Understanding and Managing Process Interaction in IS Development Projects

Bendik Bygstad[1] and Peter Axel Nielsen[2,3]

[1] Norwegian School of IT, Oslo, Norway
[2] Research Centre for Socio-Interactive Design Aalborg University, Denmark
[3] Department of Information Systems, University of Agder, Norway
bygben@nith.no, pan@cs.aau.dk

Abstract. Software-based information systems must be developed and implemented as a part of business change. This is a major challenge, since business change and the development of software-based information systems usually are performed in separate processes. Thus, there is a need to understand and manage the relationship between these two kinds of processes. In this paper we draw on a longitudinal case study. We suggest a framework to analyze the case as interaction between software development processes and organizational change processes. In the analysis we find that the framework enables us to understand critical events in the case, what led to the events, and what the consequences are. We discuss the implications for information systems research and in particular we discuss the contribution to project management of iterative and incremental software development.

Keywords: process interaction, project management, iterative software development, longitudinal process research.

1 Introduction

This paper is about the necessary interaction between two processes; the software development process and planned organizational change. The challenge that comes from the need to align IT and business is not new, but has consistently been at the top five concerns for CIO for the last decade [1-3].

We find that this also has important bearings on how software-based information systems should be developed. There is indication that IS project managers are facing several new challenges; to name a few: (1) The speed of change, driven by globalization, demands that information systems should be delivered in parallel with business change. Often it is no longer an option for the organization to "wait" while a new system is developed [4], and IT is expected to contribute to organizational agility [5] rather than hinder it. (2) The power balance between the organization and the IT departments has changed. Instead of humble users, the IS and software project managers meet powerful organizational actors who are well aware of IS failures and are inquisitive of the value of IT investments [6].

In this paper we are trying to make sense of a large development project where the project managers faced similar challenges. The existing literature is valuable in

C. Keller et al. (Eds.): SCIS 2012, LNBIP 124, pp. 25–43, 2012.

explaining the case study, but it also does not explain the core of the problems facing the project manager, namely, how to organize the interaction between the software development and the business change. We suggest in this paper a process view that is intended to provide a perspective on the interaction between software development processes and organizational change processes; or *process interaction* for short. By taking a process view we have a particular focus on what has happened in terms of events, their antecedents and implications [7]. We also bringing to the foreground the *development* processes and leave the issues in the background concerning how software-based information systems may be used in an organization and hence influence that organization.

Altogether, we are in this paper addressing the following two research questions:

- How can we *understand* process interaction, i.e., interaction between software development processes and organizational change processes?
- How can a software project manager of iterative and incremental software development processes *manage* process interaction?

Throughout this paper we take the standpoint of the project manager of software development. The software project manager has roles and responsibilities that are significantly different from managers of organizational change or general managers of the business organization.

2 Software Development and Organizational Change

In this section we provide a brief overview of research that is relevant to the research questions. The scope of the review is that of software *development* and its relationship with organizational *change*. Outside this scope is thus research that addresses information systems and their relationships with organizations or otherwise does not pertain to a development perspective.

We have identified seven related, but different research stream: IS development project management, socio-technical IS development methodologies, Scandinavian systems development, IS implementation, business process innovation, enterprise architecture, and iterative software engineering. For each research stream we present its main concern as well as how it contributes to the issues raised in this paper. Table 1 provides an overview of the literature.

IS Project Management: Normative IS project management research has for a long time addressed organizational issues like business alignment, risk management, and stakeholder analysis [9]. This research tends to be concerned with control. Common issues are: managing the systems life cycle, estimation, modeling, quality, scheduling, and cost. It is hardly surprising that control has become a common denominator given the turbulent history of software project failures. On the other hand, much of the normative software project research gives the impression that software projects are standalone projects starting from scratch.

Table 1. Research streams relevant to process interaction

Research stream	**Main concern**	**Contribution to process interaction**
IS project management [8-10]	Controlling the project and its environment's influence on its conditions. Projects should be aligned with business strategy.	The IS project manager has co-responsibility for value produced by the information system.
Socio-technical ISD methodologies [11-13]	Technical development should be part of organizational and human development.	An information system design cannot be separated from the organization design, and it should be developed in an integrated process.
Scandinavian ISD [14-16]	Systems development should be a part of organizational change, and with strong user participation.	The IS project manager should be a change agent for the organizational use of the information system.
IS implementation [17-19]	Mutual adaptation between the organization and the technology is necessary for a good solution.	Both work processes and technology may be modified in a dynamic and emergent process
Business process innovation [20]	Businesses should be organized as processes, not functions.	IT is an enabler of business process innovation.
Enterprise architecture and business process management [21]	Business aims, processes and IT solutions should be treated as an integrated whole.	A modeling approach is useful for integration and better communication.
Iterative software engineering [22-24]	Iterative and incremental development reduces technical and organizational risks.	Developers and business people should work very closely to produce useful solutions.

A recent contribution is the notion of value management, which aims to identify and manage business value in addition to cost in software projects [8, 9]. Value management is based on stakeholder analysis, and thus expands the scope of the project beyond its traditional introvert perspective. Value management is not widely used, and it has so far not been integrated with current software engineering frameworks.

A rather fundamental critique has been raised during the past 15 years against the top-down planning and control approach; that it does not reflect practice. A number of empirical studies of IS development projects find that projects are situated and emergent and require skills like empathy and improvisation rather than managerial control [25-27].

While many of the techniques from IS/software project management research may very well be necessary and useful for project management it is also a limited view.

This stream of research has little to offer on the interaction between the software development process and the organizational change process.

Socio-technical IS Development Methods: The socio-technical tradition within information systems development arose to deal with the single purpose of creating a fit between an organization and the social world on the one hand and the technologies and their employment in information systems on the other [28]. An information system design cannot be separated from the organization design, and it should be developed in an integrated process.

The socio-technical methodologies like ETHICS [13] and Multiview [11, 12] takes this holistic view into IS development. Great care is taken to ensure a correct diagnosis of the organizational problem and to establish real business objectives, to analyze the human and technical aspects of the new solution in an integrated way, to ensure real user participation and to design a socio-technical solution.

However, although the socio-technical approaches criticize a static view of the organizational implementation of information systems, ETHICS and Multiview are primarily concerned with analysis and design and ignores the development of software as the foundation for software-based information systems. Despite improvements the overall image remains; these methodologies have not really addressed the need for process interaction. They are also not much used in practice [29].

Scandinavian Systems Development: Scandinavian systems development research has consistently focused on organizational issues [30], in particular the end users [14]. Dahlbom and Mathiassen [15] described alternative approaches to the organizational issue, and concluded that the systems developer should act as an organizational change agent.

This ambitious program has not been much visible in practice, and one of the reasons is probably that most project managers lack the necessary knowledge and resources to make this happen. Some interesting cases of integrated projects are documented, for example [31] from Norwegian municipalities and Bardram's account of organizational prototyping [32]. However, these examples refer to small and relatively simple projects in terms of organizational complexity. While providing interesting cases they hardly address the more complex challenges of combining large scale systems development and organizational change, which are usually performed as separate processes managed by fundamentally different process models and management cultures [33, 34].

IS Implementation: In IS implementation research the focus has been on human, social and business effects. Leonard-Barton showed that successful organizational implementation of information systems depends on the mutual adaptation of the technology and the organization [18, 35]. Newman and Robey [19] described information systems development as a social process and suggested an integrated process model based on encounters between analysts and users. A different approach was the information infrastructure perspective, focusing on the installed base of social and technical elements, and the dynamics of bootstrapping and scaling [36].

These contributions have documented theoretically and empirically that the organizational impact of an information system is not deterministic on the structuring of work processes, and that both work processes and technology may be modified in a dynamic and emergent process.

While these contributions have provided important insights, they are not very specific in terms of practical guidance for the software project manager. They insist that the process of organizational change and technical change should be holistic and mutually adaptive, but they do not provide sufficient guidelines to solve the challenge of process interaction.

Business Process Innovation: In the early 1990s first Hammer and then Davenport introduced process innovation as a dramatic rethinking of how businesses should be organized; as processes, not functions [37]. Davenport defined the innovation process itself in five steps; identifying processes suited for innovation, identifying opportunities for innovation, develop process vision, understand existing process and design new process prototype. Software-based systems were described as key resources, both as enabler and as implementer of business processes. Davenport paid due respect to IS development methods and emphasized that IS development should fit with the corresponding processes. However, these rather general guidelines were never worked into an integrated methodology for combining IS development and process innovation [34].

Enterprise Architecture and Business Process Management: A more recent approach was enterprise architecture and business process management, which present a holistic view on both the processes and the IT capabilities of the organization, in order to ensure that individual projects can build capabilities – not just fulfill immediate needs [21]. It also emphasizes that a modeling approach is useful for integration of different levels and provides better communication.

While this approach presents an integrative view on information systems and organizational change, it does so on a relatively high level. It does not address how this should be done in more detail, and it does not relate much to the established methods for developing and implementing information systems.

Iterative Software Engineering: Modern software engineering has addressed the challenge of alignment with the organization in several ways. In 1988, as a response to the quality problems of software constructions, Boehm proposed a spiral model for software development with an iterative structure allowing for more frequent interaction with users and customers. The iterative approaches took on the challenge of unstable and changing requirements due to complex organizational issues and changing organizations. Further, both object-oriented methodologies like OOA&D [38] and Rational Unified Process [23] and the agile methodologies like Extreme Programming [39], Crystal Methodologies [40] and Scrum [24] embrace the iterative approaches for these reasons. These approaches are all strong on technical development while organizational issues and development are taken more lightly.

The dominating software engineering methodologies pay lip service to an integrated approach, but concentrate on producing the software product. The organization is seen as very important, but mainly as an arena for eliciting the requirements – not as a target for change.

3 Process Interaction

In this framework we see the organization as a combination of two: a business organization embedded in and supported by an information infrastructure. The business organization includes the formal structure of the organization, its decision and knowledge management processes as well as informal structures and processes. We regard the organization as being supported by an information infrastructure. Following Hanseth and Monteiro we regard the information infrastructure as a heterogeneous network comprising an installed base of technology, organization, culture and work practices [17, 41]. The features of this information infrastructure influence both the opportunities and the constraints. In a successful organization, this information infrastructure is an immensely valuable resource. It constitutes a backbone of the organization. However, in a world of change it is also a barrier to organizational innovation because the information infrastructure is difficult and expensive to change. This is shown in figure 1 as the organization and the information infrastructure forming together a whole and glued together in complex ways.

Planned organizational change is generally accomplished through a top-down intervention to improve the problem-solving abilities of an organization [33, 42]. Organizational change projects often use variants of Lewin's classical stage model. First, the organization is assessed and diagnosed. Then, it goes through an unfreeze stage where old patterns are loosened. In the third stage, the actual changes in routines and roles are performed, while the new structure is re-freezed in the fourth stage [42]. There are other models of organizational change like business process re-engineering [43] or total quality management [44]; but for the purposes in this paper we shall make no further assumption on how organizational change comes about.

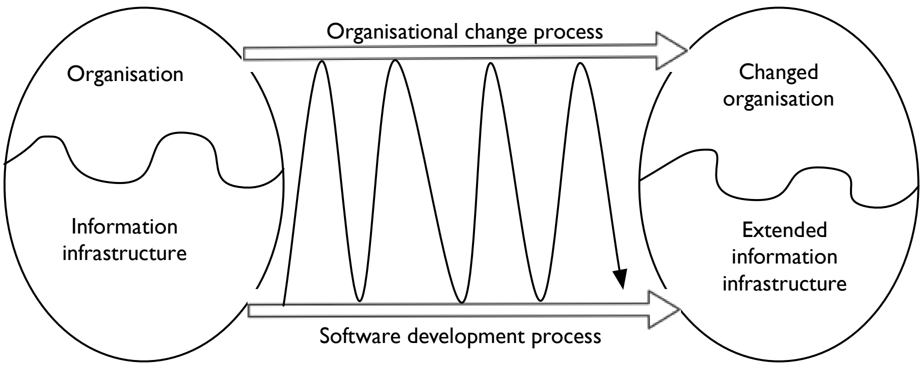

Organisational change process
Organisation
Information infrastructure
Changed organisation
Extended information infrastructure
Software development process

Fig. 1. Process interaction

We apply a process view on organizational change as well as on software development. Our concern here is the interaction between these processes. Following Newman and Robey we think it is not sufficient simply to state that these processes interact [19]. To the left in figure 1 is the existing organization forming a whole with the existing information infrastructure. We then envisage an organizational change initiative with two projects: An organizational change project performed to change the organization and a software development project performed to change the information infrastructure. In the framework, the target for the organization change processes is a changed organization that again forms a whole with the new and modified information infrastructure; but a quite different process, namely the software development process, changes this infrastructure. The software development process produces among all its results and deliverables the software that will be a major part of the extended information infrastructure.

The two processes are quite different in scope, management techniques and structure. The scope of the organizational change processes is to change the organization; while the scope of the software development processes is to develop a software-based information system. The organizational change process is usually based on management interventions on different levels: organization, team or individual [42], while the software development process is focused on an array of methodologies, techniques, tools and models. Most organizational change projects follow a waterfall model while most modern software development projects tend to be iterative and sometimes even agile. To understand and consequently manage the interaction between these processes we need to addresses the different ways that software development processes and organizational change processes interact during the period from beginning to end, as illustrated in figure 1.

4 Research Approach

The research approach is case study of the particular kind called longitudinal process research.

The case is based in the company Airline that is an international airline carrier in Scandinavia. As a part of the Marketing Division, the Airline established an Electronic Direct Channel (EDC), which was responsible for selling airline tickets and hotel reservations on the Internet. Adding sales from other online agents, Internet sales were expected to account for 25% of sales by 2005. Simplified, the tasks of the EDC were: marketing air tickets in different national markets on the Scandinavian website and receiving electronic orders. Feedback mechanisms were to be home page hits and actual bookings.

Acknowledging the commercial potential of web-based booking, the Airline decided to establish a web-based marketing channel in all important markets: Europe, Asia and the Americas. To support this new organization, a new content management and publishing system was needed. A project was initiated with the following objectives:

- To establish a web-based marketing channel in all important markets.
- To enable the editors using an easy tool to publish materials and campaigns.
- To integrate this new information system with the booking systems.

A project group of five (one project manager, one web designer and three programmers) was set up the software development project. Following earlier practice, a parallel customer project was established with an Airline project manager and a user group consisting mainly of web editors.

The case study was planned and carried out using longitudinal process research (LPR). We take LPR to be an intensive research approach that focuses attention on organizational processes as experienced by organizational actors [45-48]. LPR is the study of organizational processes with the intention of developing contextualized theories about them. According to Ngwenyama [46], the researcher conducts an intensive analysis of the context, temporal order and underlying logic of events in the organizational process under study. In our case study, we have studied organizational change processes and IS development processes as they were performed over time in and around a complex project.

LPR is based on three criteria for data collection [46]:

- Engagement with the research site is required to build any substantive theory of organizational processes.
- Participant observation enables the researcher to contextualize in making sense of practices and situations. It also makes the researcher sensitive to organizational insights encoded into actors' actions and language.
- Validity is ensured through: multiple sources of data, systematic data gathering and reliable data recording or transcription. This requires the researcher to gather empirical data so that all perspectives of the organizational processes are covered and findings can be corroborated.

4.1 Data Collection

The data was collected over a period of a year and a half while the software development project being studied lasted for almost a year. The main data source was semi-structured interviews utilizing an interview guide. The interview guide was designed to reflect a particular interest in iterative software development processes. Interviews were conducted at two sites, Stockholm and Oslo, in order to get data from inside the project and from the prospective users and other stakeholders. In addition, project meetings were observed and the findings were discussed with stakeholders. Interviews with international web editors were done by email. A secondary source of data was the huge amount of project documentation comprised of both product documentation and process documentation.

The data collection, which was done in four phases and is summarized in table 2.

Table 2. Data collection

Phase	Activities	Stakeholders	Documents
Phase 1	Initial meeting with management to agree on objectives and procedures in the study, and to collect documentation	Line manager Project managers	Project objectives and plans
Phase 2	Workshop with project and business stakeholders to get the broad picture, and separate interviews	Project manager Project group Business users	Status reports Technical documents: SW architecture, use cases, etc.
Phase 3	Separate interviews with stakeholders to construct full time line in project	Project manager Project group Business users	Status reports Project evaluation report Release notes
Phase 4	Last round of interviews. Validation meeting to confirm and discuss findings.	Line manager Project manager Business users	Case description

4.2 Data Analysis

LPR suggests three modes for data analysis to assist the researcher in closing the gaps between the findings and the empirical data [46]:

- Comprehensive analysis helps to reveal and surface deeper structures of the organizational processes.
- Temporal analysis helps to contextualize findings by placing events and situations in a narrative structure.
- Member verification ensures that interpretations and case descriptions made by the researcher are meaningful to the organizational actors.

Interview summaries and project documents were registered into an Atlas.ti database and coded. Then a systematic search for patterns was conducted using the Atlas.ti search tool. First, a timeline with significant events and iterative phases was produced. Second, iterations, context, actors, and artefacts were modeled graphically as an emerging socio-technical network. Third, a case description was written. The case description was written gradually over time, in a process of learning and also negotiation between the researcher and the stakeholders. The case description was written and rewritten for each phase (cf. table 2) of interviews as both the project stakeholders and the researcher reinterpreted the organizational processes. For example, the challenge of how the software development processes should interact with the organizational change processes was in the background in the first round, but was in the forefront during the problematic period.

The analysis further builds on the idea of critical events [48]. We split the temporal analysis into events that are critical to process interaction. We describe each critical event, its preconditions in terms of what led to the event and its consequence in terms of activities following the event. We then illustrated and explained the kind of process interaction taking place. The analysis of critical events is close to the kind of analysis where Newman and Robey [19] sliced a time scale into incidents and episodes.

5 The Case

Building on the longitudinal process analysis we identified five events particularly critical to the process interaction in the case. The five events occurred in the time order as described and the result of the former event formed the pre-condition for the next. They are described in table 3.

Table 3. Critical events in the case

Pre-condition	Critical event	Following activity	Process interaction
E-business part of airline tickets expected to grow	The Airline decides to establish a decentralized e-marketing organization.	Two projects started: Organizational change Software development	Formal agreement between the two processes
Workshops are held to specify solution	Workshops with marketing editors fail.	Editors withdraw and the project focuses on technical issues	A breakdown of the interaction between processes.
The software project lacks relevant input	The Airline project manager becomes involved in the software development.	The software solution is developed successfully	Organizational process inactive. The software development process isolated
International editors are recruited	New marketing editors enter	After a course in Stockholm, the editors start testing the system. A lot of change requests and technical problems.	Improvised interaction between the two processes.
The technical solution is stabilized	Start-up	The new business organization starts to use the new solution successfully.	The interaction is well structured even into production

Critical event 1: The Airline decides to establish a decentralized e-marketing organization

The Airline decided to establish a web-based marketing channel in all critical markets. Two projects were initiated:

- An organizational project where international editors in the actual markets were recruited, trained and put in charge of the e-business operation, as a part of the marketing division. Part of this project was a group of Scandinavian editors, who represented the Airline in the software project, headed by a project manager.

- A software development project to develop the new content management and publishing solution to be used. This consisted of an experienced project manager and four developers.

The aim of the two projects was to establish a new organizational process supported by an extended information infrastructure, as illustrated in figure 2.

Event 1:
Airline decided to establish decentralized e-marketing organization.

Result:
Formal agreement between the two processes

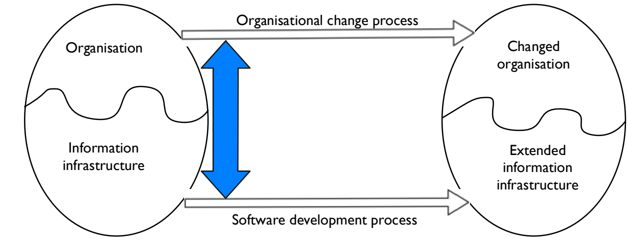

Organisational change process
Organisation
Information infrastructure
Changed organisation
Extended information infrastructure
Software development process

Fig. 2. Critical event 1

The two projects were nicely aligned and the new organization and system were planned to go into production in the following summer. However, the projects were not integrated into a common plan; rather they ran in parallel. The process models were also different: the organizational change project followed a waterfall structure, while the software development project followed an iterative and incremental structure using the Rational Unified Process. The software project was planned with five iterations. Each iteration was set up to follow the workflows in the Rational Unified Process starting with a revision of requirements, proceeding with design, coding and testing and ending with an increment, a temporary release, to be validated by the users.

We have characterized the result of this event as a *formal agreement between the two processes*; they were established with a common goal and an intention to interact during the project. This is illustrated by the unbroken arrow, which - as with subsequent figures - is used for illustration purposes and not to suggest a formal syntax.

Critical Event 2: Workshops with Editors Fail

In the two first iterations, the two project groups extended the number of use cases into 20 detailed ones. Then, they started working on a graphical prototype trying to translate the use cases visually. The workshops were not very successful and the participants interpreted them differently: the Airline project manager, who was now elaborating the software requirements specification, was moderately satisfied. According to him:

> "The workshop in the first iteration was OK because it gave the users an impression of the system. The workshop in the second was useful, but we were not able to show the users how the system would work."

Some of the editors felt alienated from the whole concept:

> "We spoke different languages, and they had no idea how we worked. We were polite and there was no conflict, but that was how we felt. We thought we might get it straight later on in the process. Use cases focused on the new system – not on how things were solved today. Development was system oriented, not on the work process."

Later, the developers said:

> "Of course, the graphical prototype should have been a full architectural prototype, but this was not possible because the necessary component from the other project was not ready. In addition, the editors did not really prioritize the workshops."

Not surprisingly, the results were unsatisfying. Nobody felt that the graphical prototype was useful. In addition, the project was held up by an important component from a sister project in Copenhagen that was delayed by six weeks. Thus, by the end of the elaboration phase (the analysis and design phase in the Rational Unified Process) the two main goals had not been reached: the business users and developers did not have a shared view of the system, and the architecture of the system was not stable.

Event 2:
Two workshops fail to convince editors of the need for a new IS solution.

Result:
Break-down in the interaction between the processes.

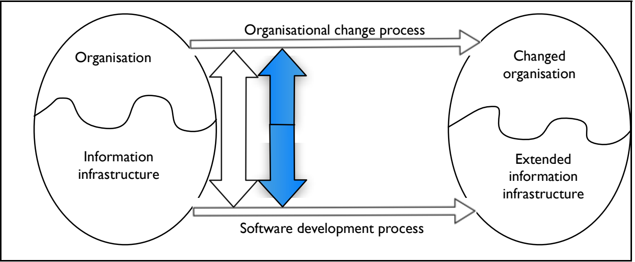

Organisational change process
Organisation
Information infrastructure
Changed organisation
Extended information infrastructure
Software development process

Fig. 3. Critical event 2

The result of event 2 was a break-down in the interaction between the two processes. The marketing editor group was unconvinced about the need for the new system, and the software project group lacked both user input and software components. In figure 3, the discontinuous arrow illustrates this.

Critical event 3: The Airline project manager becomes involved in software development

In the third iteration, the project group got a better grip on the technology and started to work more closely with the Airline project manager, who was now sitting in the same room. This iteration produced the basic functionality, enabling the users to upload content to the content database.

In the fourth iteration, the first release of the necessary component arrived and the crucial functionality of creating web pages was developed. In a few intense and informal work sessions, a design was developed as the application was prototyped. One of the developers commented:

> "When the Airline PM really joined the team, the whole atmosphere changed. We were able to experiment with screens and solutions at a practical level. Also, it was important that he really understood the technical difficulties involved. We were sitting long hours together solving real problems. It was very productive and also great fun!"

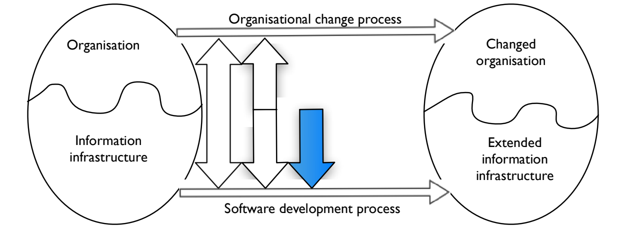
Event 3: The airline PM gets involved in software development. **Result:** Isolation of the software project.	 Organisational change process Organisation Information infrastructure Changed organisation Extended information infrastructure Software development process

Fig. 4. Critical event 3

Although the software project's spirit and technical results were greatly improved during the third and fourth iterations, the result of event 3 was that the interaction between the two processes stopped. The Scandinavian editors had withdrawn, and the international editors were not yet recruited while the Airline project manager practically had "changed sides". This left the project unintentionally "encapsulated", concentrating on the (quite challenging) technical issues. In figure 4, the half arrow, pointing at the software development process, illustrates this.

Critical event 4: New editors enter

In the winter of 2001 the international marketing editors were recruited. After a period of technical problems during testing, a beta version was presented for the international editors: In a two-day course in Stockholm for all the marketing editors, totaling at that time around 30. Most of them were introduced to the system right there without much preparation. In spite of technical stability problems and long response time due to slow APIs in the Vignette platform, the market organization and the software team perceived the course as rather successful for most of the editors. A few editors were less motivated and lacked the basic IT skills.

After the course, the editors went home and started to load materials into a test database that was later set into production. In this period, the project worked hard with error corrections and use case change orders. The project manager said:

> "Many new features were wanted from editors, both Scandinavian and the others, especially navigation features tightly connected to their work processes, page search and design. We were surprised by the volume of change orders."

The result of Critical Event 4 was that the interaction between the two processes was reinitiated. The nature of this interaction was not controlled by the RUP iteration as the previous iterations of the software project had proceeded. Rather, it was characterized by improvisation and problem solving. This is illustrated in Figure 5 by the two arrows pointing towards each other.

Event 4:
New marketing editors enter.

Result:
Improvised interaction.

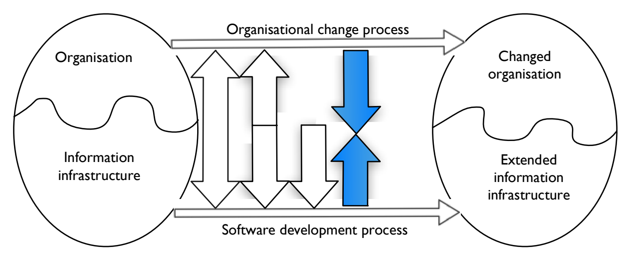

Organisational change process
Organisation
Information infrastructure
Changed organisation
Extended information infrastructure
Software development process

Fig. 5. Critical event 4

Critical event 5: Into production

Eventually all the software went into production. Some technical problems were experienced, but afterwards the technical solution was stable and in use in the new international organization. There were 50-60 users of the system: six editors in Scandinavia, the rest in Europe, USA and Asia. Most of these were part-time editors with main responsibilities in marketing or sales. Campaigns were started at a central Marketing division level or at a national level. The system allows the national editors to tailor their web pages to their local markets. The head of marketing and the editors in cooperation usually planned campaigns with external, creative consultants and bureaus. The day-to-day monitoring of the result of the campaigns was done on two parameters: the Marketing department followed the Internet traffic on the web site, while the Revenue Management monitored the actual booking. The running marketing decisions were taken on the basis of this monitoring.

Although the number of change requests remained high for the first year of operations, the international editors were satisfied. One international editor commented:

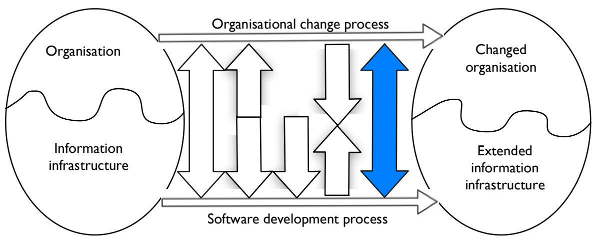
> “It aids our communication strategy of distributing information instantly (almost) of developments to the SAS product that affect our customers. It provides a means of tailoring our communication to suit the needs of our customers in Australia.”

The international editors reported that the system was relatively easy to use, but that the step-wise structure was time-consuming:

> “Pages are created in steps so once you understand the sequence it's fairly straight forward.”

> “... I do find it takes quite some time to load a new page because of all the stages you need to go through combined with the speed of the system.”

Setting the system into production was, despite some technical problems, successful, and thus, the result of the 5th critical event was that the two processes interacted as intended. The redesigned organization (decentralized web marketing) was aligned and integrated with the extended information infrastructure including the system’s publishing solution. This is illustrated in figure 6 by the unbroken arrow.

Event 5: Into production. **Result:** Organized interaction	

Organisational change process
Organisation
Information infrastructure
Changed organisation
Extended information infrastructure
Software development process

Fig. 6. Critical event 5

6 Discussion

The focus in this paper is on process interaction and we have chosen to focus our presentation on the aspects of the case that relates to this perspective. We think the evidence in this case shows that the basic problem was not poor project management. The two project managers certainly knew they were facing great challenges and they started by forming two projects with a formal agreement between them (critical event #1). The effort to establish a better e-business platform was a high profile project at the time. The Airline had also previously been through similar large projects and had considerable experience with projects of such complexity.

However, between critical event #2 and critical event #4 the two processes, for reasons partly outside the control of the two project managers, lost contact. The process interaction suffered greatly under this and was not rescued until critical event #4 where new stakeholders entered. The new stakeholders, the marketing editors, allocate time and resources to process interaction by providing detailed feedback to the software development project through testing of the software. Thus, the projects were not necessarily poorly managed, but the process interaction was not organised and not managed.

How could the project managers in the case have been improved using the proposed framework? First, the two projects could have been designed to interact better. In this planning, the project managers would have seen that the waterfall structure of the organizational change project was incongruent with the iterative structure of the software development project. The easy intervention would have been to create planned interaction at certain intervals, ensuring that the iterative software development project received the necessary input. Alternatively, and more expensively, the organizational change project could have been designed following an iterative process. Such frameworks for organization change are available [42].

Second, when the critical events occurred, the framework could have been used to assess the situation in much the same way as we have done in the case description. This would provide a better basis for intervening into both the organizational change processes and the IS development processes. At critical event #2 the software development project should have insisted that the feedback on the early prototypes was a necessary condition for getting the requirements right. At critical event #3 the Airline project manager should not have left the organizational change process; but should have insisted that time and resources should be allocated to the organizational change. At critical event #4 the process interaction increases and the only problem with process interaction was that it came very late in the process. This created much turbulence for the software development process at a time when the requirements should have been fixed well before.

We have showed in section 2 that normative IS and software project management research has a strong focus on control [9, 10]. Our framework illustrates the limitations of this perspective. There is a risk that this strong focus on control may constitute a barrier to process interaction. The reason is that the project managers (both the software development project manager and the organizational change project manager) may prefer to maintain internal project control rather than risk the uncertainties of interaction. As illustrated in the case this will increase project control, but also increase the overall risk of the project.

The software engineering frameworks, building on iterative and incremental principles of development processes [23, 40, 49] have improved software development considerably over the last years. We also find that these frameworks have an interesting potential for socio-technical innovation [51].

However, the research on agile software development [24, 40, 49] has a rather limited view on the challenges posed by process interaction . The case shows that there is much more to interaction than user participation and prototyping. To work effectively in a socio-technical context, the software development process is dependent on interaction with the organizational change process. As the case illustrates, this represents a considerable challenge. The first barrier is that there is hardly any awareness on the organizational change side of this need since software development projects are usually seen as merely technical projects. The second barrier is that the structures of the two processes are incongruent. Most organizational change projects follow a waterfall model making frequent interaction much less desirable. Software engineering projects, on the other hand, are iterative and incremental in ways that would not make sense without frequent interaction.

7 Conclusion

In this paper, we propose a framework of process interaction. Process interaction focuses attention on the meeting between the planned organizational change process and the software development process.

The framework consists of the conceptualization of process interaction as depicted in figure 1. The focus is not as much on the organization integrated with the information infrastructure as it is on the two processes leading to a changed organization and an extended information infrastructure, respectively. In particular, we have with this framework drawn attention to how the interaction plays out in an organizational and temporal context.

We have used the framework to analyze a longitudinal case from the airline carrier business and found that it enables us to understand critical events in the case. Altogether, the framework does provide understanding of process interaction and thereby we have addressed the first research question set in the introduction. We have provided a detailed answer to a particular way of addressing the question and we have in sections 4 and 5 shows the usefulness of the framework.

Our findings also lead to the need for further research into the applicability of our framework where both existing and new cases could be analyzed with the framework. A long-term vision would be to extend current software engineering frameworks and software development methodologies to encompass the process interaction.

References

1. Luftman, J., Ben-Zvi, T.: Key issues for IT executives 2010: Judicious IT investments continue post-recession. MIS Quarterly Executive 9 (2010)
2. Luftman, J., Kempaiah, R.: An update on business-IT alignment: "A line" has been drawn. MIS Quarterly Executive 6 (2007)
3. Luftman, J., McLean, E.R.: Key issues for IT executives. MIS Quarterly Executive 3, 89–104 (2004)
4. Stapleton, J.: DSDM: Business focused development. Addison-Wesley, Harlow (2003)
5. Sambamurthy, V., Bharadwaj, A., Grover, V.: Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. MIS Quarterly 27, 237–263 (2003)
6. Carr, N.: IT doesn't matter. Harvard Business Review 81, 41–49 (2003)
7. Markus, M.L., Robey, D.: Information technology and organizational change: Causal structure in theory and research. Management Science 34, 583–598 (1988)
8. Boehm, B.W.: Value-based software engineering. ACM SIGSOFT Software Engineering Notes 28, 112 (2003)
9. Cadle, J., Yeates, D.: Project management for information systems. Prentice-Hall, Harlow (2004)
10. McManus, J., Wood-Harper, A.T.: Information systems project management. Prentice-Hall, Harlow (2003)
11. Avison, D., Wood-Harper, A.T.: Multiview. Blackwell, London (1990)
12. Avison, D., Wood-Harper, A.T., Vidgen, R.T., Wood, J.R.G.: A further exploration into information systems development: The evolution of Multiview 2. Information Technology & People 11, 124–139 (1998)
13. Mumford, E.: Sociotechnical systems design. Manchester Business School, Manchester (1985)
14. Bjerknes, G., Bratteteig, T.: User participation and democracy: A discussion of Scandinavian research in systems development. Scandinavian Journal of Information Systems 7, 73–98 (1995)
15. Dahlbom, B., Mathiassen, L.: Computers in context: The philosophy and practice of systems design. Blackwell, Cambridge (1993)
16. Kyng, M.: User and computers: A contextual approach to design of computer artifacts. Scandinavian Journal of Information Systems 10, 7–44 (1998)
17. Hanseth, O., Monteiro, E.: Inscribing behaviour in information infrastructure standards. Accounting, Management and Information Systems 7, 183–211 (1996)
18. Leonard-Barton, D.: Implementation as mutual adaptation of technology and organization. Research Policy 17, 251–267 (1988)
19. Newman, M., Robey, D.: A social process model of user-analyst relationships. MIS Quarterly 16, 249–266 (1992)

20. Davenport, T.H.: Process innovation: Reengineering work through information technology. Harvard Business School Press, Boston (1993)
21. Ross, J.W., Weill, P., Robertson, D.: Enterprise architecture as strategy: Creating a foundation for business execution. Harvard Business School Press, Boston (2006)
22. Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer 21, 61–72 (1988)
23. Jacobsen, I., Booch, G., Rumbaugh, R.: The unified software development process. Addison Wesley, Reading (1999)
24. Schwaber, K., Beedle, M.: Agile software development with scrum, vol. 18. Prentice-Hall (2001)
25. Ciborra, C.: From control to drift. Oxford University Press, Oxford (2000)
26. Ciborra, C.: De profundis? Deconstructing the concept of strategic alignment. Scandinavian Journal of Information Systems 9, 67–82 (1977)
27. Orlikowski, W.: Improvising organizational transformation over time: A situated change perspective. Information Systems Research 7, 63–92 (1996)
28. Bostrom, R.P., Heinen, J.S.: MIS problems and failures: A socio-technical perspective, Part 1: The causes. MIS Quarterly 1, 17–32 (1977)
29. Fitzgerald, G.: Evaluating information systems projects: A multidimensional approach. Journal of Information Technology 13, 15–27 (1998)
30. Iivari, J., Lyytinen, K.: Research on Information Systems Development in Scandinavia. In: Currie, W.L., Galliers, R.D. (eds.) Rethinking Management Information Systems, pp. 57–102. Oxford University Press (1999)
31. Pape, T., Thoresen, K.: Evolutionary prototyping in a change perspective: A tale of three municipalities. Information Technology & People 6, 145–170 (1992)
32. Bardram, J.: Organisational prototyping: Adopting CSCW applications in organisations. Scandinavian Journal of Information Systems 8, 69–88 (1996)
33. Christensen, G., Grønland, S.E., Methlie, L.B.: Informasjonsteknologi. Cappelen, Oslo (1999)
34. Giaglis, G.: On the integrated design and evaluation of business process and information systems. Communications of the Association for Information Systems 2 (1999)
35. Munkvold, B.E.: Tracing the roots: The influence of socio-technical principles on modern organisational change practices. In: Coakes, E., Willis, D., Lloyd-Jones, R. (eds.) The New SocioTech: Graffiti on the Long Wall, pp. 13–25. Springer, London (2000)
36. Hanseth, O., Lyytinen, K.: Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet. Journal of Information Technology 25, 1–19 (2010)
37. Davenport, T.H.: Process innovation. Ernst & Young, Boston (1993)
38. Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., et al.: Object-oriented analysis & design. Marko Publishers, Aalborg (2000)
39. Beck, K.: Extreme programming explained: Embrace change. Addison-Wesley, Boston (2000)
40. Cockburn, A.: Agile software development. Addison-Wesley Publishing (2001)
41. Hanseth, O.: From Systems and Tools to Networks and Infrastructures—from Design to Cultivation: Towards a Design Theory of Information Infrastructures (2010) (unpublished manuscript)
42. French, W.L., Bell, C.H.: Organization development. Prentice-Hall, Englewood Cliffs (1998)
43. Hammer, M., Champy, J.: Reengineering the corporation - A manifesto for business revolution. Nicholas Brealey Publishing, London (1993)

44. Hradesky, J.: TQM handbook. McGraw-Hill, New York (1995)
45. Monge, P.R.: Theoretical and analytical issues in studying organizational processes. Organization Science 1, 406–430 (1990)
46. Ngwenyama, O.K.: Groupware, social action and organizational emergence: On the process dynamics of computer mediated distributed work. Accounting, Management and Information Technologies 8, 127–146 (1998)
47. Pettigrew, A.M.: Longitudinal field research on change theory and practice. Organization Science 1, 267–292 (1990)
48. Pettigrew, A.M.: Contextual research and the study of organizational change processes (1985)
49. Larman, C.: Agile and iterative development: A manager's guide. Addison-Wesley, Boston (2004)
50. Bygstad, B., Munkvold, B.E.: Software engineering and IS implementation research: An assessment of current SE frameworks. In: Kirikova, M., Grundspenkis, J., Wojtkowski, W., Wojtkowski, G.W., Wrycza, S., Zupancic, J. (eds.) Information Systems Development: Advances in Methodologies, Components and Management. Kluwer Academic/Plenum Publishers, New York (2002)