Controlling Iterative Software Development Projects: The Challenge of Stakeholder and Technical Integration

Bendik Bygstad
Norwegian School of IT
Schweigaardsgate 14
0185 Oslo
Norway
bendik.bygstad@nith.no

Abstract

Most IS projects today are not greenfield projects, but relate heavily to existing information infrastructures and business processes. How should the development process integrate with the business process? This is a crucial question, because a project that does not integrate with the business processes will be a failure, even if the software which is produced is well designed and programmed.

The Software Engineering community has embraced iterative process frameworks, like the Rational Unified Process. These frameworks claim to support a step-wise integration of the business process and the development process

This paper explores and analyses the managerial challenges of integration in such projects. Building on a longitudinal case study and an analysis of the Rational Unified process, four types of integration are identified and analyzed: External and internal stakeholder integration, and internal and external technical integration. Finding that the external integration mechanisms are just as critical at the internal, the paper concludes that the integration challenge is complex, partly outside the control of the project manager and not supported sufficiently in the RUP framework. A theory extension to RUP is suggested.

1. Introduction

Most IS projects today are not greenfield projects, but relate heavily to existing information infrastructures and business processes [5]. This affects the development of information systems in a fundamental way, making integration issues equally important as user requirements. A new system must relate to an existing information infrastructure; an often large installed base of systems, networks, users and routines [12], to which it becomes an extension.

An important contribution from the Business Processing Reengineering (BPR) movement was the focus on business processes. Business processes refer to the unique ways in which organizations coordinate work, information and knowledge, and are an important source of competitive advantage [20]. A business process may be defined as an ordered set of activities that creates value for a customer, on a relatively high level; typically, there are 10-20 processes within a company [6]. Most information systems are developed to support a business process, whether the scope is a dramatic process transformation or process improvement.

How should the development process integrate with the business process and the information infrastructure? This is an important question, because a project that does not integrate with the business process will be a failure, even if the software which is produced is well designed and programmed. Implementing change in business processes is an integration challenge [6, 29]. In simplified terms, successful implementation must be holistic, integrating both people, organizations and technology. But how should it be done in practice?

Ideally, the process models of BPR and IS development should answer this question. Unfortunately, the BPR community never really developed a full methodology for this integration [10]. During the late 1990s the software engineering community embraced iterative process frameworks, like the Rational Unified Process (RUP) [14], OPEN [13] and Catalysis [8], and also agile approaches like Extreme Programming [2]. The growing practice communities contend that these frameworks have the potential to integrate the business process and the development process through step-wise stakeholder and technical integration [1, 17]. The basic mechanism is the short iteration that produces a small release that can be tested, integrated and assessed by the business organization.

This deserves attention not only in the realm of software engineering, but also in the project management and BPR research. While skeptics have pointed to lack of practical integration support for e-business architectures [30], and poor project management support [13], the large RUP community points to a number of success projects [27, 28]. This paper explores, analytically and empirically, two important questions in this context:

- How can the project manager control the integration challenge?
- What support is there in the software engineering frameworks, like RUP?

The rest of the paper is structured as follows. In section 2 the concept of integration is defined and discussed. In section 3 the research approach is presented. Then, in section 4, the integration challenge is investigated empirically, through a longitudinal case study of an IS project in the airline industry. Section 5 discusses the findings in the case study and a content analysis of RUP. Some validity issues are also discussed. Section 6 concludes, and points to further research.

2. Integration

According to Webster, integration is "a combination and coordination of separate and diverse elements into a more complete or harmonious whole" [31].

Integrating the development process with the business process may be illustrated as in figure 1.

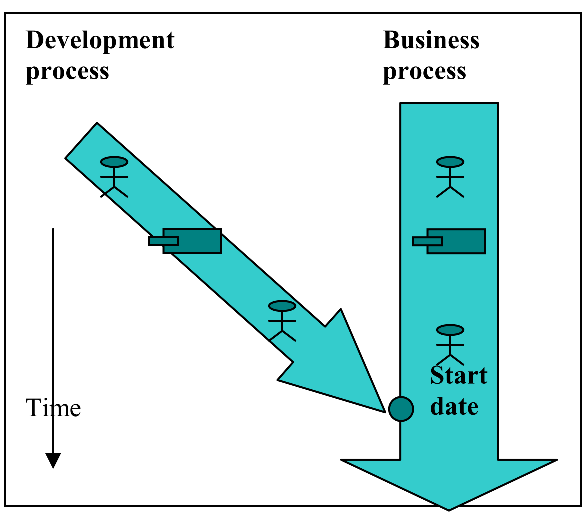

Figure 1. Development and business processes

The two key elements in both processes are seen to be technology and stakeholders, illustrated in figure 1 with simple symbols. Stakeholders are anyone (both humans and organizations) that is affected by the process, while technology may be both systems and components. The business process is often very large, and is embedded in technical and organizational infrastructures. The development process is usually small in the beginning, but grows as the software product is developed. At a certain date the software is set into production; i.e. it is fully integrated into the information infrastructure, and becomes part of the operative business process.

Both processes are unstable: It is well known that the development process is unstable in the sense that requirements, technology and teams all may be subject to change. But the business process may also change several times during the project time, due to external pressures or internal shifts of power.

The integration is multidimensional. Building on a longitudinal case study and a content analysis of RUP, four types of integration are suggested:

Perspective / *Context*	*Stakeholder perspective*	*Technical perspective*
Integration within development process	1. Internal stakeholder integration	2. Internal technical integration
Integrating with business process	3. External stakeholder integration	4. External technical integration

Table 1. Integration types

To integrate stakeholders means to convince people (or organizational units) to cooperate, and make them work together to create value. To integrate technology means to assemble components and systems in a way that make the resulting system work as intended.

It is documented that integration is usually easier if done step-wise, because it reduces the risk, and allows both the development project and the business process people to learn [3, 6]. Early user participation has long been established as an important success factor [9]. On a more political level, it also enables enrolment of important allies into the project [19]. Early technical integration secures the integrity of the system. Testing is performed in each iteration, and performance issues may also be addressed early [14].

This is recognized in the iterative software engineering frameworks: For each iteration, new stakeholders are drawn into the project. And for each iteration a small piece of solution is implemented, either in a test or pilot environment, or in real production.

It may sound easy, but it is not. As Giaglis [10] noted, the problem is not getting accept for the need of an iterative and integrative approach, but to actually manage it. To investigate the challenges in more depth, a longitudinal case study was designed.

3. Research

3.1. Case study

The integration challenge was investigated empirically, through a longitudinal case study of a software development project in the airline industry. The research approach was Longitudinal Process Research (LPR), which aims to study organizational change over time, through intensive research in the actual context [24, 25]. LPR focuses on building theories strongly embedded in the context of study. Reflecting Giddens' [11] structuration theory, context is seen by Pettigrew not only as an environment, but as" a nested arrangement of structures and processes where the subjective interpretations of actors perceiving, comprehending, learning and remembering help shape the process. The processes are both constrained by context and shape contexts" [25].

Important criteria for data collection are, according to Ngwenyama [22]:

- Ongoing engagement with the research site, to observe changes over time
- Participant observation, to contextualize and make sense of observations
- Multiple data sources, to record different interpretations of events, and to ensure validity of findings

The case, a RUP development project at Scandinavian Airline System (SAS), was researched for 18 months, using several techniques for data collection: Project managers, developers and users were interviewed at three intervals. Project meetings were observed, and a vast amount of project documentation was analyzed.

Ngwenyama [22] suggests three modes for data analysis: Comprehensive analysis helps to identify underlying structures and patterns of the organizational process. Temporal analysis helps contextualizing findings by placing events and situations in a narrative structure. And member verification ensures that the case description and interpretation researcher are considered correct and meaningful to the organizational actors.

The SAS case was analyzed in the following steps. Data was coded following the guidelines of Miles and Huberman [21]. After the videotaped interviews were summarized and registered into an Atlas database, texts were coded with in-vivo codes, using only domain (project) terms. The large volume of project documentation was coded the same way.

Then each iteration of the project was analyzed in detail, while in parallel looking for repeating patterns. Looking for integration related terms, a new layer of codes were applied: "Technical integration" was mapped to *component integration*, *testing*, *interface* and also to *legacy* (and other) *systems*. "Stakeholder integration" was mapped to *stakeholders, actors, users, customers, organizations* and *vendors*. The integration context was also coded; whether it was *internal* (to project) or *external* (to business process) integration.

Validating the case description

Suggested mechanisms and patterns were validated using the criteria suggested by Klein and Myers [15]. In this section follows some brief comments on two of these principles, and one principle taken from LPR.

The principle of dialogical reasoning addresses the validity problem of the relationship between the assumptions behind the research design, and the actual findings. In the case study this aspect was important in the coding process described in the previous section. Moving from codes to patterns, it was systematically tried to construct the opposite pattern, and ask whether it could make sense on the same data.

The principle of multiple interpretations calls for sensitivity in interpreting conflicting accounts of the same case. In the case description, special care was taken to analyze instances where data was contradictory, for example when developers and user representatives had different accounts of what had happened.

The principle of member verification [22] was followed this way. After the initial workshop with important stakeholders, a timeline for the project was constructed. This was sent to the participants for verification. After about a year of studying the project, a preliminary case description was written, and commented by stakeholder. An lastly, a validation meeting, where project managers developers and business users were invited, assessed and commented on the final case description.

3.2. Integration mechanisms of RUP

RUP is both a process framework and a commercial web based product. As illustrated in figure 2, RUP is structured in four phases: inception, elaboration, construction and transition (Figure 1). Within each phase there is one or several iterations consisting of disciplines, starting with business modeling and ending with the physical deployment of software components

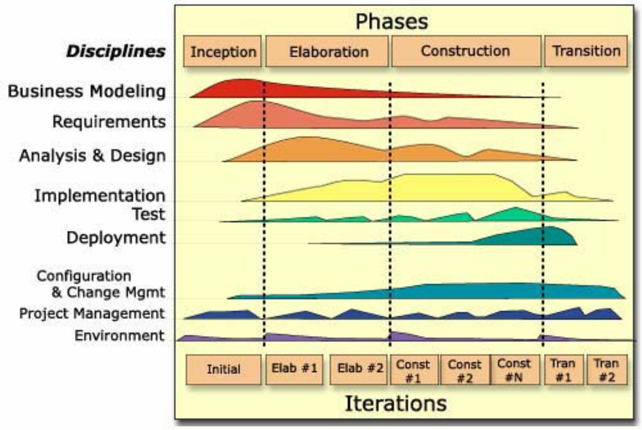

Figure 2. Structure of RUP [26]

Through a content analysis [16] of RUP, it was analyzed which mechanisms in RUP that support the integration of stakeholders and components. The main source was RUP on-line [26], 3700 web pages describing the principles of RUP, and providing a rich toolbox of guidelines, risk lists and document templates.

To analyze the integration mechanisms three levels of detail were defined:

- High level: Principles describing the rationale behind RUP, focusing on important business and development issues.
- Medium level: Guidelines describing iteration structures and risks, focusing on project management issues.
- Low level: Detailed work descriptions and document templates, focusing on specific tasks and artifacts.

The content analysis was designed to provide data for answering two questions: At what operational level does RUP support step-wise integration, and in which *phases of the project* is the support given.

First, a systematic search in RUP on-line for key integration terms was done. Search terms included *component integration*, *testing*, *interface*, *legacy* (and other) *systems*, and s*takeholders*, *actors, users*, *customers, organizations* and *vendors*. Each finding was categorized into the High/Medium/Low level. Frequencies would indicate the *operational level* of support for integration.

Second, the model iteration plans in RUP on-line were analyzed. (RUP on-line generously provides model iteration plans for each phase). Each activity was categorized into the stakeholder/technical integration scheme (table 1). Frequencies were assumed to indicate the level of integration support by *project phase*.

4. A RUP case study: Developing an e-business system for an airline company

This section contains a case description of the SAS CMI Project.

Business and IT Context

Scandinavian Airlines System is an international airline carrier based in Scandinavia. It is owned by the Swedish, Danish and Norwegian governments (50%) and private investors. In 2001 the SAS group had 25.000 employees, and turnover was 51 billion SEK (SAS Annual Report 2001). The IT services are organized as a company, Scandinavian IT Group (SIG), owned by SAS. Cooperation is close, but SAS are free to choose other vendors. IS projects are often organized in a "twin structure", one SAS project and one SIG project. The SIG employees in general have good domain knowledge, SAS being their largest customer.

The Business Process: Marketing and selling airline tickets on the Internet

As a part of the Marketing Division SAS has established the Electronic Direct Channel (EDC), directed at marketing and selling airline tickets and hotel reservations on the Internet. The EDC owns the SAS home page, including campaigns and on-line booking services. In 2002 this part of SAS was expected to sell around 3 % of total sales, and is seen as an important competitive asset for SAS in the coming years, increasing to an expected 14% in 2005. Adding sales from other online agents, Internet sales are expected to account for 25% of sales in 2005. The business process is, in simplistic terms, to market air tickets in national markets over the scandinavian.net site on the Internet, receive and process electronic orders. Feed-back mechanisms, monitored on a day-to-day basis, are web hits at the SAS home page, and actual booking.

The first generation of web marketing was based on relatively simple technology. In 2000, acknowledging the commercial potential of web-based booking, SAS decided to form a project EC R2 (Electronic Channels Release 2) to build a more advanced solution on a content management platform, Vignette. A number of sub-projects were initiated, to build on this platform. One of these projects was the Content Management Interface (CMI), which was set up to design and develop the interface for the marketing editors who maintain the SAS homepage and campaigns.

The first generation of the SAS web site had mainly been published in Scandinavia, by editors using html and FrontPage as publishing tools. The new tool should support a distributed organization around the world, with a large number of part time editors having little technical experience.

Project objectives and organization

The main objectives for the CMI project were:

- To establish a web-based marketing channel in all important SAS markets, including Europe, Asia and Americas.
- To enable the editors with an easy to use tool for publishing materials and campaigns, without having to use html coding.
- To integrate this new system with the booking systems.

In spring 2001 the CMI project was set up by SIG with a project group of five; one Project Manager, one web designer and three programmers. Following earlier practice, a parallel customer project was established, with a SAS project manager, and a user group, consisting mainly of Scandinavian web editors.

Project challenges

There were two main challenges in this project. First, to develop a publishing and marketing solution with new technology, fitting into a large and very complex infrastructure, consisting of a new content management technology, a web booking platform, SAS' ('legacy') booking systems and the international Amadeus booking system. Second, to implement this solution into an international marketing organization that was being established in parallel with the development of the CMI solution.

RUP had been chosen by SAS in 1999 as their standard software development methodology. Following up, the central process and tool unit in SIG had chosen RUP as a basis to develop a process model for SIG, and provided process support and RUP education and coaching. For SIG, the main reason for choosing RUP was to provide process support for component based development.

The EDC project had tailored RUP for use in the projects (a "development case"), linking it to established practices in SIG. The EDC project had also developed an initial software requirements specification and a software architecture description. Each iteration was set up to follow the disciplines in RUP, starting with a revision of requirements, proceeding with design, coding and testing, and ending with an increment, a temporary release, to be validated by users. Also, in accordance with RUP, a stakeholder and risk analysis was done.

The first two iterations: Unsuccessful stakeholder integration.

The initial requirements specification, written by IT personnel at SIG, contained 14 use cases for the CMI application. In two workshops the two project groups extended the number of use case 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 they were interpreted differently by the actors: The SAS project manager, who was now elaborating the requirements specification, was moderately satisfied:

"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 like strangers to 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.

The use cases were very - theoretical, and it was hard to participate. Also, I was so busy running the existing site that I had no time to spend on this, really. I felt guilty because of that, because I should have given much more input."

Another editor said:

"Use cases focused on the new system, not on how things were solved today. Development was system oriented, not on the work process"

The developers later said:

"Of course, the graphical prototype should have been a full architectural prototype - but this was not possible, because the presentation engine component were 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. The presentation engine component, which was crucial for creating a page, was developed in another project in Copenhagen, and it was 6 weeks delayed. Thus, by the end of the elaboration phase, the two main goals were not reached: The users and developers did not have a unified view of the system (though this was not realized), and the architecture of the system was not stable. The situation was analyzed and the risks were assessed by the customer and the two project managers. Because of time pressures, they decided to proceed, and took care to prepare as much as possible to integrate the missing component later.

Iterations three and four: Successful internal technical integration

No code had been produced in the first two iterations. In the third iteration, the project group got a better grip on the technology, and started to work more closely with the SAS PM, who was now sitting in the same room. This iteration produced the basic CMI functionality, enabling the CMI users to upload content to the content database.

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

Commented one of the developers:

"When the SAS project manager 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!"

Test cases were also produced, and at this time the project group felt things were on the right track. While the user group was not involved, the SAS project manager (PM) communicated frequently with the editors, assuring them that the project was on track.

After the start of the fourth iteration, the Sept 11th terrorist attacks in New York shook the whole airline industry, and as a result of this, combined with a tight project deadline, the project was scaled down. Only 10 of the initial 20 use case should be developed. By the end of November, the user group started testing the system. Towards the end of the iteration the SIG project manager went on maternity leave, and the SAS PM was made PM for the whole project.

Iteration 5 - Improvised integration

So far the project had proceeded more or less as planned. But now a different phase started, which was not characterized by planned iterations, but rather by problem solving and improvisations. The PM commented:

"At a certain point in a project the focus changes: Budget and plans are no longer the main consideration, and everything is focused on getting the product finished."

There were two reasons for this. First, the project experienced external technical integration problems. In the fourth iteration the project waited for the completion of two important software components, the SVE and the presentation engine, which were programmed by another SIG project in Copenhagen. Both were seriously delayed and not stable until spring 2002.

The developers commented:

"It was all very frustrating, to wait for components that we were absolutely dependent on."

There were also other technical problems. SIG had four different technical environments: Development, integration testing, production testing and production. In this case, the test environments did not fully support the CMI, and intermediary solutions had to be made. Adding to this, a memory leak occurred which took a long time to identify and fix. These problems made testing more difficult.

Second, after beta testing had started, there were a rising number of change requests from the user group, and changes were introduced frequently during the spring 2002. One of the editors said:

"From the end of February, we started testing at our workplace. At that time I had no mental picture of the whole systems, only fragments. In March, when we started to load real data, I was able to understand the workings of the system. Until then, it was like looking at the system through a pair of binoculars."

As the editors got more experience the number of change request increased. Said the Oslo editors:

"The period around the start of the system we sat together with the developers and really understood the whole system. This was great, and the first time we really could give real input".

In March there was a two-day course in Stockholm for all the editors, at that time around 30. Most of them were introduced to the system right there, and in spite of technical stability problems, and long response time due to slow interfaces (APIs) in the Vignette platform, the course was perceived by the EDC and CMI project as rather successful for most of the editors. A few of them were less motivated, and lacked the basic IT user skills.

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

Results

Everything was set into production on May 30th 2002, and has been in stable production since then. The three goals of the project were achieved within time and budget: A web based marketing channel in all SAS markets was established. The marketing editors in this organization had access to an easy-to-use tool to tailor the marketing and selling of SAS services to their local markets. And the web pages were integrated with the booking systems, enabling users to buy tickets online.

However, one important question remains. When problems arose in the 5th iteration, most of the RUP control mechanisms were abandoned, and the later part of the project was characterized by ad-hoc problem solving and improvisations. Why? The next section will discuss the integration problems in the project.

5. Discussion

5.1. The managerial challenges of integration

The findings of the case study is discussed within the framework of four integration types (presented in table 1). They will also be related to the analyis of integration support in RUP.

The managerial challenges associated to the four integration types are summarized in table 2.

Integration type	Mangerial challenge
Internal technical integration	Step wise integration of components needs careful planning and competent execution, but is well supported in RUP.
Internal stakeholder integration	Step wise integration of project team members calls for sensitivity, especially in relation to user representatives.
External technical integration	Integrating to the technical infrastructure is outside the direct control of the PM. It calls for careful planning, but also for improvisations.
External stakeholder integration	Integrating to the business process is also outside the direct control of the PM. It calls for planning, but also for political skills and improvisation.

Table 2: The managerial challenges of integration

Internal technical integration

This aspect was, after some initial problems, solved very nicely in iterations three and four. The CMI system was built incrementally and controlled, utilizing the rich mechanisms in RUP for step-wise integration and testing.

Internal stakeholder integration

Iterative development projects are not based on a frozen requirements specification, but relies on a step wise interaction with user representatives through the iterations. Thus, it is critical to convince these representatives that the project is important.

At the start of the CMI project the most central users in the business process, the Scandinavian editors were drawn into the project, and given a central role in the workshops. The workshops were not successful, mainly because the PMs failed to convince the editors of the need of a new system. The project group responded to this with a certain degree of encapsulation, establishing a close team with the SAS PM, who "changed side", i.e. identified more with the project than with the business process.

The progress of software development benefited very much from this cooperation. But when the number of change requests increased in the later parts of the project, it was an indication that the iterative approach was not working optimally regarding user requirements.

External technical integration

Integrating with a large, existing information infrastructure calls for careful planning, but also for improvisations. Often, the infrastructure is subject to change during the project, and it is therefore impossible to integrate in a step wise manner. Obviously, the project manager has to comply and adjust to these changes, whether they were planned or not.

The CMI project suffered from being dependent on components that were delayed. Thus, while skillfully planned, the step wise technical integration was not working properly. Instead, the external integration problems appeared at a time when the editors tried to integrate the system into their organization, and threatened the whole implementation.

The main technical problems arose from the fact that the development environment, and also the test environment, was configured differently from the production environment. In the test phase this led to a series of small but important problems of instability. In the production phase it led to the cache problem. As the project was designed this kind of risk was hard to avoid, because a test environment mirroring a changing production environment would have been exceedingly expensive. At the end, the problem was gradually solved by a competent Data Centre, using problem solving techniques, and not a structured development process.

External stakeholder integration

Integrating with the business process calls for political skills and improvisations. Political skills are needed to build the alliances that make the solution work. Improvisations are necessary to integrate with an unstable process.

The main target group for the CMI specification was the international editors, needing a standardized and simple interface. Unfortunately, at this point they were not yet appointed, and could not participate. Thus, the continuous change of the business process made step-wise stakeholder integration unfeasible.

On the other hand, the international stakeholders were successfully enrolled into the project from March 2002, and during the busy, and partly improvised, period towards the production date, most of their requirements were satisfied. It should be noted, however, that during the first six months of production there were still a large number of change requests, indicating that important business needs were not covered in the specification. It should also be noted that the late integration of stakeholders was done by improvisation.

5.2. Support in RUP

To sum up, the project was very successful regarding internal technical integration, had some problems regarding internal stakeholder integration, and serious problems regarding external technical and stakeholder integration. What support did the PMs get from RUP in this challenge?

The integration support in RUP (measured as described in section 3.2.) is illustrated in table 3. The activities in RUP were counted, and classified in the four integration types. They were then distributedin the four phases: Inception, Elaboration, Construction and Transition.

Phase ----> Integration type	Incep.	Elab.	Constr.	Trans.	Sum
Int.techn. integration	11	23	23	14	**67**
Int.stake-holder integration	5	4	4	4	**17**
Ext.techn. integration	3	3	1	4	**11**
Ext.stake-holder integration	14	5	3	7	**29**

Table 3. Distribution of RUP activities supporting the integration types through RUP phases.

As the table shows, the integration support is unevenly distributed. It is very strong regarding internal technical integration, throughout the phases, while it gives surprisingly little external technical integration support. Internal stakeholder integration is weak thoughout the project, confirming, perhaps, the critique against RUP for poor project management support.

Support for external stakeholder integration is strong at the early phase, but gets weaker in the later phases. Analysis of the three levels - high, medium and low - shows that RUP gives strong high-level support to external stakeholder integration, but little practical support at the operational level. The same pattern is found regarding external technical integration. This may indicate an internal contradiction in RUP: It gives strong, declarational support to step-wise external integration, but too little practical support.

Of course, it is unreasonable to blame RUP for the late integration problems in the CMI project. But the "extended iteration" in the project, where the structured method gave little support, and the PMs resorted to improvisation and problem solving, indicates that this challenge is underrated. Thus, the CMI case illustrates two important findings.

First, the integration challenge of iterative software development projects is larger, and more complex, than is described in the research and text book literature. Even the very competent CMI team experienced serious problems with external integration. For practitioners this is an important factor in their risk management: To succeed, the project manager must manage something that is basically outside his control.

The root of this risk is the fact that both processes are unstable. As illustrated in the CMI case, the focus in a development project change a lot over time, making it hard to plan i detail. And changes in the business process should be treated as a normal feature, not as an unexpected incident. For the proejct manager of iterative projects this aspect also presents opportunities, but in a world of tight schedules and budgets the integration of two unstable processes is a major risk.

Second, the support in RUP is not sufficient to handle this challenge, in spite of some minor contributions[18]. Especially on a practical level RUP needs stronger support for external stakeholder and technical integration on the detailed level, in particular in the construction and transition phases. In practice this means to emphasize the external component/legacy system issues in the architecture description, and to supplement risk lists, iteration plans and task descriptions with activities that address business process stakeholder issues more explicitly. At the same time it should be acknowledged in RUP that the external integration cannot be planned in detail.

Planning and improvising

External technical and stakeholder integration were seen to depend on both planning and improvisations to succeed. Finding that improvisations are important in change processes is congruent with the emergent view on organizations and with other IS research [23, 22]). However, it should not lead us to conclude that better planning is unnecessary or impossible. As the CMI case shows, improvisation puts a lot of pressure on a project manager, and introduces more risk into the project. The strength of iterative software engineering framework is to reduce that risk, through step wise integration. Thus, it is important to extend the capabilities of RUP and other frameworks, to include better external integration.

6. Conclusion

The initial question in this paper was how the IS development process should integrate with the business process. This paper explores and analyses the managerial challenges of integration in iterative software development projects. Based on a longitudinal case study and a content analysis of RUP, there are three findings.

- Integrating the two processes is a complex challenge, illustrated by the four integration types: External and internal stakeholder integration, and internal and external technical integration.
- The iterative approach enables a step wise and controlled integration. However, to succeed in integrating the two processes, the project manager has to integrate something which is basically outside his control: A changing infrastructure and an unstable business process. Thus, integrating two moving targets is really the objective.
- Iterative frameworks do, in principle, have mechanisms to address this challenge. RUP, the software engineering framework analyzed here, supports internal integration, but does not give sufficient external integration support. It needs a theory extension, addressing this challenge: Activities and checklists that explicitly support external integration, especially in the construction and transition phases. At the same time it should be acknowledged that the external integration cannot be planned in detail, and is dependent on competent improvisation.

Further research could study other iterative projects and look closer into which patterns that emerges in iterative projects. One might also investigate new models for how IS project managers may contribute more actively to integrate the development projects and the business, as recently done by Boehm [4]. This is an area well suited for cooperation between researchers and practitioners.

Acknowledgements

Thanks to the different stakeholders of the SAS/SIG project, in particular Christian Mejlvang, Jorun Akerberg and Gunilla Johanssson, for access to the project and for commenting on various versions of the case description. Thanks to Peter Axel Nielsen and Bjørn Erik Munkvold for discussion and comments. The reviews from the anonymous HICSS reviewers were also very useful.

6. References

[1] Blomberg, J. (2001). "Using the RUP for Enterprise e-business Transformation." The Rational Edge(January).

[2] Beck, K. (2000). Extreme Programming Explained: Embrace Change. Boston, Addison-Wesley.

[3] Boehm, B. (1988) A Spiral Model of Software Development and Enhancement. IEEE Computer, 1988, May, pp. 61-72.

[4] Boehm, B. (2003). "Value-based software engineering." ACM SIGSOFT Software Engineering Notes. Volume 28:2

[5] Ciborra, C. (2000). From Control to Drift. Oxford, Oxford University Press.

[6] Davenport, T. H. (1993). Process Innovation. Boston, Ernst & Young.

[7] Davenport, T. H. (1998). "Putting the Enterprise into the Enterprise System." Harvard Business Review July-August, pp. 121-131.

[8] D'Souza, D., Wills, A. (2002). "The Catalysis Approach."www.catalysis.org

[9] Garrity, E. J., Sanders, G.L. (1998). Information Systems Success Measurement. Hershey PA, Idea Group Publishing.

[10]Giaglis, G. (1999). "On the integrated design and evaluation of business process and information systems." Communications of the AIS 2(1).

[11] Giddens, A. (1987). The Constitution of Society. Cambridge, Polity Press.

[12] Hanseth, O. (2002). "From systems and tools to networks and infrastructures. Toward a theory of ICT solutions and its design methodology implications."www.ifi.uio.no/~oleha/ib_ISR_3rd_resubm2.html

[13] Henderson-Sellers, B., Unhelkar, B. (2000). OPEN Modelling with UML. Harlow, Addison-Wesley Longman.

[14] Jacobson, I., Booch, G., Rumbaugh, R. (1999). The Unified Software Development Process. Reading, Addison Wesley.

[15] Klein, H. K., Myers, M. D. (1999). "A Set of Principles for Conducting and Evaluating Interpretive Field Studies In Information System." MIS Quarterly 23(1): 67-94.

[16] Krippendorff, K. (1980). Content analysis : An introduction to its methodology. Beverly Hills, Sage Publications.

[17] Kruchten, P. (2000). The Rational Unified Process. Reading, Addison Wesley Longman.

[18] Kruchten, P. (2001). "Using RUP to Evolve a Legacy System."www.therationaledge.com/content/may01t_legacy_pk.html 2001August 17

[19] Latour, B. (1987). Science in Action: How to Follow Scientists and Engineers Through Society. Cambridge, MA, Harvard University Press.

[20] Laudon, C. K., Laudon, J. P. (2003). Management Information Systems. Upper Saddle River, New Jersey, Pearson Education.

[21] Miles, M. B., Huberman, A. M. (1994). Qualitative Data Analysis. Thousand Oaks, Sage Publications.

[22] Ngwenyama, O. (1998). "Groupware, Social Action and Emergent Organizations: On the Process Dynamics of Computer Mediated Distributed Work,"." Accounting, Management and Information Technologies 8(2-3): pp. 127-146.

[23] Orlikowski, “Improvising Organizational Transformation Over Time: A Situated Change Perspective,” Information Systems Research, 7(1), 1996, pp. 63-92.

[24] Pettigrew, A. M. (1985). Contextual Research and the Study of Organizational Change Processes. Research Methods in Information Systems. E. Mumford, North-Holland.

[25] Pettigrew, A. M. (1990). "Longitudinal Field Research on Change Theory and Practice." Organization Science 3: pp. 267-292.
[26] Rational Corp (2001) RUP on-line.
[27] Rational Corp (2003) IBM Rational Customer Success. http://programs.rational.com/success/
[28] Royce, W. (1998). Software Project Management. Reading, Mass., Addison-Wesley Longman.
[29] Sarker, S., Lee, A. S. (2002). "Using a Positivist Case Research Methodology to Test Three Competing Theories-in-Use of Business Process Redesign." Journal of the Association for Information Systems 2(7), pp 1-72.
[30] Smolander, K. (2003). "The Birth of An E-Business System Architecture: Conflicts, Compromises, and Gaps in Methods". 36th Hawaii International Conference on System Sciences.
[31] Webster, N. (1986) Webster's third international dictionary of the English language. London, Encyclopedia Britannica.