Italy. These sites were initially run as local profit centers, and major objectives of the Hydro Agri project were to harmonize the processes across sites and use the ERP system to integrate data and build a pan-European well-coordinated organization.

The project involved more than 500 people over 4 years and was carried out in cooperation with Accenture. 10 modules of SAP R/3 were partly or fully implemented, and more than 3,000 end-users were trained. Even though the solution was fairly successful at the end, there were some problems that caused considerable concern and can now be related to our requirements engineering model. The analysis below is based on the two papers published [9,10], as well as the author’s full participation in the project and interviews/discussions with other project members. Some of the observations also stem from a later SAP upgrade project, in which the author also participated, that was carried out with the help of Deloitte Consulting in 2000.

- The management needs were somewhat underestimated at both the strategic and tactical level. Hydro Agri’s use of Value-Based Management was not taken into account, and its basic reporting requirements were not completely fulfilled in the first version. Partly as a consequence of this, an improvement project was afterwards formed that concentrated more on the business side of the system. Whereas the strategic requirements were worked out in close collaboration with the top management team, the tactical requirements did not get the same attention.

- The workflow requirements were well analyzed and followed up both at the strategic and the tactical level. The organization spent considerable time on defining a pan-European business flow that encompassed both production facilities and
The workflow requirements were given a higher priority at the strategic level than management requirements and technology requirements. There was a good balance between strategic and tactical requirements.

The technology requirements involved the replacement of numerous applications and the implementation of interfaces to internally developed production software and externally bought specialized software. Some of the strategic challenges linked to the realization of a common infrastructure for the whole company were somewhat underestimated and caused problems at the tactical level. As is often the case with ERP projects, the tactical technology problems were dominating, reflecting some of the project and staff problems at the end.

What is also clear in the Hydro Agri project, is that the management requirements and workflow requirements were not ambitious enough with respect to the original vision of a pan-European business supported by one integrated enterprise system. The management was reluctant to changing their old reports, which meant that the project had to implement add-on reports that were almost identical to and sometimes even inferior to standard reports in SAP. The Sales & Distribution of SAP was substantially reprogrammed to fit the existing sales strategies of Hydro Agri. Too much of the existing reporting structures and workflow were kept, forcing the project to customize extensively and implement add-ons that helped the employees use the new system as they had used the old one. Some of these issues were later addressed in a continuation project. Other modules, like Materials Management and Plant Maintenance, were kept more or less unchanged, leading to organizational changes to accommodate the system functionality. For the technology requirements, they seemed to be pretty well balanced between system customization and organizational change. A new IT support department was set up centrally to replace many of the local IT support staff that were not needed any more.

In Figure 2 the characteristics of the Hydro Agri project are shown graphically using the model introduced. We cannot know what the ideal balance should have been, but have tried to indicate what was planned in the project (planned balance) and how it was assessed afterwards (real weight). In general, the project focused more on the workflow tier than on the other two tiers, and there was less discrepancy between planned priorities and real priorities at the technology tier than at the other two. As illustrated on top of Figure 2, the management needs were deliberately given a somewhat low priority. As the project proceeded, some of these needs were almost ignored while the technology needs received increasingly more attention than what was planned. If we look at the balance of planning horizon, we see a related mismatch between planned and real priorities. Whereas the idea was to use as much effort on tactical management requirements as strategic ones, the project ended up specifying the strategic requirements in more detail. Since inter-company process harmonization was so important, the project had deliberately chosen to emphasize strategic workflow requirements. More tactical workflow requirements than anticipated were however realized as the system was customized.
planned technology requirements had a tactical emphasis that was also reflected in later project work. At the bottom of Figure 2 we also see the intended balance between organizational change and extensive ERP customization. The project’s ambition was both to change the organization to take full advantage of the standard ERP functionality and to customize the system extensively where cultural or competitive issues were at stake. In reality they ended up customizing more than anticipated to satisfy existing management structures and existing workflow. Both the management part and the workflow part were set up too close to the existing organizational structures, which led to unnecessary complications in the customization phase. At the technology level, the intention was to substantially change the way the organization worked. This succeeded, and a new and efficient central support organization was built up as the local applications were phased out and local staff cut down.

The analysis above helps us clarify important dependencies in enterprise systems projects. It does not tell us whether an ongoing project will fail or not, but can help us understand why certain problems show up in a project and what can be done with them. The problems in the Hydro Agri project indicated that management needs did not receive as much attention as other needs, especially at the tactical level. They also indicated that too much customization may have been carried out. We can then go back to our model and see how the weight of the requirements may be changed to create a better balance. In this case, it would have been advisable to spend more time on working out tactical management requirements and new routines and structures at the management and process organization level. Additionally, our model tells us which critical success factors we should consider to deal with the situation at hand.

8. Related Work and Conclusions

We have introduced a model that explains the success of the requirements engineering process in terms of the project’s ability to balance various concerns. Rather than defining how a project should be run or how the requirements should be, the model acknowledges that each individual project has to define its own priorities with respect to customization and organizational change, organizational tiers, and strategies and tactics. The model then defines a number of indicators that may expose unbalanced requirements and potential project failures. The critical success factors discussed in many recent publications are classified according to their ability to preserve or establish a certain balance. Using the indicators as early alarm signals of project problems, the projects can now verify how the critical success factors may be applied to alleviate the situation. The model, thus, provides the context for understanding both the effect of the critical success factors and their limitations.

Previous research on the success and failure of enterprise systems projects tends to focus on guidelines for how to organize and run large projects. This ranges from simple and informal recommendations (e.g. [8]) to detailed surveys of proposed critical success factors [7]. Although valuable in itself, this research does not take fully into account the fact that the best choice of implementation approach varies from one project to another. It is also mainly concerned with operative issues and has little to offer for the nature of system requirements. Another strain of research tries to classify projects on the basis of success rates and/or overall implementation approach [4, 15, 16, 21]. Our model addresses the requirements engineering phase only and tries to explain how unfortunate requirements may lead to project failures. It is at a more detailed level and of a more explanatory nature than the work on project classification.

Even though most of the empirical material underlying our model comes from ERP projects, the model has a wider applicability to enterprise systems in general. All complex and generic organizational software needs to reconcile potentially conflicting requirements from different stakeholders, and many project failures stem from the fact that one or more stakeholders have an unhealthy influence on the total requirements. The three dimensions of the model make explicit which needs that have to be balanced or prioritized in the requirements engineering phase.

Further work includes a refinement of the critical success factors, so that they better match the indicators in the model. The factors are now described at different levels of detail, they do not cover all the potential requirements problems, and they are partly overlapping. The CSFs would be more useful if they could be linked directly to the corresponding indicators and problem situations.

It is worth noting that such a detailed account of requirements is not directly incorporated into ERP methodologies like ASAP (Accelerated SAP). A modification of this methodology, in which structuring and balancing requirements is emphasized, would be beneficial. This may also require that requirements engineering techniques be modified or extended. Both Gulla & Brasethvik [11] and Rolland & Prakesh [17] have suggested more powerful enterprise modeling
languages that allow a more accurate modeling of management and workflow requirements. Reusing requirements from old enterprise systems projects may also make it easier to find the correct balance of requirements [3], though there is a danger that the project copies requirements and policies from other projects that had very different scopes or goals.

References


