Upgrading to a New Version of an ERP System: A Multilevel Analysis of Influencing Factors in a Software Ecosystem

Philip Holst Riis
Copenhagen Business School
Department of IT-Management
Copenhagen, Denmark
ph.itm@cbs.dk

Petra Schubert
University of Koblenz-Landau
Department of Computer Science
Institute for IS Research
Koblenz, Germany
petra.schubert@uni-koblenz.de

Abstract

This paper presents research findings about the process of upgrading from an old to a new version of a pre-packaged enterprise system in a software ecosystem of independent software vendors (ISVs) and value added resellers (VARs). Empirical data was collected from documents, observations and interviews with practitioners. Grounded theory was used to analyze the data. The resulting theory illustrates the upgrade decision process from the perspective of ISVs and VARs respectively and from the perspective of the software ecosystem as a whole. The findings suggest that the interdependence of the actors in the ecosystem may cause inertia in the diffusion of new versions.

1. Motivation and Research Questions

This paper investigates the factors that influence the adoption of, or the upgrade to, a new version of a packaged enterprise system in the ecosystem of a large enterprise system vendor. We are using the term “upgrade process” to refer to the change from one release to a new release. Kremers and van Dissel [20] call this process “ERP system migration” acknowledging at the same time that there are different names used by different ERP vendors (e.g. upgrade, update or new release). We follow their definition of “[…] a major change process resulting from the implementation of a new version of an already installed ERP system”.

Standard enterprise systems have been available for more than 20 years [8]. In the course of these 20 years the available ERP systems have gone through a constant process of development and technological upgrade from formerly COBOL-based monolithic 2-tier systems to service oriented 3- or 4-tier architectures. In parallel with the technological development, the customers of such systems (ERP user companies) have been required to upgrade to new versions of their existing system or decide to switch to a new system (from a different vendor). The typical life cycle for a standard ERP system from the perspective of the user company is between 10 and 20 years [7]. This means that user companies are frequently facing the decision to upgrade/switch to new versions. The pressure to change to new versions/upgrades has been additionally increased by changes in the legal and economic environment such as an increase in intercompany collaboration (B2B integration) and changing legal regulations (e.g. batch tracing). Another reason for change stems from improvements in functionality and ease of use. Users are increasingly used to “easy to use”, intuitive interfaces and expect the same comfort from their ERP environments. Without a regular update of the look and feel of the application, ERP systems are likely to be perceived as “old” just by their graphical impression.

Against this background it is interesting to examine the consequences of this constant need for technological and functional adaptation. In our research we aimed to understand the ecosystem (the “distribution channel” or “supply chain” for ERP systems) and the factors that influence vendors and users of ERP systems alike to change to an updated or new version of a standard ERP system. The research question examined in this paper is the following:

Who are the players that shape the composition of an ecosystem for ERP systems implementations and how does the composition of the ecosystem influence the diffusion of a new/upgraded version of a packaged enterprise system from vendor to customers?

The paper is structured as follows: We start by outlining our motivation and research questions. We then present a short literature review and the background of our research. We explain the research method and the process of data collection and analysis. The main part
is dedicated to the emerging explanatory model (our research findings). We then discuss our findings and conclude with implications and future research.

2. Literature Review and Research Method

The two terms “enterprise system” and “ERP system” have been widely discussed in the literature. Looking at their definitions, these terms seem to be used almost synonymously. In his influential article from 1998, Davenport described enterprise systems as “commercial software packages” that “promise the seamless integration of all the information flowing through a company” ([8], p. 121). This broad definition comprises all software that is used by a company to support its business processes. Staehr [29] suggests a similar definition for ERP systems: “Enterprise resource planning (ERP) systems are large, complex software packages that provide an integrated real-time environment based on an enterprise-wide data model with a set of software applications that allow processing of the core transactional data of the organization” (p. 213). In this article we are using the two terms enterprise system and ERP system synonymously.

1.2 Literature Review

While early implementations of enterprise systems relied on the development work of a software vendor to make the software fit to the individual needs of a company, pre-packaged enterprise systems that suit the needs of many customer companies have now become dominant [6]. The delivery model of pre-packaged enterprise systems is increasingly developing from a two-party (vendor-customer) configuration to loosely coupled value networks [4], also referred to as “software ecosystems” [25].

The ecosystems typically consist of a vendor at the center, also referred to as a keystone [1] or a hub [17], which develops the core of the enterprise system. A number of partners also referred to as niche players [1], or spokes [17], deliver a range of products and services complementing the core system delivered by the vendor. Previous research has addressed multiple perspectives of enterprise software ecosystem, including the motivation for forming the partnerships [4], coupling and control ([21]; [17]), value creation [10], and competitive advantage ([19]; [3]). A significant amount of previous research has also addressed the adoption and implementation process of enterprise system in customer organizations ([24]; [32]; [23]) and the impact on the organization ([11]; [35]).

The inter-linked nature of ecosystems suggests that the success of adoption of innovations in enterprise system ecosystems is dependent on adoption of all players in the ecosystem rather than the adoption of any single player [1]. However, little research has addressed the process of adoption in the ecosystems (i.e. the software selection) preceding the implementation of enterprise systems in customer organizations.

2.2 Background of the research setting

The enterprise system vendor in the study is a major global player in the market for enterprise systems. The vendor followed the consolidation of the enterprise systems market in the early 2000’s [18] and acquired a number of enterprise system solutions resulting in a portfolio of systems primarily targeted at small and medium enterprises (SMEs). The vendor releases a new major version of the enterprise system approx. every 2-3 years and a so-called service pack with bug fixes and other improvements in between the major releases. The new major version included in this study was launched in late 2008 and a service pack was released in autumn 2009.

The vendor sells and implements the enterprise system only through an ecosystem of partner companies and the consultants in the partner companies thus handle all implementations. The consulting companies can broadly be categorized into two different types: Independent Software Vendors (ISVs) and Value Added Resellers (VARs). Figure 1 displays the generic roles of the players in an ecosystem for enterprise systems.

The ISVs develop additional reusable software modules for the enterprise system, also termed add-ons. These add-ons complement the core enterprise system in a range of areas from generic horizontal functions, such as payroll, online banking, and ship-
ping to very specialized, vertical solutions, such as education, veterinary medicine, legal companies, and furniture manufacturing. The business model of the ISVs is to sell licenses for the add-ons to customers through the VARs, who in turn get a share of the license fee.

The consultants at the VARs take on the implementation of the enterprise system at the customers. The consultants make customizations to the enterprise systems on request from the customers but unlike the ISVs the customizations are customer-specific and seldom reused across different customers. The VARs generate the majority of their revenue from invoicing the time spent on the implementation and customizations and only a smaller part of the revenue is generated from getting part of the license fee. On a typical implementation of the enterprise systems only 1-2 consultants are involved depending on the amount of customizations needed.

Some of the partner companies have characteristics of both an ISV and a VAR, meaning that they develop reusable add-ons which they sell to VARs and they have a staff of consultants implementing the enterprise system together with their own add-ons. Table 1 shows the distribution of VAR and ISV characteristics in the partner companies in which the interviews for the study were conducted.

2.3 Research Method

The study was carried out using grounded theory [15] as the frame for data collection and analysis. Grounded theory is a “data centric” inductive method for analyzing (primarily qualitative) data for the purpose of building or extending theory [31]. The method has been evolved and applied to multiple research studies in the field of information systems [33].

The method stands out from many other research methods by emphasizing that researchers rid themselves of theoretical pre-conceptions about the area of inquiry as the grounded theory should emerge from the data – not through deduction or hypothesis testing [16]. The extent of this condition has not only fuelled debate among researchers using the method but also among the two founders of the method, concerning the risk of forcing theory from the data instead of allowing the theory to emerge [12]. The details of this debate is beyond the scope of this paper but the implications forces a stance on the role and use of existing theoretical literature in the study. The approach to existing literature in this study was a “middle of the road” approach where a general orientation within the literature of adoption of technology and diffusion of innovations was present prior to the analysis of the data, but no preexisting theoretical constructs were forced on the data and a detailed comparison with existing literature was not conducted until after the grounded theory had emerged.

Urquhart et al. [34] provide five guidelines for conducting grounded theory in the IS field: Constant comparison; Iterative conceptualization; Theoretical sampling; Scaling up; and Theoretical integration. Besides providing a guide and support for IS researchers embarking on using grounded theory the five guidelines also explicate the essence of the method.

Constant comparison is the process of constantly comparing instances of data to a particular concept or category for the purpose of exposing theoretical properties of the concepts and categories. This guideline was followed by constantly comparing all the coded instances of data to other coded instances of data sometimes resulting in the same instance of data being recoded several times and in an iterative process of splitting and merging codes.

Iterative conceptualization suggests that researchers should increase the level of abstraction and relate categories to each other to expose the different relationships between theoretical constructs. This should be done through the process of theoretical coding [13], or axial coding [31]. This guideline was followed by writing theoretical memos as the analysis progressed and the memos were in turn used to generate theoretical codes used for coding the data.

Theoretical sampling stresses the importance of deciding on analytical grounds where to sample from as the research progresses [9]. This approach helps saturate the categories of the emerging theory and ensures that the theory is actually grounded in the data [30]. This guideline had a significant impact on the research as agreements with interviewees and consulting companies could not be made prior to initiating the research study but had to be established as the data analysis played out.

Scaling up proposes that the researchers group higher level concepts into broader themes to help escape the descriptive level of analysis and help contributing to the generalizability of the emerging theory. This process was aided by extensive use of theoretical memos and by iteratively visualizing the emerging theory through the use of diagrams in order to reach a substantive theory rather than mere description.

Theoretical integration calls for integration of the developed substantive theory with other theories in the same or similar fields in order to create a formal theory [14] that extends beyond the substantive area in which the theory originally emerged. In this study the substantive theory was related to other theories within and outside the IS field by comparing the grounded theory...
with theory addressing adoption of technology and diffusion of innovations.

2.4 Data Collection

Three types of data were collected and analyzed as part of the research: Documents; Observations; and Interviews. Documents, primarily from the vendor, were used in the beginning of the study for gaining background information about the new version and to gain insight into the documented differences between the old and the new version.

Two types of observations were made during the study. The first type consisted of participatory observations [2] where the observing researcher participated in three presentations and three workshops with consultants concerning the new version. The second type of observations came from in-depth experimenting with a demo version of the new version of the enterprise system, provided by the vendor.

The interviews were collected between December 2008 and March 2011. All interviews were semi-structured [22] with an interview guide relying on explorative and open-ended questions aimed at saturating the theoretical constructs as they emerged from the data. The initial interview guide was broad and explorative but as the research progressed the interview guides became more focused on saturating the emerging categories and thus varied from one interview to another.

A total of 15 interviews were carried out throughout the study. Two interviews with representatives from the vendor and an interview with an internal project manager at a customer that had implemented the new version were also conducted to both triangulate statements from the consultants and to saturate concepts and categories based on the principle of theoretical sampling. An overview of the conducted interviews is shown in Table 1. All interviews were recorded and fully transcribed to allow detailed coding of the data.

### Table 1: Participating companies in the study

<table>
<thead>
<tr>
<th>Company alias</th>
<th>No. of employees</th>
<th>Company type</th>
<th>Interviewee role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor</td>
<td>90000 global/1000 local</td>
<td>Vendor</td>
<td>Vendor Marketing Manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Partner Technology Advisor</td>
</tr>
<tr>
<td>Customer</td>
<td>75</td>
<td>Customer</td>
<td>Internal Project Manager</td>
</tr>
<tr>
<td>Partner 1</td>
<td>28</td>
<td>ISV + VAR</td>
<td>CIO</td>
</tr>
<tr>
<td>Partner 2</td>
<td>1100 global/250 local</td>
<td>VAR</td>
<td>Unit Manager</td>
</tr>
<tr>
<td>Partner 3</td>
<td>50</td>
<td>VAR</td>
<td>Consultant</td>
</tr>
<tr>
<td>Partner 4</td>
<td>14</td>
<td>VAR</td>
<td>Chief Consultant</td>
</tr>
<tr>
<td>Partner 5</td>
<td>1</td>
<td>VAR</td>
<td>CEO</td>
</tr>
<tr>
<td>Partner 6</td>
<td>39000 global/250 local</td>
<td>ISV + VAR</td>
<td>Product Manager</td>
</tr>
<tr>
<td>Partner 7</td>
<td>50</td>
<td>VAR</td>
<td>Chief Consultant</td>
</tr>
<tr>
<td>Partner 8</td>
<td>180</td>
<td>ISV + VAR</td>
<td>Consultant</td>
</tr>
<tr>
<td>Partner 9</td>
<td>1800 global/80 local</td>
<td>VAR</td>
<td>Product Manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Consultant</td>
</tr>
<tr>
<td>Partner 10</td>
<td>23</td>
<td>ISV</td>
<td>CEO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Product Manager</td>
</tr>
</tbody>
</table>

3. Data Analysis

Analysis of the data began after the first two interviews were conducted with the consultant in Partner 3 and the CEO of Partner 10. The interviews were analyzed using open, axial, selective coding [31]. Open coding consisted of conceptualizing the text in the transcripts of the interviews on a line-by-line basis. The process was aided by the use of the Atlas.ti software [26] and the initial open coding resulted in approx. 41 concepts in three categories. The process proceeded to the phase of axial coding in which the categories and concepts were related to each other. Finally the phase of selective coding entailed the selection of a single core category to which all other categories and concepts were related. At this point the categories were far from saturated and many new questions arose. The collection and analysis of the remaining 13 interviews focused on saturating and extending the concepts and categories by selecting companies and interviewees based on the guideline of theoretical sampling. A non-sequential iteration of open, axial, and selective coding continued through the remaining analysis and by the end of the final iteration of coding a total of 1080 instances of data had been coded and 93 theoretical memos had been written through the coding process. Table 2 shows an excerpt of the categories, codes, and examples of quotes that led to the conceptualization. The complete list is available from the authors.
4. Emerging Explanatory Model

The grounded theory emerging from the analysis of the study revolves around the core category of “version transitioning” (the process of changing from an installed version to a new version). The transition process that emerged from the interviews with the vendor, VARs, ISVs and customer is depicted in Figure 2 and 3. The figures show the categories emerging through the analysis of the study and how they interact with each other.

The theory depicts the process of transitioning from an old to a new/upgraded version of a packaged enterprise system from the perspective of the two types of partner companies in the software ecosystem. The presented categories and concepts are not proposed as being exhaustive and only the most central and saturated of the concepts are presented.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Concepts</th>
<th>Examples of data from the study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategizing</td>
<td>Understanding of the new version</td>
<td>“It is seriously a different way of thinking” (Product Manager – Partner 10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“You have to understand the concept of [the new version] to see the point” (CIO – Partner 1)</td>
</tr>
<tr>
<td></td>
<td>Comparing benefits and shortcomings of the new version</td>
<td>“Much of the key functionality from [the old version] was not there” (Product Manager – Partner 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“[The new reporting tool] has some tools that are much smarter than the old reports” (Consultant – Partner 3)</td>
</tr>
<tr>
<td></td>
<td>Availability of add-ons</td>
<td>“One of the very important factors in this has been that some of the add-ons we always implement when we sell a solution have not been ready for the new version.” (Product Manager – Partner 6)</td>
</tr>
<tr>
<td></td>
<td>Experience with the new version</td>
<td>“I only have experience from one implementation” (Consultant – Partner 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“It was very new to me” (Chief Consultant – Partner 7)</td>
</tr>
<tr>
<td></td>
<td>Experience with the old version</td>
<td>“[…] and I had much experience with the old version […]” (Consultant – Partner 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“[…] the classic version that we are used to […]” (Product Manager – Partner 6)</td>
</tr>
<tr>
<td>Pushing</td>
<td>Pushing the new version</td>
<td>“So we asked [the customer] if they felt like trying out [the new version]” (Consultant – Partner 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“[…] and that convinced them” (Unit Manager – Partner 2)</td>
</tr>
<tr>
<td></td>
<td>Pushing the old version</td>
<td>“The are many that offer the old version” (Product Manager – Partner 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“[The new version] was not interesting for us to try to push […]” (Consultant – Partner 8)</td>
</tr>
</tbody>
</table>
Figure 2: Transition process of the ISVs

Figure 3: Transition process of the VARs
5. Discussion of the Findings

The following sections describe the findings from the data analysis.

5.1 Transition process: Independent Software Vendors (ISVs)

The first stage in the transition process of the ISVs is “strategizing”. At this stage the ISVs form a strategy on which add-ons they should upgrade to fit the new version. The strategy is based on understanding the new version comparing the benefits and shortcomings with the old version, and on the demand for upgraded add-ons. Following the strategizing stage the ISVs move to the stage of “upgrading”. Based on the strategy for upgrading, the ISVs either upgrade an add-on to fit the new version or leave it to only be compatible with the old version.

Finally, the ISVs move to the phase of “selling” in which they either sell an upgraded add-on or an add-on that is only compatible with the old version. Following the selling stage the process returns to the stage of strategizing.

5.2 Transition process: Value Added Resellers (VARs)

Similar to the transition process of the ISVs, the transition process of the VARs begins with the stage of strategizing. At this stage the VARs form a strategy on whether to push the new or the old version to customers.

The strategy is based on the understanding of the new version and the benefits and shortcomings compared to the old version – similar to the strategizing process of the ISVs. However, as the VARs take on the actual implementation at the customer organizations, the strategy of the VARs is also dependent on the availability of upgraded add-ons from the ISVs. Furthermore, the outcome of the strategizing phase of the VARs is based on their experience with implementing the new and the old version. At the early iterations of the transition process the experience with implement-
ing the new version is low, influencing the subsequent stage of pushing. At the pushing stage the VARs either try to push the new version or the old version when communicating with customers.

The outcome of the pushing stage is the stage of implementing in which the VARs either implement the new version or implement the old version at the customer. Although the VARs may have been pushing the new version, customer demand for the old version may result in the old version being implemented, and vice versa, as illustrated by the crossing paths in Figure 3. If the VARs implement the old version no experience is gained with the new version. However, when an implementation of the new version is carried out, the experience gained from the implementation influences the strategizing stage in the next iteration of the transition process which may cause a change in strategy on pushing the new or the old version to customers.

5.3 Transition process: Ecosystem

Scaling up the transition process to the level of the ecosystem as a whole provides a perspective of the influence between the actors in the ecosystem depicted in Figure 4.

The vendor exerts pressure on the ISVs to upgrade their add-ons to be compatible with the new version. In parallel the vendor also exercises pressure on the VARs to begin selling and implementing the new version. The ISVs and VARs in turn respond to the perceived shortcomings of the new version by demanding improvements of the core enterprise system package. As part of the transition process of the VARs, demand for upgraded add-ons is directed (backwards in the supply chain) towards the ISVs, influencing the transition process of the ISVs, as described previously. The outcome of the transition process of the ISVs is a supply of upgraded add-ons to complement the implementations carried out by the VARs. The influence between the VARs and the customers is characterized by the VARs pushing the new or the old version to the customers. The customers correspondingly create a pull for either the new or the old version of the enterprise system package.

6. Conclusions and Limitations

The grounded theory of the dissemination of upgrades at both the level of the individual actors in the ecosystem and the ecosystem as a whole provides insight into the transition process from an old to a new version of an enterprise system package. The theory casts light on some of the potential weaknesses of the dependence on ecosystems for diffusing a new version of a pre-packaged enterprise system from the vendor to the customer. The findings suggest that the supply of upgraded add-ons, delivered by the ISVs, may cause inertia in the diffusion of the new version as VARs are dependent on the add-ons when implementing the enterprise system at the customers. Correspondingly, the ISVs are dependent on a demand for upgraded add-ons from the VARs before they initiate the upgrade process.

The findings also suggest that even though the VARs try to push the new version to the customers, the customers may create a reciprocal pull for the old version of the enterprise system. As the VARs transition strategy is influenced by the amount of experience they have with implementing the new version, the pull for the old version from the customers may cause a barrier to beginning to push the new version. However, as customers may also pull for the new version when the VARs are trying the push the old version, the pull from customers can stimulate the transition process of the entire ecosystem. The findings are thus consistent with previous suggestions (e.g. [28]) that neither a technology-push nor a customer-pull perspective is sufficient for understanding diffusion of innovations. The research also supports the importance of addressing adoption and diffusion of innovations from a network perspective rather than looking at individual companies [1].

The grounded theory on the upgrade process indicates a gradual iterative nature of the process of adoption of innovations. This is somewhat understated in some of the previous adoption literature, which proposes a more linear process of adoption (e.g. [27]). Although the introduction of a new version of a packaged enterprise system in the ecosystem will eventually lead to the discontinuation of the old version, the process resembles more a transition rather than a sudden shift, and it aligns closer with the perspective that “as innovation develops and diffuses, learning occurs; the old and the new exist concurrently, and over time these are linked together” [5].

There is an inherent limitation of building theory from the dissemination of a single version in a single ecosystem so future research should look into the process of the diffusion of upgrades and adoption in other software ecosystems and preferably in other types of ecosystems to extend the currently proposed substantial theory into a formal theory.
7. References


