

The Birth of An E-Business System Architecture: Conflicts, Compromises, and Gaps in Methods

Kari Smolander

Lappeenranta University of Technology, Department of Information Technology, Telecom Business
Research Center (TBRC), P.O. Box 20, FIN-53851 Lappeenranta, Finland
E-mail: kari.smolander@lut.fi

Abstract

This paper describes the system architecture development process in an international ICT company, which is building a comprehensive e-business system for its customers. The implementation includes the integration of data and legacy systems from independent business units and the construction of a uniform web-based customer interface. The research focuses on the creation of e-business system architecture and observes that the architecture is not completely designed through rational decisions and trade-offs. Instead, it emerges through somewhat non-deliberate actions obliged by the situation and its constraints, conflicts, compromises, and political decisions. The interview-based qualitative data is analyzed using grounded theory and a general explanation model for the creation of e-business system architecture is introduced. The role of technical system architecture in e-business systems development is clarified and e-business specific requirements for development methods are listed and compared to the support provided by UML and RUP.

1. Introduction

Robust system architecture is considered as one of the key issues when building successful e-business systems [e.g. 1]. The design of system architecture is usually seen as a set of trade-offs between available resources (such as available personnel and money) and operational requirements related to technical architecture, such as scalability, capacity, response times, security, and availability [e.g. 3]. The software architecture research provides design tools for architecture design, including, for instance, architecture description languages [4, 5], common architectural patterns [6] and styles [7], architectural trade-off methods [8], component frameworks for business systems [9], and technologies for e-business implementation [10]. In an ideal world, the work of an architect would be to find out

the explicit requirements for architecture, select the best possible design tools and implementation technologies, make rational trade-offs concerning the requirements, and produce the best realistic solution for the architecture with the design tools and implementation technologies.

However, literature contains many examples of cases where this kind of technical rationality has not been sufficient for success in IS projects [11-13]. In directly systems or software architecture related research, some researchers have found that the work of an architect and the usage of architecture include also more diverse connotations than the technically oriented architecture research commonly perceives. These include for example the diverse tasks of an architect in an organization as observed by Grinter [14] and varying uses and meanings of architecture in practice [15, 16]. The message of these studies includes the facts that an architect has a social and even a political role in an organization and that different stakeholders relate different meanings to architecture to fulfill their informational requirements in the development process.

In this paper, we describe a process where e-business system architecture is being created. By e-business, we denote to a more comprehensive concept than e-commerce. In addition to e-commerce systems serving external customer transactions, e-business includes also integration and streamlining of internal information systems to serve the new digitally enabled business processes [17]. Laudon and Travers [18] define e-business only as the digital enablement of transactions and processes within a firm, but for the sake of simplicity, we understand e-business here to cover both the transactions and processes within a firm and the integrated external e-commerce systems as in [17]. This enables us to interpret the process in the studied organization as the process of building an integrated e-business system.

By systems architecture development, we mean a process where early design decisions are realized into an architecture defining system's composition from various viewpoints [cf. 19] and overall plans regarding system's

implementation from individual conceptual and physical components. Different stakeholders can relate their concerns to the issues made concrete by the system architecture and discuss their needs in concrete terms given by the architecture. They can also make decisions concerning system development strategies and policies using system architecture as a common reference. This conception sees system architecture not only as a technical artifact but also as a phenomenon having strong organizational connotations, as evidenced by [15, 20, 16].

We interviewed various stakeholders in an international ICT company, which was building a comprehensive e-business system for its customers. The company already had several e-commerce systems in individual business units. The e-business project included the integration of data and legacy systems from independent business units and the construction of a uniform web-based customer interface hiding most of the differences between the business units. We wished to find out what factors influenced the creation of e-business system architecture: was it designed purposefully by the involved architects through rational decisions and trade-offs or did it emerge through somewhat non-deliberate actions obliged by the situation and its constraints, conflicts, compromises, and even political decisions. Our aim was also to clarify the role of technical system architecture in the change required when implementing e-business systems and to provide requirements for the development of methods for e-business systems development. We analyzed the qualitative data from the interviews using grounded theory [21] as the research method and concluded the analysis by stating the implications on e-business development methods in real-life situations.

The paper is organized as follows: in Chapter 2, we describe the research process and method used. Chapter 3 describes the situation the company is facing and the motives for the change and implementation of the e-business system. In Chapter 4, we describe the situation and the context of the development project aiming at e-business implementation and continue with Chapter 5 to clarify the consequences of the situation to the development. From these observations, we derive a list of requirements for e-business development methods, which will be presented in Chapter 6. Finally, in Chapter 7 we end the paper with conclusions.

2. Research process

The studied organization is a globally operating ICT company having thousands of employees worldwide. Its customers include both consumers and businesses for which the organization provides various products and services. Software is one of the key assets in the organization's service production. Historically, the organization has had several independent business units targeted at di-

verging business sectors. In addition, the information management of the organization has been distributed to these business units and the functions of enterprise level information management have included mainly the provision of network infrastructure, enterprise level accounting, and basic office tools. Most of the operative information systems have been implemented and operated by the business units that have been quite independent in the decisions concerning strategies for information management. However, recent developments in markets and technology have led the organization to set its strategies to a more integrative direction. For this reason, the organization has set an objective to provide an integrated e-business solution to both its consumer and business customers. This will include both implementation of a uniform web-based customer interface and sufficient integration between the distributed operative back-end information systems, such as customer management and billing systems.

In the study, we made 12 interviews during January and February 2002 among the stakeholders of the ongoing e-business system architecture design project. The interviewees included five systems architects, three representatives from the project management, two representatives from information management, one requirements analyst, and one vice president closely involved in the project. The interviews lasted from 45 to 120 minutes and they were completely transcribed. The interviews were based on a theme list derived from the research questions. The emphasis of the themes varied according to the interviewee and his/her role in the project. The themes were also adapted during the process to the changing conceptions of the researchers to enable theoretical sensitivity [21].

The qualitative data obtained from the interviews was analyzed inductively using a grounded theory based approach [21, 22]. The analysis started with the open coding phase where mentions from the interviews were searched that could be interpreted as meaningful related to the research questions. The found mentions were categorized using a software tool ATLAS.ti. During the open coding phase, altogether 187 categories were found, and the categories were assigned to several super categories or category families, including for instance changes, conflicts, consequences, experiences, problems, purposes, and solutions emerging in the e-business system architecture design and implementation process. The axial coding phase, which was made partly parallel with the open coding, concentrated on finding the relationships (e.g. causes, is associated with, contradicts) between the categories. The axial coding phase produced in total 200 relationships between the categories.

Inductively, deriving from the data, we formed a general explanation model based on the causalities found in the e-business development process, which is shown in Figure 1. The organization is facing market changes and changing the organization according to the changing mar-

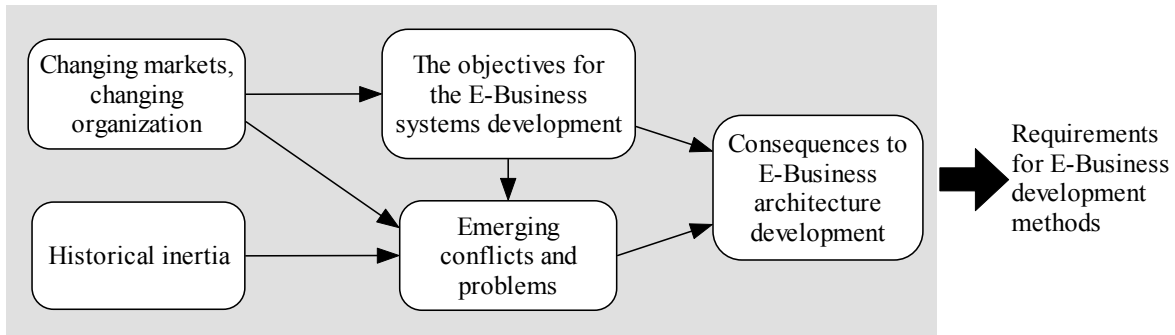


Figure 1. The explanation model

kets. The objectives for the e-business development emerge from these changes and because the change is continuous and it brings all the time new requirements for the e-business system, the objectives are quite fluctuating. In addition, the history and legacy structures of the organization cause conflicts and problems in the development when combined with the need for change. These fluctuating objectives and emerging conflicts and problems bring certain consequences to the e-business system architecture development in the organization. The formation and description of this causal model can be considered as selective coding [22] and its instantiation in the studied organization is explained in detail in the next three Chapters.

3. Starting point: changing markets, changing organization

There have been considerable changes in the ICT market during the last years and the organization under study has undergone a deep change. Few years ago, the strategies emphasized growth and utilization of the possibilities in the stock market. This enforced independent business units inside the organization since the growth was easier to handle through independency. Each of the business units built independent e-commerce solutions and customer extranets, which resulted to a fragmentary set of e-commerce solutions to customers. Each business unit had its own Internet sites, sales and billing systems, and web-based customer support, with diverging architectures and appearances.

When the beliefs in markets on the possibilities of ICT sector's continuing growth diminished, the organization had to change its strategies from growth to profitability and from stock market to customer orientation. This included both aims to reduce costs and concentration on the integration of the customership to enable evaluation of costs and profits associated with products and customer groups. With independent business units, there was no authority in the organization, which would see a customer as a whole. Instead, each business unit kept track of the

customers only in the context of its independent business. It was recognized that for the enterprise to see and evaluate a customer as a whole, a profound change to the way of building information systems was needed. This would include building of an integrated e-business solution instead of separate e-commerce systems. This change would also require changes in business practices and organization – the organization should operate in a more integrated fashion and the barriers between independent units should be lowered.

The organization began to see technical e-business system architecture as an enabling technology for the change. The IS organizations in independent business units were obliged to cooperate and enforce commitment to the integration of information systems. This also emphasized the role of central information management, which had been in a minor role this far. Its tasks had included mostly maintaining the common infrastructure, including corporate networks, electronic mail, office tools, and enterprise-level accounting. Now its roles would include the enforcement of information systems integration and enabling technically the unification of the sales channels and customer management for the planned e-business solution. At this point, the organization decided to establish a working group of systems architects from various parts of the organization. The objective of the group became to design the e-business system architecture supporting the needed integration. In the following chapter, we shall describe the context and the forces under which this group of architects are developing and designing the architecture.

4. Development context: conflicts, problems and varying purposes

The context for e-business system architecture development included many issues, which the working group for system architecture development had to face and be aware of. These included:

- Market changes and their effects on the organization and on the objectives of the e-business (as described above).
- The historical inertia caused by the independency of the businesses and their independent information management policies, architectures, and legacy systems.
- Fluctuating requirements and objectives for the e-business development caused by changing markets, changing organization, and different conceptions in independent business units.
- Emerging conflicts and problems arising from the market and organization changes, historical inertia, and moving objectives.

In the following sections, we will clarify these issues and forces affected on the work of the architecture development group.

4.1. Historical inertia

The organization's history with independent businesses and their diverging functions and objectives had both psychological and technical consequences causing slow progress and conflicts in the integrated e-business development. Each of the business units had varying legacy systems with different information structures, technical architectures, and operating principles. It was not possible in practice to replace the operative legacy systems with a uniform solution at once. Some of the systems had been under construction and renovation for more than ten years and they formed the very basis for processes in the business units. This far, the development efforts had mostly focused on the systems of individual businesses and therefore their areas have overlapped and even included competitive goals from the viewpoint of the whole enterprise.

The historical inertia had effects also on the organization responsible for information management and information systems. Because of the independence, the organization had no clear role of central information management that could take responsibility of the e-business system architecture development. The role of the enterprise level information management had been a minor one, whereas the roles of individual information managements in business units had historically been more central in the development of systems. The transition to an integrated e-business system requires a central authority that can take the responsibility and such organization was in the initial state missing. Many of the conflicts and problems described later arose from this situation.

4.2. The observed objectives for the e-business system

The fluctuating objectives, meanings, and requirements for the e-business system architecture created another

source of conflicts and problems. In a large organization with a high degree of independency, the conceptions among different business units and individuals about the purposes of an e-business solution vary considerably. Among the interviewees, we identified a large set of different purposes for the e-business system, which were then classified in five distinct classes:

- *Creation of a unified electronic customer interface.* This included an objective to make the company appear uniformly regardless of the business area in question. Technically, this would require at least the creation of a uniform web site with a single sign-on procedure for the users and the integration of existing customer extranets to the unified interface. More general objectives related to this class included the opening of a new channel for customers and the clarification of the relationships between the company and its customers.
- *Reduction of costs.* Some of the interviewees were optimistic on reducing the IT costs through the integrated e-business system under construction. This conception was based on the application of customer self-service and on the reduction in the number of customer-oriented information systems. The system would reduce the amount of needed direct customer support through a planned self-service functionality and the number of systems related to customers would diminish because of the integration of customer information.
- *Integration of information systems.* In addition to the integration of customer information, the project was also seen as a means to implement more general enterprise integration solution, including the integration of product, contract, and sales information. This would also mean the rationalization of used technology and the rationalization of systems development processes in the organization.
- *Gaining business advantage.* The integration involved in the project was seen also as a rationalization of the business. The uniform and understandable customer interface with a faster and more comprehensive customer service than before would bring competitive advantage to the organization over its competitors who do not currently have such a uniform customer interface.
- *Implementing an organization change.* Surprisingly many interviewees saw this project also as a means to implement and enforce a new organization structure. Through the project and the integration it implements, the organization would get some of its concrete features and also the technical people participating in the project would become more committed to the organization change.

This list of observed purposes for the e-business system looks quite comprehensive and ambitious. Different interviewees emphasized the purposes differently and many saw that the only realistic objective was to implement a

single sign-on procedure with a minimal level of customer information integration. The list anyhow shows the complicated and conflicting nature of objectives for the e-business system when it is developed for a large enterprise.

4.3. Emerging conflicts and problems

Changes in markets and organization, the history of the organization, and the complicated objectives for the e-business system put the architecture development group in a situation where a number of conflicts and problems emerged. The architecture development group and its members were obliged to respond to the emerging conflicts and problems by some means and these responses shaped the development process to a quite deterministic direction where the role of deliberate design mitigated. In open coding, we identified in total 48 categories of conflicts and problems, of which we present here in a condensed and combined format those that appeared most essential in our analysis:

- *Varying requirements and unclear objectives.* As described above, the purposes of the system were manifold and complicated and the requirements varied according to the business needs in the business units. The architects held this unclarity of objectives and requirements as the biggest obstacle for the development. Those in the managerial level recognized the problem as well, but explained it as unavoidable in the situation and expected that the first prototypes of the system will bring more clarity to the objectives. This resembles the chicken-egg problem: architects must know well the objectives to design the system architecture, but the objectives are further clarified only after the first version of the architecture is built.
 - *Unclear project organization and organizational responsibilities.* The transition to the integrated e-business necessitates an organizational change away from the independent units to a more integrated organization and business practices. This caused fuzziness within the project too. Especially the architects had trouble in discerning the project organization and the changing organizational responsibilities related to it.
 - *Problems in the cooperation between technical and business people.* Architects expected the business responsables to explicate clear requirements and objectives for the system and its architecture. However, they also considered the task impossible, because they thought that the business responsables do not possess enough understanding about the possibilities of current technology. They felt that this leads to unrealistic objectives, which were manifested especially when considering the possibilities of legacy systems integration. Those with business background had more optimistic
- views on the possibilities for the integration of legacy systems than architects had.
- *Getting and preserving commitment.* To get and preserve commitment to the project formed one of the drivers for the development. The historically independent business units needed observable benefits to stay committed to the effort and the project management recognized the need for quick and visible results. As mentioned above, the requirements and objectives included much variation between the business units and this led to certain suspicion between the business units concerning the architectural decisions. In order to preserve their commitment to the effort, the architects from the business units were also supervising the interests of their businesses in the development process.
 - *Conflict avoidance and problems in decision-making.* Again, because of the history of independency, a central authority that could take care of the architectural decisions for the integrated e-business solution was missing. This led to the avoidance of conflicts and enforced the tendency towards compromises. A frequently occurring phrase among the architects included the term “lowest common denominator”, which was usually noting to the compromised solution with a single sign-on procedure and a minimal level of customer information integration.
 - *Conflict between schedule and expectations.* The organization’s internal reporting was tied to the quarters of the year. To preserve commitment and funding, the project had to show visible and beneficial results after each quarter. This leads easily to quick-and-dirty and architecturally unsound solutions that may fail to meet the expectations.
 - *Role of central information management and missing working practices.* The independency of businesses and the minor role of central information management had implications on the working practices. The architectural and development practices of the business units contained considerable differences implying that also common working practices needed to be established for the development process of the e-business system.
 - *Difficulties in creating common understanding about the architecture.* Since the business units have had their own histories and produced their own legacy systems and information architectures, the interpretations on the situation and objectives diverged. This, combined with changing organization, unclear objectives, and missing common working practices, created difficulties in understanding and transferring architectural knowledge between the participants from different business units.
 - *Integration level determination.* The ownership of the information becomes an issue even in the most modest single sign-on e-business solution serving the whole organization. The question becomes “who owns the

customer information?” and relates to determining the integration level to the currently independent back-end legacy systems. The more ambitious integration, the more out-of-control the customer information (and possibly other information too) shifts from the business units. This was another source of mistrust from the point of view of individual business units.

- *Implementation of integration.* In addition to determining the integration level, the actual implementation of integration may prove problematic. Since the diverging legacy systems could not be replaced, they all had to be interfaced. According to one interviewee, this meant a change from web surfing to transaction-based e-business, which may bring issues with e.g. security and undeniability of transactions.

Of these conflicts and problems occurring when creating e-business system architecture, only the last one was mainly a technical problem. The others were more related to the change in organization and practices that happen when developing an e-business system in a large organization with independent businesses. In the following, we shall look closer on what consequences these conflicts and problems caused for the architecture design and development process.

5. Consequences: limited designs and minimal solutions

The abovementioned conflicts and problems led to some unfavorable consequences. While it was of paramount importance for the company to be able to streamline its systems and develop a more coherent architecture enabling the creation of an e-business system, the realities of legacy systems and the organization led to situation where it was best to seek satisfying, even minimal, solutions instead of optimal ones.

Even if architectures are seen more as general blueprints or roadmaps, evolutionary prototyping was considered the only possibility in the system architecture development. Because the schedule was tight, the objectives and requirements unclear and changing, and because the business units were rather independent, it was hard to achieve common understanding and commitment. Therefore, evolutionary prototyping was seen almost as the only possible principle for the development. With prototyping, it would be possible to clarify objectives and commit stakeholders by showing them visible results from which they can benefit. This could however lead to new problems: combined with the quarter-based reporting system in the organization, evolutionary prototyping easily produces quick-and-dirty and ad hoc solutions. We could classify the interviewees to those with positive attitudes towards prototyping and to those with negative or doubtful attitudes. In general, the project management believed positively that

“somehow” the prototypes would transform to the final e-business solution, whereas technical architects presented more doubts and wanted to have explicit requirements and objective statements before committing to certain architectural solutions.

In addition to inevitable evolutionary prototyping, we found a strong tendency towards minimal instead of optimal solutions. The history of independency between the business units created trouble in determining the level of integration, e.g. what parts of customer information and functionality related to customers (and other information too) are common and what parts belong within the control of the business units. In addition, the independency with a desire to avoid conflicts in the decisions concerning the system architecture may lead to a solution containing nothing more than the functionality that is absolutely necessary and agreeable to all parties. Almost all interviewees named the single sign-on functionality with a minimal level of customer information integration as such a minimal solution, and many interviewees stated that this would probably form the main result of the project. This kind of a solution, however, meets only a small portion of the objectives named for the project.

Not only the results but also the designs may become severely limited. The market and organization changes combined with the history of independent businesses have created many constraints and obstacles to the system architecture design. Existing legacy systems, the evolutionary approach, varying requirements, unclear objectives, difficulties in creating common understanding, and problems in decision making create a complex situation where rational system architecture design, as it is conceived in the literature, has no possibilities for immediate success. The degrees of freedom of design become limited: the system and its architecture cannot be designed as a whole, but rather one need to accept the conditions and limitations caused by factors above and to keep the day to day operations running while the new systems are continuously created through evolution.

The situation builds also organizational consequences. We found clear hints of low-level networking and formation of shadow organizations as the result of unclear project organization and problems of decision-making and objective setting. As the organization and responsibilities change, new and perhaps inexperienced persons come into crucial official positions related to the e-business system development. At the same time, the experienced architects and other key persons continued to stay in contact with each other. This unofficial shadow organization balanced the mismatch in skills and experience that might otherwise seriously impede the development.

The final consequence from all the above becomes that in fact, the e-business system architecture becomes emergent: it is created gradually through compromises, constraints, and conflicts. The exact objectives and responsi-

bilities will be resolved as the architecture emerges through evolutionary prototyping. Compared to the conventional view on architecture design [23, 24], most of the claimed benefits of rigorous architecture development seem to be lost. There is no “grand plan” since the work is proceeding in a day-to-day basis and the well defined responses and interfaces between systems do not necessarily emerge in a rationally planned way, but rather most duplicate functions are kept and there is agreement only on a few items that become the “architecture”.

6. Derived requirements for e-business systems development methods

From abovementioned observations and explanations, we can derive a set of requirements that an e-business systems development method should meet. Based on this study, we can claim that the method requirements mentioned below will occur when aiming at e-business integration in an organization with several business areas. However, the situation in particular organizations will vary and therefore the emphasis and priorities between these requirements will vary too.

In the following, we shall extract from the case the general requirements for e-business development methods. We found thirteen general requirements that were specific to e-business development. The requirements are first described in text and a link (R1-R13) will be made to Table 1 that summarizes the requirements as a list.

The first requirement comes from the overall conclusion of the case. The transition from heterogeneous E-commerce to integrated e-business is not only technically challenging: it is more a profound change to the organization. In fact, the primary challenge is in the change of the organization, not in the implementation of the technology. Therefore, e-business systems development methods should support also the description of organizational change (R1).

In this change of organization and implementation of technology, the role of central information management or some kind of central authority in the organization is crucial. The central authority should take care of the multitude of conflicts occurring when aiming at integration and coordinate the creation of objectives for the system. An e-business development method should enable the creation of a common vision (R2), which is then enforced by the central authority.

An integrated e-business system architecture necessitates the integration of information systems in the organization and the rationalization of technology and development processes. Existing legacy systems will be integrated to the e-business functionality. This requires the selection of an integrative technology and the construction of development processes supporting the implementation of the integration. Because the integration is the basis and char-

acteristic to e-business development, the development method should have specialized and usable techniques for describing information systems integration (R3).

The chicken-egg problem between objectives and architecture becomes problematic in e-business development. To design robust system architecture, one must have clear objectives, and to select realistic objectives, one must understand the possibilities of the system architecture. To overcome this problem, it is necessary to have a close cooperation between technical architects and business responsables. This, however, induces a language problem. These groups do not have a common language and their world of experience does not have a large overlap. To overcome the language problem, we need architecture description languages that business responsables understand (R4) and business descriptions that are explicit enough for technical people (R5). This enables realistic views on the possibilities to implement e-business systems.

Evolution with modest but growing objectives may be the only way to develop integrated e-business systems. To foster commitment, some immediate benefits should be shown with the prototypes for each stakeholder. However, at the same time, the path to robust system architecture should also be secured and enough time and resources must be given to technical architects. This very difficult and complex trade-off must be made in every e-business project (R6).

The implementation of e-business integration deals not only with technical issues but also with difficult political ones. An organization shifting to integrated e-business must resolve issues concerning the internal ownership of information related for instance to customers, sales, contracts, and products. The ownership and responsibilities related to information must be decided and described during the development process. The development methods should include means for such descriptions (R7).

The technical requirements of e-business development methods do not differ much from those of methods for traditional transaction-based information systems. E-business system development includes methodical requirements concerning e.g. distribution, error recovery, and networking, but those requirements can be met without a special “e-business support”. A standard way to describe such technical solutions is of course required (R8).

The key issue in the development of e-business systems is the keeping of the day-to-day operations running and at the same time implementing the integration between existing legacy systems and the new e-business functionality. This means that the nature of development is in many cases more analogous to a maintenance project than to a green-field development project. Current systems development methods and models of thought are mostly aimed at designing new systems instead of changing existing ones. This problem is already recognized before the advent

of e-business [e.g. 25], but it becomes more critical in the e-business development. From this we can derive a requirement that the development method for e-business systems should support evolutionary approaches to architectures and systems (R9).

Identifying and agreeing about objectives became the most difficult problem in this case. When an organization wants to integrate its otherwise independent businesses

into a unified e-business solution, the problems become probably quite similar. To become valuable in practice, e-business development methods should support not only the formation and recording of objectives but also measuring of success related to objectives to enable successful development (R10).

The problems of objectives and integration culminate on system architecture design because the designs and

Table 1. Summary of the requirements for e-business development methods

	Requirement	Rationale	Support in UML and RUP
R1	Support for the description of organizational change	e-business involves deep changes to organization	Poor; some thoughts of “organization engineering” in RUP’s Business Architecture
R2	Support for the description of a common vision	Resolve conflicts, build objectives	Poor; no common language for all stakeholders
R3	Specialized techniques for describing the information systems integration	IS integration is characteristic to e-business development	Poor; no specialized technique for the description of integration
R4	Architectural description languages that business responsables understand	To enable realistic objective selection, business responsables must have some understanding on architecture	Poor; the descriptions necessitate too much technical skills and knowledge
R5	Business descriptions that are explicit enough for technical people	To understand the objectives, technical people must have understanding on business	Poor; no description techniques showing overall aggregate view
R6	Both prototyping and careful architecture design needed	Gain commitment and resolve objectives through prototyping, aim at robust architecture	Mediocre; iterative basis in RUP, but its implementation is difficult in practice [2]
R7	Method should contain descriptions for organizational responsibilities and ownership of information	The ownership of information becomes an issue when aiming at e-business integration	Poor; only general thoughts
R8	Technical issues (like distribution, error recovery, and networking) must be described in a standard way.	These issues will occur as in all modern systems development	Good; this is what UML and RUP are for
R9	The development method should support evolutionary approaches to architectures and systems.	The change and maintenance of existing systems forms a major part of the e-business systems development	Poor; UML and RUP are mainly targeted at the development of new systems
R10	e-business development methods should support the formation and recording of objectives and measuring of success related to objectives	Identifying and agreeing about objectives is one of the most difficult issues in e-business development	Poor; the objectives are mostly supposed to be given to the development project
R11	The methods should take the development to a very concrete level (both politically and technically) soon after the project initiation	The more architecture becomes concrete, the more stakeholders become aware of the consequences, conflicts, and problems	Good (technically), none (politically)
R12	The architecture designs and descriptions (and their implications) should be approachable and intelligible by the various stakeholders participating the process	To enable wide understanding to the consequences of architectural selections (cf. R4).	Poor; no relevant description technique besides Use Case diagrams
R13	The development process should support organizationally both effective control structures and flexibility	Strong authority is needed to handle the conflicts and unofficial structures for creative solutions	Poor; development organization “design” in a general level

prototypes related to system architecture become the first concrete artifacts in the development showing concrete implications to businesses and to the information management. In the process before system architecture design, the plans and designs have been on the “PowerPoint presentation” level, showing ambiguous and general roadmaps and noble objectives. The more concrete the architecture becomes, the more various stakeholders become aware of the consequences, conflicts, and problems they will be facing. This leads to two distinct requirements for the development methods: the methods should take the development to a very concrete level (both politically and technically) very soon after the project initiation (R11) and the architecture designs and descriptions (and their implications) should be approachable and intelligible by the various stakeholders participating the process (R12).

The requirements concerning an e-business development organization are quite conflicting. On the one hand, the development requires a strong authority that can control the process through conflicts, and on the other hand, forming of unofficial and shadow organization (peer-level networking) should be fostered to allow creative solutions and frictionless cooperation between businesses (R13). This requirement is, however, not a new one when developing organizations [c.f. 26].

Table 1 summarizes these derived requirements for e-business development methods. The last column in the table (“Support in UML and RUP”) analyzes how Unified Modeling Language [27] and the Unified Process [28, 29] support the e-business specific characteristics of the development process. This is important, because UML and RUP together form the current methodological basis for many software organizations. The column shows that the support is generally poor. The e-business specific requirements are not met by UML and RUP – only the standard technical issues are well covered. This conclusion calls for method development supporting better these e-business specific requirements.

7. Conclusions

We have described a process where a large ICT company is developing system architecture for a comprehensive e-business system. The external situation, changing markets, caused motivation to build an integrated e-business system, but left the objectives of the system unclear. The history of the organization combined with changes in markets and organization, and unclear objectives, created a large set of conflicts and problems that the architecture development group had to face. These led to several consequences: evolutionary prototyping was considered as the only possible principle for the development, minimal instead of optimal solutions predominated, and the possibilities of design became limited. In other words,

e-business system architecture became emergent. The exact objectives and responsibilities will be resolved as the architecture emerges through evolutionary prototyping.

From the case, we extracted 13 general requirements for methods supporting integrated e-business systems development. We also compared the requirements to the support that UML and RUP offers and concluded that the e-business specific requirements are not met in UML and RUP. Successful e-business development requires alternative approaches that support better organization change, communication between stakeholders, systems integration, objective formation, and evolutionary development.

In our study, system architecture manifested itself as a catalyst that makes business and organizational conflicts and problems concrete. When making decisions about architecture, the systems architects had to consider the organizational situation in the company. The architects also realized that technical rationality is not enough for success in this kind of a situation. To succeed in e-business architecture development, one has to be aware of the political and organizational forces that are driving the development and its objectives. E-business architecture development can therefore be characterized as a process of seeking boundaries, finding consensus, and identifying commonalities across organizational borders. Most previous literature on architectural methods has neglected this and sought to develop description languages for describing the actual architectures for systems with clear problem statements, whereas we claim that it would be more important to seek tools that aid in building common understanding about the system and its architecture and tools for resolving the emerging conflicts. Thus, we maintain that the field of architecture for e-business would benefit more from methodology and tools that help to identify and process the emerging conflicts than tools that aid in developing a technically “perfect” and optimized solution.

In the future, the research should provide tools and methods that are more appropriate for e-business systems development and take seriously the requirements listed here. Our objective is to follow this studied process of e-business system development also longitudinally to achieve a full picture of an e-business implementation process. Through these experiences, we wish to formulate a full model of the forces affecting on e-business implementation and to provide a full list of requirements for successful and in practice usable e-business development methods.

References

- [1] P. Fingar, "Component-Based Framework for E-Commerce," *Communications of the ACM*, vol. 43, no. 10, 2000, pp. 61-66.
- [2] W. Hesse, "Dinosaur meets Archaeopteryx? Seven Theses on Rational's Unified Process (RUP)," *Proceedings of the Sixth CAiSE/IFIP8.1 International Workshop on Evaluation of Model-*

- ing Methods in Systems Analysis and Design (EMMSAD'01), Interlaken, Switzerland, 4-5 June 2001, 2001.
- [3] S. Koushik and P. Joodi, "E-Business Architecture Design Issues," *IT Professional*, vol. 2, no. 3, 2000, pp. 38-43.
- [4] N. Medvidovic and R. N. Taylor, "A Classification and Comparison Framework for Software Architecture Description Languages," *IEEE Transactions on Software Engineering*, vol. 26, no. 1, 2000, pp. 70-93.
- [5] J. Conallen, "Modeling Web Application Architectures with UML," *Communications of the ACM*, vol. 42, no. 10, 1999, pp. 63-70.
- [6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*: John Wiley & Sons, 1996.
- [7] R. T. Monroe, A. Kompanek, R. Melton, and D. Garlan, "Architectural styles, design patterns, and objects," *IEEE Software*, vol. 14, no. 1, 1997, pp. 43-52.
- [8] R. Kazman, M. Barbacci, M. Klein, S. J. Carrière, and S. G. Woods, "Experience with Performing Architecture Tradeoff Analysis," *Proceedings of the 1999 International Conference on Software Engineering*, 1999, pp. 54-63.
- [9] P. Herzum and O. Sims, *Business Component Factory - A Comprehensive Overview of Component-Based Development for the Enterprise*: John Wiley & Sons, 2000.
- [10] M. Bichler, A. Segev, and J. L. Zhao, "Component-based E-Commerce: Assessment of Current Practices and Future Directions," *SIGMOD Record*, vol. 27, no. 4, 1998, pp. 7-14.
- [11] H. Drummond, "Riding A Tiger: Some Lessons of Taurus," *Management Decision*, vol. 36, no. 3, 1998, pp. 141-146.
- [12] C. Sauer, G. Southon, and C. N. G. Dampney, "Fit, Failure, and The House of Horrors: toward A Configurational Theory of IS Project Failure," *Proceedings of the eighteenth international conference on Information systems*, Atlanta, Georgia, USA, 1997, pp. 349-366.
- [13] E. Oz, "When Professional Standards are Lax: The CONFIRM Failure and its Lessons," *Communications of the ACM*, vol. 37, no. 10, 1994, pp. 29-36.
- [14] R. E. Grinter, "Systems Architecture: Product Designing and Social Engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 24, no. 2, 1999, pp. 11-18.
- [15] K. Smolander and T. Päiväranta, "Describing and Communicating Software Architecture in Practice: Observations on Stakeholders and Rationale," *Proceedings of CAiSE'02 - The Fourteenth International Conference on Advanced Information Systems Engineering*, Toronto, Canada, May 27 - 31, 2002,, 2002, pp. 117-133.
- [16] K. Smolander, "Four Metaphors of Architecture in Software Organizations: Finding out The Meaning of Architecture in Practice," *2002 International Symposium on Empirical Software Engineering (ISESE 2002)*, Nara, Japan, October 3-4 2002, 2002.
- [17] R. Kalakota and M. Robinson, *e-Business 2.0: Roadmap for Success*: Addison-Wesley, 2001.
- [18] K. C. Laudon and C. G. Traver, *E-commerce: Business, Technology, Society*: Addison-Wesley, 2002.
- [19] IEEE, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE, IEEE Std 1471-2000, 2000.
- [20] K. Smolander and T. Päiväranta, "Practical Rationale for Describing Software Architecture: Beyond Programming-in-the-Large," *Software Architecture: System Design, Development and Maintenance - IFIP 17th World Computer Congress - TC2 Stream / 3rd Working IEEE/IFIP Conference on Software Architecture (WICSA3)*, Aug 25-30, 2002 Montréal, Québec, Canada, 2002, pp. 113-126.
- [21] B. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chigago: Aldine, 1967.
- [22] A. L. Strauss and J. Corbin, *Basics of Qualitative Research: Grounded Theory Procedures and Applications*. Newbury Park, CA: Sage Publications, 1990.
- [23] M. Shaw and D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*: Prentice Hall, 1996.
- [24] C. Hofmeister, R. Nord, and D. Soni, *Applied Software Architecture*. Reading, MA: Addison-Wesley, 1999.
- [25] G. Forte and R. J. Norman, "A Self-Assessment by The Software Engineering Community," *Communications of the ACM*, vol. 35, no. 4, 1992, pp. 28-32.
- [26] H. A. Simon, *Administrative Behaviour: A Study of Decision-Making Processes in Administrative Organization, Fourth Edition*. New York: The Free Press, 1997.
- [27] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*: Addison-Wesley, 1999.
- [28] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. New York: Addison-Wesley, 1999.
- [29] Rational Software Corporation (2001). *Rational Unified Process*. Online documentation, Version 2001A.04.00.