Chapter 5
Supply chain management in the ERP industry

5.1 Introduction

Supply chain management is a management philosophy that takes a systems approach to viewing the supply chain as a single entity (Mentzer et al., 2001). It seeks integration of the business processes that make up the system across organizational boundaries to increase the performance of the supply chain as a whole. Implementing the supply chain management philosophy therefore entails a shift from market transaction towards long-term relationships through contractual and non-contractual agreements (Jagdev and Thoben, 2001). In the early days of supply chain management, these agreements where primarily targeted at increasing the efficiency of the flow of goods (Sila et al., 2006). Nowadays, their scope has been extended to include all functions within the supply chain. Vehicle manufacturers, for example, do not only rely on inventory management to increase performance, they also give their suppliers more design responsibility to reduce time-to-market of new product introductions (Twigg, 1998).

Researchers in the field of supply chain management generally focus on industries that are based on physical goods, such as the automotive (Choi and Hong, 2002; Holweg and Miemczyk, 2003), the food processing (Dzever et al., 2001; Van Donk, 2001), and the construction industry (Vrijhoef and Koskela, 2000; Fearne and Fowler, 2006). In these industries, supply chain management is mainly concerned with reducing inventory levels across the supply chain. The exchange of sales information to prevent large fluctuations in production rates upstream of the supply chain, generally referred to as the bull-whip effect, is a well-known example (Lee, 2000). Another example is vendor-managed inventory, which refers to a supplier taking full responsibility for replenishing the stock held by its customer (Slack, 2007).

The software industry, in contrast, has received little attention so far, despite the fact that software in general and enterprise software in particular has several interesting properties. First, because of the non-physical nature
of software, the marginal costs of reproducing and distributing a copy of a software system are negligible. Second, the supply chain of enterprise software does not only cover the software product, but also a range of value-adding services, such as installation, configuration, and customization. Finally, many development decisions in the domain of enterprise software are directly related to change of installed products. An example of this is how the consistency of an installed software product can be guaranteed if parts of the product can have multiple versions. Another example is how the migration of a customer’s data from one version to another can be achieved.

From the industry-specific characteristics outlined above, it follows that reducing inventory levels across the supply chain will not be an issue for companies that are involved in the production, sales, and delivery of enterprise software. Instead, it can be expected that producers of this type of software are mostly concerned with integrating quality management and requirements management across the supply chain. Customers use an enterprise system often in combination with extensions from other software vendors. Without any form of coordination, this may result in malfunction when a customer upgrades the system. In addition, producers of enterprise software generally incorporate some externally produced components, so the quality of their systems is at least partially affected by the quality of the components that they receive from their suppliers. Quality management, i.e. the process of planning and controlling the quality of a product or service (Silag et al., 2006), is therefore one of the business processes that is likely to be object of supply chain coordination. The requirements for enterprise systems are constantly changing because of changes in the business environment, such as the introduction of new legislation, technologies, and business models. Because of this, requirements management, i.e. the process of controlling changes to system requirements (Sommerville, 2004), is the other business process that is likely to be object of supply chain coordination.

The objective of this chapter is to explore how software companies integrate quality management and requirement management across the supply chain by conducting two exploratory case studies in the domain of enterprise software. First, an overview is provided of relevant concepts from the field of operations management. Then, the general supply chain concept is adapted to the specific characteristics of the software industry. Subsequently, the research methods are discussed. Next, the results from the two case studies are presented. Finally, the chapter concludes by discussing the implications from this research and by providing some suggestions for further research.
5.2 Conceptual framework

To define the focus of our research, this section specifies the concepts we intend to study. According to Eisenhardt (1989), a priori specification of constructs is valuable because “it permits researchers to measure constructs more accurately”. Voss et al. (2002) go even further by arguing that without a clear focus case study research risks “degenerating into a fishing expedition, where the observer is hoping to catch valuable insights that in turn will lead to research questions”.

5.2.1 Supply chain

A supply chain is a network of firms engaged in the production, storage, distribution, and delivery of a product or service. As is shown in Figure 5.1, the individual firms in a supply chain are linked to each other through a forward flow of goods (solid arrows) and a backwards flow of information, such as customer orders, and payments (dashed arrows). Taking the manufacturer as focal firm, we use the term supply network to refer to the upstream part of the supply chain and the term distribution channel to refer to the downstream part of the supply chain. In analyzing a supply chain, a distinction can be made between primary and supporting members. Primary members perform value-adding activities designed to create a specific product or service for a particular customer or market, whereas supporting members support primary members by providing (non value-adding) assets, resources, knowledge, and utilities (Lambert and Cooper, 2000; Min and Zhou, 2002).

![Figure 5.1: A schematic representation of a basic supply chain.](image)

5.2.2 Supply chain management

The purpose of supply chain management is to coordinate the activities of the independent entities in order to increase the performance of the supply
Chapter 5. Supply chain management in the ERP industry

chain as a whole. The different coordination mechanisms that supply chain members use to integrate and manage their business processes across organizational boundaries can be classified along two dimensions (Simatupang et al., 2002): (1) the mutuality of coordination, and (2) the focus of coordination. The mutuality of coordination refers to “the underlying values of responsibility among partners with a strong emphasis on sustaining relationships in order to build effective goal attainment” and consists of two elements: (1a) complementarity of processes, and (1b) coherency of understanding. The focus of coordination, the second dimension of the framework proposed by Simatupang et al. (2002), also consists of two elements: (2a) focus on operational linkages, and (2b) focus on organizational linkages.

Table 5.1, which is taken from Simatupang et al. (2002), shows the four modes of supply chain coordination that result from combining the two dimensions. The first three coordination modes, i.e. decision synchronization, information sharing, and incentive alignment, have already been described in Section 1.2.3 and are therefore not further elaborated on in this section. Collective learning, the fourth and final mode of supply chain coordination, is concerned with how supply chain members “tackle the coherency problem of initiation and diffusion of knowledge across borders”, and serves to “extend each partner’s capability that is useful for accomplishing ongoing improvement” (Simatupang et al., 2002). Two examples of this latter coordination mode are supplier development teams, who assist suppliers in improving their operational performance (e.g. by eliminating manufacturing problems), and guest engineers, i.e. technical specialists who are employees of a supplier of technology or design expertise, but who reside on a semi-permanent basis at a customer organization (Twigg, 1998).

Table 5.1: Four different modes of supply chain coordination.

<table>
<thead>
<tr>
<th></th>
<th>Complementarity of processes</th>
<th>Coherency of understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational linkages</td>
<td>Decision synchronization (Object: products and services)</td>
<td>Information sharing (Object: information)</td>
</tr>
<tr>
<td>Organizational linkages</td>
<td>Incentive Alignment (Object: benefits and risks)</td>
<td>Collective learning (Object: knowledge and capability)</td>
</tr>
</tbody>
</table>
5.3 The software supply chain

Firms in the software industry provide software assets or services related to software assets, such as deployment, configuration, and customization. Software assets can either be complete systems, also referred to as software packages, or individual parts of a system, called components (Szyperski, 1998; Sommerville, 2004). A component may be large or small and can be thought of as a black box that can be called by other components to perform a certain task. The details of the execution of the task are hidden. For example, a postcode checker may be one of the components of a logistics system. The logistics system itself can, in turn, be one of the subsystems of a larger ERP package. The component concept is one of the enablers of the emergence of supply chains in the software industry because it enables division of work and purchasing of components from external suppliers (Hopkins, 2000).

5.3.1 Supply network

Firms in the supply network of a software manufacturer add value by providing components and subsystems. Before the primary members of the software manufacturer’s supply network can be described and analyzed, some kind of starting point is needed for determining the boundaries of its software system unambiguously. The license agreement between a software manufacturer and an end user turns out to be a suitable criterion because it contains a description of the components that the software manufacturer considers to be an integral part of the system. It is for these components that the software manufacturer will take full responsibility for planning and controlling the release of new versions. In this way, consistency of the software system throughout its lifetime is guaranteed as changes to the system’s integral components are extensively tested before the new versions are made available to the customer. In order to safeguard customers against version inconsistencies, a relatively broad definition of the software license is required: it should encompass all relevant components, including those that are publicly available (i.e. open source components).

From our criterion for the boundaries of a software system, we derive the following rule for demarcating the supply network of a software manufacturer: a component supplier is a primary member of the supply network if the use of the component is covered by the license agreement between the software manufacturer and the end user. Consider an operating system supplier
as an example. An ERP system depends technically on the services provided by the operating system, but the use of this system is generally not included in the license agreement. The operating system supplier is therefore not considered to be a primary member of the supply network of the manufacturer of the ERP system.

5.3.2 Distribution channel

Firms in the distribution channel of a software manufacturer add value by distributing the system, installing it at the customer’s site, and tailoring it to the customer’s specific requirements. In particular, a distinction can be made between resellers, which are mainly responsible for sales and implementation of the system at the customer, and independent software vendors (ISVs), which, depending on the degree of customer specification freedom that is supported by the software manufacturer (see Figure 2.4), develop interfaces with other software packages and create extensions for specific application domains, such as fashion retail.

5.4 Methods

To identify some of the supply chain management initiatives in the software industry, semi-structured interviews were conducted at two companies from the domain of enterprise software. One of them (referred to as company A) was a manufacturer of ERP software, the other company (company B) was a reseller of enterprise information systems, including ERP software. The reason for selecting these two companies for further investigation was the following. Although the companies belonged to different supply chains, they operated under similar market conditions as both supply chains delivered an ERP system that is targeted at midsize and larger companies. Because of this, the case study conducted at the second company can be seen as a literal replication of the case study conducted at the first company, which increases the external validity of the study’s findings (Yin, 2003). To increase construct validity, a document study was performed to complement the answers provided by the respondents. Internal validity is not an issue for our study because we are not concerned with making causal claims (Yin, 2003).
5.5 Results

5.5.1 Case A: the ERP manufacturer

Company A is a producer of ERP software. Spread across several countries, the company serves over 100,000 customers through an extensive network of resellers. The company has organized its ERP system as a product line: the system consists of a large number of components that can be composed in customer-specific ways.

Controlling the quality of components developed externally turns out to be problematic. As a result, company A has developed most of its ERP system in-house. An exception is the off-the-shelf database management system that forms an integral part of the ERP system. The use of the database is covered by the license agreement of company A, so the database supplier is a primary member of the company’s supply network. Company A actively maintains the relationship with the database supplier by organizing monthly meetings in which changes, requirements, and new features are discussed. In developing the ERP system, the company uses several commercially available software tools to support user interface development, communication among components, etc. The tool suppliers are considered to be supporting members of the supply network of company A.

The distribution channel consists of a large number of resellers that add value by selling and installing the product. Company A provides a software development kit (SDK) that allows resellers to extend its ERP system by creating customizations (i.e. (small) changes/extensions to one or more of the system’s existing modules) and add-ons (i.e. new modules that contain additional functionality and/or business logic). By keeping the messaging architecture as stable as possible, the company guarantees a forward compatibility of the ERP system with all extensions that are developed with the SDK. Some companies in the distribution channel of company A are specialized in providing solutions for specific industries, such as healthcare or construction. To control the quality of the domain-specific add-ons produced by these ISVs, company A has started a certification program. The company distinguishes two types of certification: certified development partner and certified software solution. An ISV is entitled to use the predicate “certified development partner” if at least one of its employees has successfully completed one of company A’s training programs on the SDK. The predicate “certified software solution” is given to all add-ons that have been implemented according
to company A’s development guidelines and quality standards. All certified solutions are included in a catalogue that is published on the company’s website.

In the selection of the requirements to be fulfilled for a new release, quantitative market information is needed. This requires an information flow from the resellers to company A, which is supported by a web portal. The information flow back to the resellers on the requirements selected for development is not supported.

5.5.2 Case B: the reseller

Company B is a reseller of enterprise information systems. The company serves over 1,200 customers in more than 10 different countries. Its product portfolio contains several ERP systems. The one of company A is not included. The case study is concerned with the ERP system that is targeted at midsize and larger companies.

Recently, company B privatized its business unit engaged in the development of domain-specific add-ons for the ERP system. The resulting company is further referred to as company C. The reason for privatization was that the market for the domain-specific extensions produced by this former business unit is much broader than the customers of company B only. Some of company C’s extensions are marketed, sold, and supported by the ERP manufacturer as part of an industry solutions program. In this program, the ERP manufacturer works together with selected ISVs to deliver complete solutions for specific industries. The industry-specific extensions developed by the ISVs in the program are extensively tested to ensure that they meet the quality standards of the ERP manufacturer. In addition, the ISVs commit themselves to support every new release of the ERP system.

Company B incorporates domain-specific add-ons from company C as well as extensions from other ISVs. Custom solutions are developed in-house, sometimes in close cooperation with company C. If several customers require a solution that was initially developed for one particular customer, the solution may be sold to company C and then exploited as a domain-specific add-on.

Company B acts as a test organization for new releases of the ERP system. The company keeps complete copies of its customer configurations, so testing includes integration testing with all customizations and add-ons that are used by these customers. In this way, compatibility problems are detected
before the release is delivered to and installed at the customer’s site. New releases are only accepted if they pass company B’s quality standards. Occasionally, a new release is not accepted. Bugs in the software are reported to the manufacturer of the ERP system. Sometimes, if the ERP manufacturer does not solve the bug fast enough or if company B has a better technical solution, company B produces a workaround for a bug in the ERP system. It may happen that the work-around of company B is transferred to and used by the ERP manufacturer as a final solution for that bug.

5.5.3 Case analysis

Table 5.2 displays the data from the two individual cases according to a uniform framework. For each row in the table, the entry in the first column indicates the link in the supply chain that is integrated, and the entry in the second column shows the mechanism that is used to achieve supply chain coordination. To contribute to a better understanding of how software companies integrate quality management and requirements management across the supply chain, Tables 5.3 and 5.4 classify the coordination mechanisms from Table 5.2 along the two dimensions described in Section 5.2.2.

Quality management

Many customers use an ERP system in combination with domain-specific add-ons from other software vendors. To control the quality of these extensions, both ERP manufacturers have initiated a certification program that comprises two modes of supply chain coordination: decision synchronization and incentive alignment. ISVs that participate in the program are required to implement the same development guidelines and quality standards as the ERP manufacturer, which can be seen as a form of decision synchronization as it guarantees consistency with the ERP package. ISVs find it important to have their solutions certified because it increases the value that customers attach to the solution. In addition, the solution is marketed and sold through the reseller channel of the ERP manufacturer. This latter aspect gives ISVs an extra motivation to participate in the certification program and can therefore be seen as a form of incentive alignment.

The two ERP manufacturers differ with respect to how they manage the risk of extensions breaking down after an update of the ERP system. To prevent extensions from breaking down after an update, the ERP manufacturer
## Table 5.2: Summary of the case study results.

<table>
<thead>
<tr>
<th>Coordination Mechanism</th>
<th>Case A</th>
<th>Case B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database supplier - ERP manufacturer</td>
<td>Exchange of customer requirements and new features are discussed (A1)</td>
<td>Testing of new releases of the ERP system (B2)</td>
</tr>
<tr>
<td>ERP manufacturer - Reseller</td>
<td>Exchange of workarounds for bugs in the ERP system (B3)</td>
<td>Exchange of customer requirements (A2)</td>
</tr>
<tr>
<td>ERP manufacturer - ISV</td>
<td>Standardization of the messaging architecture (A3)</td>
<td>Certification of domain-specific add-ons (A4)</td>
</tr>
<tr>
<td>Reseller - ISV</td>
<td>Certification of domain-specific add-ons (B1)</td>
<td>Certification of ISVs (A5)</td>
</tr>
</tbody>
</table>

### Case A
- Database supplier - ERP manufacturer: Database supplier and ERP manufacturer coordinate on monthly meetings in which changes, requirements, and new features are discussed. The coordination mechanism includes exchange of customer requirements, and new features are discussed. The coordination mechanism also includes standardization of the messaging architecture.

### Case B
- Reseller - ISV: The coordination mechanism includes exchange of workarounds for bugs in the ERP system.
- ERP manufacturer - Reseller: Testing of new releases of the ERP system.
- ERP manufacturer - ISV: Exchange of customer requirements. Certification of domain-specific add-ons and ISVs.

<table>
<thead>
<tr>
<th>Coordination Mechanism</th>
<th>Case A</th>
<th>Case B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP manufacturer - ISV</td>
<td>Certification of ISVs (A5)</td>
<td>Certification of ISVs (A4)</td>
</tr>
<tr>
<td>ERP manufacturer - Reseller</td>
<td>Exchange of customer requirements (A2)</td>
<td>Testing of new releases of the ERP system (B2)</td>
</tr>
<tr>
<td>Reseller - ISV</td>
<td>Exchange of workarounds for bugs in the ERP system (B3)</td>
<td>Exchange of customer requirements (B1)</td>
</tr>
<tr>
<td>Database supplier - ERP manufacturer</td>
<td>Standardization of the messaging architecture (A3)</td>
<td>Certification of domain-specific add-ons (B4)</td>
</tr>
</tbody>
</table>

### Chapter 5: Supply chain management in the ERP industry

Chapter 5 focuses on the coordination mechanisms and the respective roles among different players in the supply chain of the ERP industry. The table above provides a summary of the coordination mechanisms observed in the case studies. The coordination mechanisms include exchange of requirements, testing of new releases, and exchange of workarounds for bugs. The cases demonstrate the importance of coordination and communication among different players to ensure smooth and efficient operations in the supply chain.
Table 5.3: An overview of the coordination mechanisms that software companies use to integrate quality management across the supply chain.

<table>
<thead>
<tr>
<th>Operational linkages</th>
<th>Complementarity of processes</th>
<th>Coherency of understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Decision synchronization</em> A3, A4, B1, B2, B3</td>
<td><em>Information sharing</em></td>
</tr>
<tr>
<td>Organizational linkages</td>
<td><em>Incentive Alignment</em> A4, B1</td>
<td><em>Collective learning</em> A5</td>
</tr>
</tbody>
</table>

from case study A has developed an SDK. By keeping the messaging architecture as stable as possible, the company can guarantee a forward compatibility of the ERP system with all customizations and add-ons that are developed with the SDK. This can be seen as a form of decision synchronization as it decouples the release planning of the ERP manufacturer from the release planning of the reseller or ISV that develops the customization or add-on. The ERP manufacturer from case study B, in contrast, has made little attempt in preventing extensions from breaking down due to an update of the ERP system. Instead, the company relies heavily on its resellers to detect and resolve compatibility problems before the new release is deployed at the customer’s site. This explains why company B keeps complete copies of all customer configurations.

**Requirements management**

In the first case study, requirements management is coordinated at two different links in the supply chain. Meetings between the ERP manufacturer and the database supplier are used to discuss the features that are going to be incorporated in the next release of the database management system. These meetings allow the ERP manufacturer to align its future releases with those of the database supplier and can therefore be seen as a form of decision synchronization. The sharing of customer requirements between the ERP manufacturer and its resellers serves to align future releases of the ERP system with the observed changes in the needs of customers and can be seen as an example of information sharing.

In the second case study, company B and company C have integrated the
Table 5.4: An overview of the coordination mechanisms that software companies use to integrate requirements management across the supply chain.

<table>
<thead>
<tr>
<th></th>
<th>Complementarity of processes</th>
<th>Coherency of understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational linkages</td>
<td><em>Decision synchronization</em></td>
<td><em>Information sharing</em></td>
</tr>
<tr>
<td></td>
<td>A1, B4</td>
<td>A2</td>
</tr>
<tr>
<td>Organizational</td>
<td><em>Incentive Alignment</em></td>
<td><em>Collective learning</em></td>
</tr>
<tr>
<td>linkages</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

requirements management process by exploiting customer-specific solutions as domain-specific add-ons when it turns out that there is a larger market for these solutions. The transfer of extensions from company B to company C can be seen as a form of decision synchronization as it allows both companies to focus on their core competences.

5.6 Discussion

In industries that are based on physical goods, supply chain management is mainly concerned with reducing inventory levels across the supply chain. However, because of the non-physical nature of software, this will not be an issue for companies that are involved in the production, sales, and delivery of enterprise software. Instead, we have argued that producers of this type of software are mostly concerned with integrating quality management and requirements management across the supply chain. Our results showed that software companies mainly rely on decision synchronization to coordinate their activities with regard to these two business processes, but examples of information sharing, incentive alignment, and collective learning were found as well. The case analysis also suggested that the current coordination mechanisms are insufficient to achieve the desired level of product quality across the entire supply chain. This applies especially to the quality of the components that software manufacturers receive from their suppliers, which has led company A to develop most of its ERP system in-house.

The management of quality in supply chains is an issue that has received attention recently (Lo and Yeung, 2006; Sila et al., 2006). An important con-
clusion from this stream of research is that extending quality management practices to the whole supply chain requires a high level of quality planning and control within the individual companies. It is also stressed that quality management strategies must be aligned with those of supply chain partners. One way for organizations in the software industry to increase their performance would therefore be to apply these lessons from the field of operations management.

For example, several quality standards have been developed for the software industry, such as CMMI (Ahern et al., 2007). The CMMI model assumes that software development organizations evolve according to a staged maturity model. The process areas that require supply chain coordination are mainly concerned with the CMMI category “engineering”, which comprises the process areas requirements management, requirements development, technical solution, product integration, and verification and validation. Requirements management is needed to reach maturity level 2 (“processes performed”); the other ones are required to obtain maturity level 3 (“processes defined”). This latter maturity level requires standardization of processes at an organizational level. Lockamy and McCormack (2004) argue that supply chain management can only be applied successfully when organizations have defined and documented their processes. It can therefore be expected that the coordination of quality in software supply chains can only be achieved if all primary members have reached the third maturity level. In addition, a kind of decision synchronization is needed to ensure that software manufacturers and their suppliers apply uniform quality management strategies. In the case of CMMI, this requires the alignment of goals and practices in the process areas that are covered by the model.

To increase external validity, future research may be directed at replicating the study’s findings. In addition, one should take into account that ERP software has specific product and market characteristics. Software companies belonging to supply chains that deliver other types of product, such as embedded or tailor-made software, may face different challenges in managing their supply chains. Further research is therefore required to determine to what extent our results can be generalized to other parts of the software industry.