

Four integration patterns: a socio-technical approach to integration in IS development projects

Bendik Bygstad,* Peter Axel Nielsen[†] & Bjørn Erik Munkvold[‡]

*Norwegian School of IT, Schweigaards gt. 14, 0185 Oslo, Norway, email: bendik.bygstad@nith.no, [†]Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg, Denmark, email: pan@cs.auc.dk, and [‡]Agder University College, Serviceboks 422, 4604 Kristiansand, Norway, email: bjorn.e.munkvold@hia.no

Abstract. *This paper aims to contribute to a theory of integration within the field of information systems (IS) project management. Integration is a key IS project management issue when new systems are developed and implemented into an increasingly integrated information infrastructure in corporate and governmental organizations. Expanding the perspective of traditional project management research, we draw extensively on central insights from IS research. Building on socio-technical IS research and software engineering research, we suggest four generic patterns of integration: big bang, stakeholder integration, technical integration and socio-technical integration. We analyse and describe the advantages and disadvantages of each pattern. The four patterns are ideal types. To explore the forces and challenges in these patterns, three longitudinal case studies were conducted. In particular we investigate the management challenges for each pattern. We find that the patterns are context-sensitive and describe the different contexts where the patterns are applicable. For IS project management, the four integration patterns are a contribution to the management of integration risks – extending the vocabulary for assessing and mitigating these risks in IS development. For practitioners the four integration patterns represent an analytical framework to be used in planning modern IS development projects.*

Keywords: information systems development, integration patterns, project management

1. INTRODUCTION

The aim of this paper is to contribute to a theory of integration within the field of information systems (IS) project management. The background for the need of such a theory is a gradual but fundamental change in the discipline of IS development. Two decades ago most IS projects

were greenfield projects: on the basis of a requirements specification, a brand new system was designed and programmed. When (eventually) finished it was installed and the users were trained in new interfaces and routines. The main challenges for the IS project manager were to get the requirements specification correct and complete, and to make sure that development and testing produced a technically stable system.

Today, IS development is still a demanding craft, requiring specialized skills, sound methodologies and competent project management. However, the IS project manager faces two additional challenges:

- *The business environment is more turbulent.* Many IS are developed or customized as an important part of an organizational change process (Davenport, 1993; Laudon & Laudon, 2004). In the turbulent world of the 21st century where organizations continually change, merge and network, there is less time for organizational implementation after the completion of the system; a new system must *align-in-action* (Ciborra, 1997; Larman, 2004) with the organization. A system that is not aligned and integrated quickly with the organization will be a failure, even if the software is well designed and programmed.
- *The technical environment is more complex.* A new system must also relate to an existing information infrastructure. This is seen as a heterogeneous network, consisting of an installed base of systems, physical networks, organization, culture and work practices (Ciborra, 2000; Hanseth & Lyytinen, 2004). Both opportunities and constraints for change are heavily influenced by the attributes of this base. In a successful organization this heterogeneous network is an immensely valuable resource; it constitutes the backbone of the organization. However, in a world of change it is also a barrier to business adaptation or innovation, because the information infrastructure is difficult and expensive to modify.

The successful IS project manager needs to address both these challenges. In a way, this redefines the role of the project manager, making him or her an integrator of both business and technical elements. The business forces urge rapid innovation, while the information infrastructure supports incremental change. To make systems useful the IS project manager must make stakeholders and technology work together in a complex and changing setting.

On a general level, the alignment between the organization and the IS is a long-time concern for both IS research and the practice community. Successful IS development and implementation depends on a socio-technical approach (Coakes *et al.*, 2000), both technical and organizational alignment (Leonard-Barton, 1988), as well as comprehensive business and user participation (Kappelman & McLean, 1994). There has also been a shift from the somewhat simple concept of users to the broader notions of social actors (Lamb & Kling, 2003) and stakeholders (Coakes & Elliman, 1999).

However, at the more operative project level the processes of integration and mutual adaptation are less understood (Majchrzak *et al.*, 2000). There are many questions that the IS project manager must address. To what extent do modern software engineering (SE) methodologies support the stepwise integration between technology and organization? How and when, should the different stakeholders be enrolled into the project? How and when, should the new software components in the development project be integrated with the existing technical

infrastructure? What are the management trade-offs in such projects? Somewhat surprisingly, these issues are not addressed in any detail in the mainstream IS project management literature (McManus & Wood-Harper, 2003; Cadle & Yeates, 2004), which are mostly concerned with internal project control. The multitude of questions facing IS project managers we turn into a single question that has guided our research reported in this paper: *What are the main patterns of integration in IS development projects?*

In this paper we address these questions and seek answers in a set of four generic patterns of integration, which we developed based on previous theoretical research and an empirical study of three iterative software development projects.

The next section discusses integration in a socio-technical context. Then, in section 3, we present our research approach and in section 4 we describe four integration patterns in detail. In section 5 the four patterns are illustrated through three longitudinal case studies. In section 6 we discuss practical and theoretical implications, and limitations. We conclude briefly in section 7 with our main findings and point to further research.

2. INTEGRATION: A SOCIO-TECHNICAL PERSPECTIVE

This section discusses and defines integration in a socio-technical context. Then we assess the contributions from IS development methodologies regarding integration. Finally, we describe integration in IS development projects in more detail.

2.1 Integration in a socio-technical context

According to Webster, integration is 'a combination and coordination of separate and diverse elements into a more complete or harmonious whole' (Webster's, 1986). The term *social integration* is a key concept in sociology, denoting a reciprocity of practices between actors (Giddens, 1987), while *technical integration* is commonly understood as the assembly of different technical artefacts, such as components and systems (Jacobson *et al.*, 1999). A socio-technical approach must build on both these definitions, in the sense that it must relate to two very different research traditions.

Although integration has not been extensively researched, previous IS research has shown that it is a complex concept. Wainwright & Waring (2004) analysed the term in three domains – the technical/systems, strategic and organizational – and suggested a high-level framework to understand all these elements. Other high-level contributions, mostly in an Enterprise Resource Planning (ERP) context, include for example Hsiao & Ormerod (1998) and Chang (2006).

In a socio-technical context, integration aims at creating a working solution that includes both social and technical elements. An important question, then, is what is the unit of investigation? Some proposed answers to this are:

- a socio-technical work system (Bostrom & Heinen, 1977a; Mumford, 1983; Leonard-Barton, 1988; Alter, 1999);

- an actor-network (Hanseth & Monteiro, 1996; Walsham, 1997; Monteiro, 2000);
- a web of computing (Kling & Scacchi, 1982);
- an IS (Avison & Fitzgerald, 2003); and
- an information infrastructure (Ciborra, 2000; Hanseth & Lyytinen, 2004).

Although their key terms vary, all these contributions share the view that the object of study is a heterogeneous socio-technical network, consisting of both social and technical elements. For example, Bostrom and Heinen wrote in their classic article in *MIS Quarterly* (Bostrom & Heinen, 1977b) that an organization is a socio-technical work system, consisting of two interacting systems:

The technical system is concerned with the processes, tasks and the technology needed to transform inputs to outputs. The social system is concerned with the attributes of people (e.g. attitudes, skills, values), the relationships among people, reward systems and authority structures. It is assumed that the outputs of the work system are the result of the joint interactions between these two systems. Thus, any design or redesign of a work system must deal with both systems in an integrated form. (p. 18)

Another example is Avison & Fitzgerald's (2003) description of the EBay internet auction:

The electronic auction is an information system, comprising people, rules, procedures, technology, software, communications and allied services, such as delivery companies and third-party insurance companies. (p. 6)

We understand integration in this tradition as the process of *linking* these elements into socio-technical networks, consisting of both social and technical nodes. Very briefly, we note that the ontological status of socio-technical networks may be contested. For example, Latour states that 'network is a concept, not a thing out there. It is a tool to describe something, not what is being described' (Latour, 2005, p. 131). In contrast, most contributions within IS socio-technical research seems to accept the pragmatic view that it is *useful* to describe the technical and the social network as an artefact (Bostrom & Heinen, 1977a; Alter, 1999). This paper subscribes to the pragmatic view. We acknowledge, however, that in an ontological sense it may be problematic to treat associations between stakeholders in the 'same' way as associations between technical components.

The integration process connecting the nodes may be planned or emergent. For example, Bostrom & Heinen (1977b), Avison & Fitzgerald (2003) and Alter (1999) describe integration as basically planned and controlled, while Kling & Scacchi (1982), Ciborra (2000) and Hanseth & Monteiro (1996) describe it as basically emergent. Leonard-Barton, in her seminal paper from 1988, described the process as mutual adaptation between technology and organization, emerging in a set of smaller and larger cycles. Our position is that integration is seldom clean-cut; it is often partly planned and partly emergent. The emergent aspect, however, emphasizes that integration is *happening in time*; in our turbulent world usually it cannot be planned in detail. In other words, a successful socio-technical solution is more dependent on a successful integration process than of a socio-technical specification.

Building on these contributions we define integration as *the planned and emergent process of linking different stakeholders and technology into existing socio-technical networks*.

2.2 Integration in IS development methods

The socio-technical IS development methodologies, such as ETHICS and Multiview offer a holistic view on integration. Great care is taken (1) to ensure a correct diagnosis of the organizational problem; (2) to analyse the human and technical aspects of the new solution in an integrated way; (3) to ensure real user participation; (4) and to design a socio-technical solution. The strength of these methodologies is the careful analysis and design of a real-world situation and the design of a socio-technical solution to improve on this situation.

However, this is also a limitation of the methodologies; a too strong emphasis on analysis and design. The ETHICS method has 15 steps, but only the two last ones are concerned with implementation, and at a rather high level of detail (Mumford, 1983). Multiview has five steps, where the last one is technical design and construction (Avison & Wood-Harper, 1990). It is fair to say that the implementation aspect was improved by the introduction of Multiview 2 (Avison *et al.*, 1998), but the overall picture remains; the focus is on the planned analysis and design. They do not address integration as a process.

In some important aspects, this was significantly improved with the introduction of iterative and incremental methods in the 1990s. Building on Boehm's spiral model (Boehm, 1988), modern SE frameworks such as Rational Unified Process (RUP) (Jacobson *et al.*, 1999), Microsoft Solutions Framework (MSF) (Microsoft 2004), XP (Beck, 2000) and DSDM (Stapleton, 2003) provide some excellent mechanisms to understand and manage integration. First, the iterative structure allows for learning and organizational experimentation. Second, the incremental structure allows for stepwise technical integration to reduce risk. And third, a strong stakeholder orientation acknowledges that SE needs a user and business perspective, and also to address organizational issues.

We will use RUP in our further analysis, because RUP is both a widely used method and also quite comprehensive in its description. As illustrated in Figure 1, RUP is structured in four phases: inception, elaboration, construction and transition. Within each phase there is one or more iterations consisting of disciplines, starting with business modelling and ending with the physical deployment of software components. Thus, each iteration is a mini-project, resulting in a small increment that extends the existing solution.

While the support for stepwise technical integration is easy to identify in this structure, several researchers have argued that iterative frameworks also have the potential to integrate the business organization and the new IS through stepwise stakeholder and technical integration (Kruchten, 2000; Blomberg, 2001; Stapleton, 2003). There are limitations to this view in a socio-technical integration context. Although RUP is process-oriented and stepwise, integration is – at the operative level – mainly addressed in the context of component integration and integration testing of software, not as a socio-technical process (Bygstad, 2004).

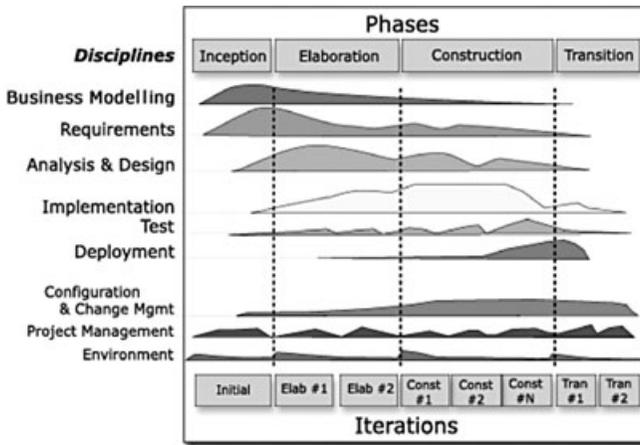


Figure 1. Structure of the rational unified process (IBM, 2002).

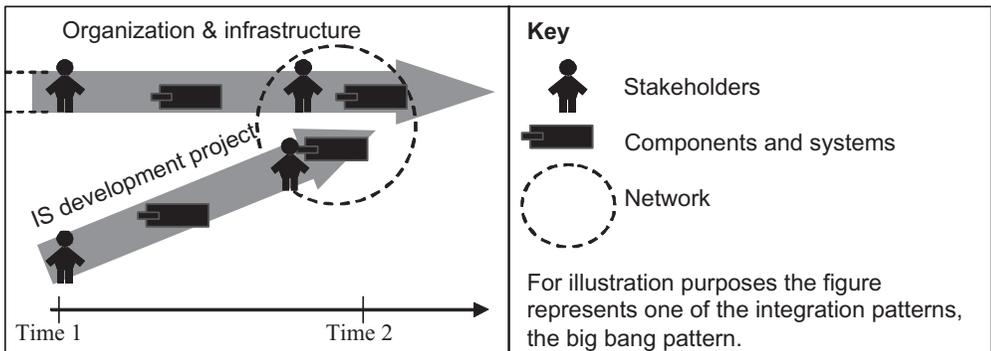


Figure 2. The relationship between the development process and the organizational process.

2.3 Integration in IS development projects

Building on the socio-technical insights, and on the iterative SE methods, we are now in a position to reframe integration in IS development projects.

In the practical context of IS development we frame our analysis as the interaction of two processes, the business organization and the IS development project. The aim of the IS development project is to extend the existing organization and infrastructure, i.e. to extend the existing socio-technical network. The extension includes not only the new software, but also new stakeholders such as system owners, business users, vendors and support staff. It also includes the new links to existing legacy systems, realized as middleware, web services or other technical solutions.

The integration between the IS and the business organization is seen through a process lens (Newman & Robey, 1992); we analyse integration over time. Figure 2 provides a general

illustration of the relationship between the two processes. The organization is often large and complex, coexisting and depending on information infrastructures (Ciborra, 2000). After project initiation at time 1, the IS typically grows gradually in scope and complexity as the project proceeds. The IS is then set into production (at time 2), and the project organization is usually (but not always) terminated. For a successful development project, somewhere along the paths from times 1 to 2, the new stakeholders and technology are integrated into the (extended) socio-technical network.

Integration is a demanding process, and is a main responsibility of the project manager. As argued in the previous section, it is to a certain extent possible to plan and design parts of the integration, and this is usually done in a requirements specification and systems design document. However, because of the emergent and social nature of the socio-technical network, important parts cannot be specified at the beginning of the project. In this research we analyse the integration process in three steps:

1 Enrolment of stakeholders and/or technology. Formally, this is an act of agreement that specifies the role of the project and the new system in the organization or the information infrastructure. In practice it is often necessary to convince stakeholders that it is in their interest to participate (Hanseth & Monteiro, 1996; Monteiro, 2000). Key stakeholders include future users, managers, other projects, IT managers, data centre technicians, vendors, consultants and also customers. Technology is enrolled through high-level architectural descriptions and interoperability requirements (Jacobson *et al.*, 1999).

2 Adaptation of technology and/or organizational processes. Fitting the software to the organization is done through different techniques such as requirements elicitation, user input and prototyping (Jacobson *et al.*, 1999). Conversely, the organization may have to change to achieve the benefits expected from the technology; business processes may be redesigned and relationships to other organizations may be affected (Mumford, 1983; Leonard-Barton, 1988; Davenport, 1993). At the level of information infrastructure, a new system must comply with technical architectures and interfaces, but the infrastructure may also have to change because of the new system (Ciborra, 2000).

3 Stabilization. After adaptation, the different elements may be validated and set into production. A successful implementation is defined here as the stabilization of the socio-technical network; i.e. that the solution is technically stable and key stakeholders agree on the value of the network (Mumford, 1983; Hanseth & Monteiro, 1996). Of course, the term stabilization does not imply that the organization is permanently 'frozen'; rather it means that the socio-technical network is successfully extended.

The three steps of integration are often difficult to achieve, for several reasons. First, both the development process and the organization change during the time of the project, thus making the requirements a moving target (Boehm, 1988). Second, the new software system must integrate with the existing technical environment in the business organization, which often is a very complex task. Today, these environments often have the character of very large information infrastructures consisting of legacy systems, global networks and heterogeneous user communities (Ciborra, 2000).

And third, implementation success is often defined differently by different stakeholders. For the software engineers, a successful implementation is usually achieved when requirements are fulfilled and the system is technically stable; while for the users, a successful implementation is to make a new or modified organizational process fully operational through the use of the new IS (Alter, 2000).

Thus, what is needed to be researched is the interplay between two dimensions of integration in IS development projects – the socio-technical dimension and the temporal dimension. The socio-technical dimension concerns the relationship between the stakeholders and technology in the IS development project and business organization. The temporal dimension relates to how the steps of integration are distributed in time. The next section describes how we addressed these issues empirically.

3. RESEARCH APPROACH

The research was conducted within an interpretive tradition (Klein & Myers, 1999) using a mixed approach (Mingers, 2001). The approach is mixed with a theoretical argument and literature study on the one hand and three longitudinal case studies on the other hand. The case studies all follow the longitudinal process research (LPR) as the empirical approach (Pettigrew, 1985; 1990). Table 1 gives an overview of the mixed approach.

Through a series of steps we constructed a generic set of integration patterns, and explored them through three longitudinal case studies. The research steps are shown in Table 1, illustrating that the relationships between literature review, data collection and data analysis were iterative rather than sequential.

Table 1. Research steps

Step	Activity	Main source
1	Understanding the challenge of integration in IS development practice	Exploration: Interviews with 12 IS project managers
2	Assessing the support for integration in IS research	Literature review: IS project management research, IS implementation research, SE research
3	Constructing the first two patterns of integration	LPR: Initial phase of first case study Literature study: RUP
4	Designing the four integration patterns	LPR: Socio-technical IS research Literature study: RUP
4	Exploring the forces in the patterns	LPR: Three case studies, following the structure illustrated in Figure 3
5	Modifying and detailing the four patterns	LPR: Data analysis Literature review: Socio-technical IS research, RUP
6	Validating the patterns	LPR: Member validation Literature review: Socio-technical IS research

IS, information systems; SE, software engineering; LPR, longitudinal process research; RUP, rational unified process.

The point of departure was the real-world challenge of managing integration for IS development project managers. Twelve experienced IS project managers were interviewed on their experiences with iterative SE methods. Their view on integration was that this aspect was generally underrated in the methods – and also by their own business clients. On the other hand, integration was identified as a key factor for success. A review of current IS project management literature, IS implementation research and SE research was conducted. From this point of departure, a longitudinal case study was designed to study integration in IS development projects. Three longitudinal case studies were conducted, following the approach of LPR (Pettigrew, 1985; 1990). The strength of LPR is that it enables the researcher to engage deeply in a socio-technical change process, and to elicit patterns and causations from a rich and often contradictory material. LPR also allows both the researcher and the organization to learn over time because perspectives and frames may change, making the studied organization a reflective participant. The cases were:

- Airline case – development of an e-business solution at an international airline.
- Government Audit case – Development of an audit support system for a large government auditing agency.
- University case – Implementation of a learning management system (LMS) at a university.

The cases were selected on two criteria. They all addressed the challenge of integration, and they also followed modern, iterative software development process models.

3.1 Data collection

Data was collected in accordance with the principles of LPR (Ngwenyama, 1998):

- Engaging with the research site at several times during the study, to collect data reflecting changes over time.
- Participatory observation, to understand the actors' language and problem solving, and to make sense of different situations.
- Collecting systematically different types of data, to secure validity.

Figure 3 illustrates the research process followed in each of the studies of the three development projects. The cases were followed over a period of 16–18 months each, starting when the project had run for some months, and ending after the system had been in production for more than 6 months. Thus, the focus was set on the later parts of the development and at the start of the production period.

Data collection was conducted using a wide range of sources: a total of 40 semi-structured periodic interviews with various stakeholders, participatory observation, and an extensive amount of project documents.

3.2 Data analysis

Interview summaries, project documents and technical reports were coded in the *Atlas.ti* tool for coding qualitative data, following the guidelines of Miles & Huberman (1994). LPR suggests three modes for data analysis (Ngwenyama, 1998):

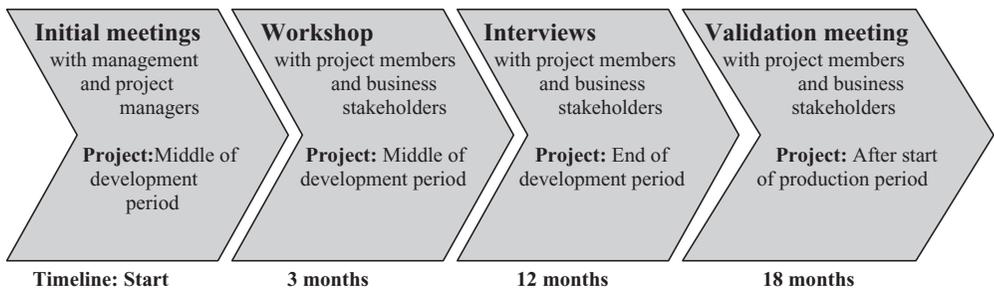


Figure 3. Data collection process in each of the three case studies.

- 1 Comprehensive analysis helps to identify underlying structures and patterns of the organizational process.
- 2 Temporal analysis helps contextualizing findings by placing events and situations in a narrative structure.
- 3 Member verification ensures that the case description and researcher's interpretation are considered correct and meaningful to the organizational actors.

The first case study was designed to study integration in a large RUP project in an international airline company. After the first workshop an initial analysis was done. First, a time line with important events and development process iterations were documented. Second, development process iterations, context, stakeholders and artefacts were modelled graphically. A graphical representation of temporal and socio-technical aspect of the case helps to reduce the complexity of the narrative (Nandhakumar, 2002). Then we applied a socio-technical framework as our interpretive lens. Stakeholders, technical components, organizational processes, information infrastructure and political factors were all identified to have played important roles. We then tried to understand the dynamics of the case by identifying the mechanisms that integrate the socio-technical network.

The Airline development project was using RUP to structure the process, aiming at a stepwise integration of both technology and stakeholders. From this analysis the two first patterns emerged, the *stakeholder integration pattern* and the *technical integration pattern*. A more detailed analysis revealed, however, that the project struggled with the apparent complexity of this approach.

Returning to IS socio-technical research and using the Klein & Myers (1999) principle of *dialogical reasoning*, we asked which alternative approaches could have been available to the project manager? The answer was a first draft of the last two patterns. Drawing on concepts from SE research (Jacobson *et al.*, 1999) and earlier research on patterns of process integration (Lloyd *et al.*, 1999), four process patterns emerged as a generic and comprehensive set of integration patterns. These patterns are explained in detail in the next section.

More than a year after this initial design, with a much richer body of data from all the three cases, we did a more comprehensive analysis. The second case, the Government Audit

project, answered many of our questions emerging from the Airline case, and emerged (somewhat surprisingly) as an instructive example of the socio-technical pattern. The third case, the University LMS implementation, had been selected more consciously as an example of the Big Bang pattern.

The core of the analysis was not to verify the completeness of the patterns, but to investigate the forces in them; i.e. to understand the dynamics and the managerial challenges. In this context we also assessed the degree of support in RUP for each of the patterns. Using the principle of the hermeneutic circle in interpretive IS research (Klein & Myers, 1999), the cases were reinterpreted within the patterns approach. This analysis is described in section 5.

Member validation was performed at various steps of the research process. First, the graphical socio-technical network was discussed with central stakeholders. Later, the case reports were discussed with the interviewees at a validation meeting. Lastly, the draft versions of the scientific papers on the patterns were sent to central stakeholders for comments. After an exchange of views and some extensions, the organizations accepted that their names were disclosed.

4. FOUR INTEGRATION PATTERNS

The term pattern is used in the broad sense as a general solution to a common problem in a context (Ambler, 1998). We have described the IS development project and the organization as two socio-technical networks, consisting of both stakeholders and technology. During an IS development project, the integration between the IS and the organization may happen early and stepwise or it may be performed at the end of the development project when the system is put into production and becomes fully operational in the business organization.

Thus, we have two types of elements (stakeholders and technology), and two types of integration modes (stepwise and at the end of the development project). Combining these two dimensions we identify four different integration patterns (Table 2):

- 1 Big bang – the integration of both stakeholders and technology is performed at the end of the development project.
- 2 Stakeholder integration – the integration of stakeholders is done stepwise, while the integration of technology is done at the end of the project.
- 3 Technical integration – the integration of technology is done stepwise, while the integration of stakeholders is done at the end of the project.

Table 2. Integration patterns

		Integration of components and systems	
		At end of project	Stepwise
Integration of stakeholders	At end of project	Big bang	Technical integration
	Stepwise	Stakeholder integration	Socio-technical integration

4 Socio-technical integration – integration of both stakeholders and technology is done stepwise.

The next section describes these patterns in detail. The main structure of the pattern is described, as well as its advantages and disadvantages. The support from RUP is assessed, and we also draw on insights from socio-technical IS research.

4.1 The big bang pattern

In this pattern (Figure 4) the network between the IS development project and the business organization is established at the time when the development project is completed, and the software product is delivered. Integration between the processes is described in a requirements specification. The development project then produces the required software and documentation. At the production date, the system is organizationally implemented, i.e. the components are installed and set into production, and the users are enrolled and taught how to use the system.

The main advantage of this pattern is a high degree of internal project control as well as software system integrity. The focus of the project is on the software system, executing control by traditional project management techniques. Supporting mechanisms in RUP are requirements, phases and workflows.

Disadvantages are well documented: late integration runs the risk of both user resistance and of technical integration problems (Boehm, 1988; Kappelman & McLean, 1994; Lamb & Kling, 2003). Both the organization and the infrastructure are subject to change, making it very difficult to predict the integration in detail at the start of the development project. Both user and technical requirements may be incomplete or inadequate at the start of production. Integrating a large number of new technical components into legacy systems has been made feasible the past years through interoperability standards and middleware, but puts heavy demands on testing and configuration management and tools. Even small and seemingly trivial technical

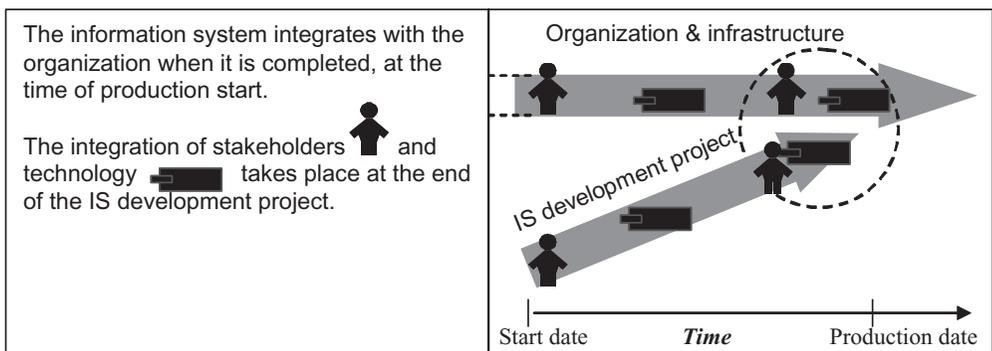


Figure 4. The big bang pattern.

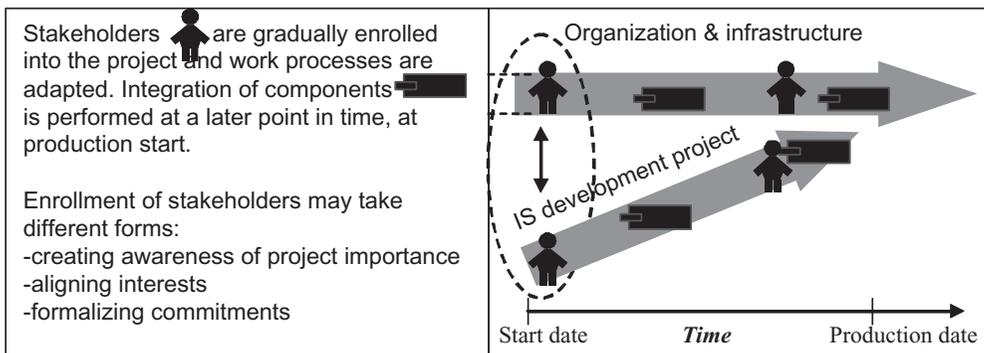


Figure 5. The stakeholder integration pattern.

mismatches may lead to serious production problems. Thus, stabilizing the socio-technical network is difficult within this pattern, because there is very little room for adaptation and learning during the project.

4.2 The stakeholder integration pattern

In this pattern (Figure 5), the stakeholders from the business organization are enrolled stepwise through workshops, user groups and other organizational mechanisms. Early user participation has long been established as an important success factor (Mumford, 1983; Avison & Fitzgerald, 2003). Further, it secures more reliable requirements and early user validation of the product (Jacobson *et al.*, 1999). The supporting mechanisms in RUP for this integration pattern are iterative (stepwise) and stakeholder-based development (including, in principle, all actors that are influenced by the project).

A disadvantage with this pattern is that it increases the complexity of the project. In addition to managing the project group and the technical development challenges, the project manager also has to spend time and effort negotiating with external stakeholders that may not prioritize the project or change their views frequently.

Disadvantages also include the risk of unstable commitment from the business stakeholders if this is only declared but not supported in practice. They may lose interest in the project and return to their own organizational processes. They can also become so involved that they 'change side' and become part of the development project and move away from the business organization. The late technical integration may also lead to some of the stability problems described in the big bang pattern, since even small technical mismatches may lead to serious production problems.

4.3 The technical integration pattern

In this pattern (Figure 6), the focus is directed at stepwise technical integration. A network of components and systems is established early in the project by stepwise enrolment. The main

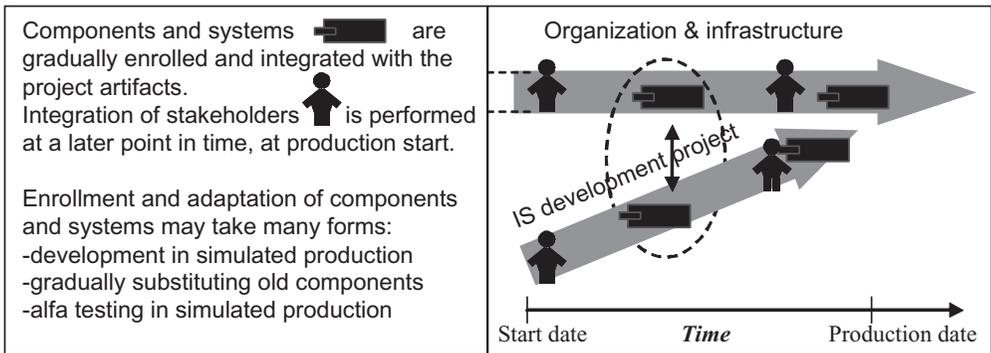


Figure 6. The technical integration pattern.

advantage of this pattern is that early and stepwise technical integration secures technical stability at the time production starts. It may also validate the technical quality of the software system.

Stabilization is enhanced by the technical nature of the network, but without stakeholder integration there is the risk of friction at the production start date. Late stakeholder integration may lead to poor requirements and user resistance, as described in the big bang pattern. SE research has documented that it is easier and cheaper to correct faults as early as possible in software development (Royce, 1998). Supporting mechanisms in RUP are component-based and incremental development (each new component extends the earlier version), which allows for integration and testing in each iteration (Jacobson *et al.*, 1999).

Disadvantages include the aspect that early technical integration increases the complexity of the development process. Testing environments must include more of the production technology, which sometimes is not possible. Also, the logistics of the project becomes more complex, with both sides (the development team and the data operations centre) waiting for other components or other's support.

4.4 The socio-technical integration pattern

In this pattern (Figure 7) there is a dynamic coupling between the IS development project and the business organization.

Innovations in one process may change the other process, and technology is used for early routinization of work practices in the business organization. This pattern has the potential to support organizational innovation through a stepwise and dynamic interaction between stakeholders and technology. Stakeholders are enrolled through arguments of concrete and short-term benefits of the technology, and encouraged to experiment with the software components in the organizational processes. If this is successful, the solution will attract new users who may use the technology in new ways or request more, or modified functionality. This may

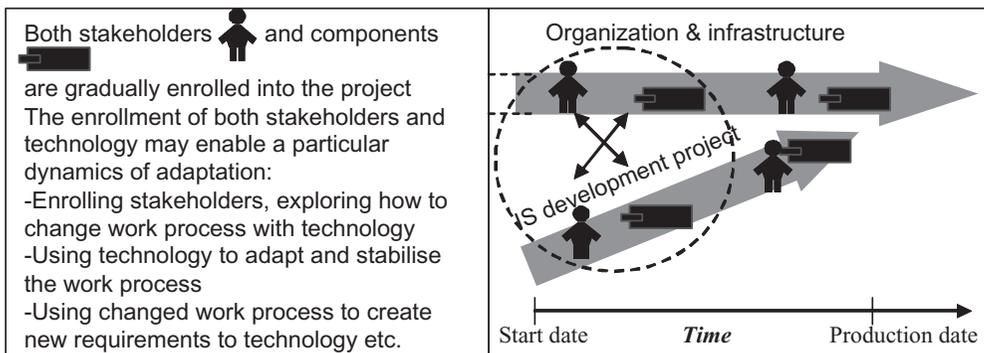


Figure 7. The socio-technical integration pattern.

broaden the applicability of the technology, attracting more users and so on. We may describe this process as self-feeding, where behaviour is stabilized through technology, and the installed base of users and technology grows. This perspective is also supported by IS implementation research contributions, regarding innovation as the mutual adaptation of organization and technology (Leonard-Barton, 1988).

However, while this pattern greatly reduces the risk of integration problems at the production start date, the disadvantages include huge challenges for project control. It increases the complexity of the development process, making it less predictable. Although RUP to a certain extent supports both stakeholder and technical integration, it contains no explicitly described mechanisms supporting this kind of socio-technical dynamic.

5. CASE ILLUSTRATIONS OF THE PATTERNS

The four integration patterns are ideal types, and one should not expect to find them in pure forms. Rather, our interest is to explore the forces and challenges within them. In the following we use the three cases to illustrate the four patterns, as shown in Table 3. For each case we describe the enrolment and adaptation of stakeholders and technology, and the stabilization of the socio-technical network. In particular we focus on the managerial challenges faced by the project managers in each case.

5.1 An example of the big bang pattern: implementing a LMS at a university

The University of Oslo (UiO) is Norway's largest and oldest institution of higher education, currently having 32 000 students and 4600 employees. As with most universities, both external (relevance) and internal (quality) pressures call for a more coherent organizational and technical support for learning. One important response is the establishment of pedagogically

Table 3. Four cases of integration patterns

Integration pattern	Case organization	Development project	Managerial challenges
1. Big bang	University	Learning management system for supporting learning environment	Stabilizing a top-down solution
2. Stakeholder integration	Airline	E-business: Marketing and selling tickets on the Internet	Stepwise enrolment of users in an unstable organization
3. Technical integration	Airline	E-business: Marketing and selling tickets on the Internet	Stepwise enrolment of components in a large, changing technical infrastructure
4. Socio-technical integration	Public auditing agency	Audit process support system for financial auditing	Establishing an organizational mechanism for mutual adaptation

motivated integration of ICT in teaching, learning and student collaboration. In Oslo this took the form of implementing an LMS, of which the main services were course, content and student management. University top management initiated the development project, and the University Senate-allocated financial resources to the central ICT department (USIT).

The project was organized and conducted by the central ICT department (USIT), who chose a top-down approach in its work to establish enterprise-wide common infrastructure solutions and services for learning. After a failed pilot project based on generic software (Lotus Notes) and local knowledge (within medicine), USIT decided to go ahead with a top-down development strategy and building on a standard LMS package (Classfrontier). Integrating with administrative systems, the LMS was developed over a period of 2 years, using an iterative in-house development approach. The main challenge was enrolling scientific staff. Resistance was strong, with the faculties and departments generally being hostile both to the use of an LMS and to the central IS staff. Two strategies for enrolment were followed. The first strategy was to integrate the existing service structures through a large project of automating the input from administrative systems. The second was to establish a number of activities that were directed at scientific staff, including a formal organization and the provision of rich information on the project.

In 2001 the solution was set into production in a full-scale implementation, including software installation and educating and supporting users. The LMS was not integrated with existing department-based systems, but a new infrastructure was established, tightly integrated with administrative IS. Among academic staff and students there was strong initial resistance towards the LMS. One year after the LMS was set into production, only a small number of students were actively using the system. The organizational implementation was greatly helped by external pressures. In 2003, the national so-called 'Quality Reform' was implemented at the University. The LMS supported key aspects of the reform, such as net-based

learning in a group-supporting environment, and became important for the university management in implementing the reform. During 2003 the LMS stabilized, and became an integral part of the learning process at UiO. The case is described in detail in (Lanestedt & Bygstad, 2003).

5.1.1 Managerial challenges

Supported by top management, the development project team chose to develop an ambitious and complex solution, and integrate with the university's learning processes at a late stage. This big bang strategy had the advantages of concentrating the efforts of the project group on developing a strategically aligned and well-designed solution. On the other hand, the implementation risk of this strategy is clearly illustrated. The organizational implementation was seriously delayed, and without the help of external pressures it is likely to have failed altogether.

5.2 The RUP project at SAS: an attempt to combine the stakeholder integration pattern and the technical integration pattern

SAS is an international airline carrier based in Scandinavia. In 2001 the SAS group had 25 000 employees, and turnover was 51 billion SEK. SAS's first generation of web marketing had been in use mainly in the Scandinavian market, and based on relatively simple technology. In 2000, acknowledging the commercial potential of web-based booking, SAS decided to establish an electronic marketing channel and e-business. In spring 2001, a project based on the RUP (Jacobson *et al.*, 1999) was set up with the following objectives:

- to establish a web-based marketing channel in all-important SAS markets;
- to provide the international marketing editors with an easy content management interface (CMI) tool to publish campaigns; and
- to integrate the new system with the SAS legacy booking systems and the Amadeus international booking system.

The project made an attempt to use RUP to accomplish both stakeholder and technical integration. The case is described in detail in (Bygstad & Nielsen, 2005).

5.2.1 The stakeholder integration pattern

In spring 2001, the project was set up with a project group of five members: one project manager, one web designer and three programmers. Following earlier practice, a parallel organizational change project was established in the marketing department with an SAS project manager and a user group consisting mainly of Scandinavian marketing editors. The main target group for the system was the international marketing editors, in need of a standardized and simple interface. Unfortunately, at this point they had not yet been appointed, and thus could not participate. Instead, the Scandinavian editors were invited to participate.

Table 4. The SAS project, enrolment of stakeholders during five iterations

Iteration	Actors enrolled	Mode of enrolment	Result
1	Customer project manager Scandinavian marketing editors	Requirements specification workshop	Unsuccessful. Editors not convinced of need for new solution.
2	Customer project manager Scandinavian marketing editors	Workshop	Unsuccessful. Use cases not mapped to work practices.
3	Customer project manager	Prototyping	Quite successful prototyping in close cooperation with customer project manager.
4	User test group	Screen testing	Partly successful, but not really testing work process.
5	International marketing editors	Workshop GUI testing	Partly successful after a lot of improvisation. But many change requests after production start date.

After two disappointing workshops where the project group failed to translate the editors' work practice into the system use cases, the project manager from the marketing division became a part of the development project and very successfully prototyped the main part of the software product. In a way he changed sides, and later on became the project manager of the whole project. Late in the project the international marketing editors were appointed, and were sent to Stockholm to learn the new system. This led to a large number of change requests, and a rush to get the system finished on time, abandoning the iterative RUP structure. Thus, stabilization was achieved by improvisation at a late stage of the project. The five iterations, with actors and modes of enrolment are illustrated in Table 4.

5.2.2 Managerial challenges

Stepwise integration of stakeholders certainly demands open communications with users and fora for discussing requirements. But as the case shows, this is not enough. Real enrolment depends on a successful translation of needs, mutually adapted to the work practices. When this was not achieved in the first iterations – for reasons not controlled by the project – the project responded through a certain 'encapsulation', i.e. concentrating on the internal tasks of the development project and postponing the enrolment of the external actors. The price for this postponement was paid later in the project, when stakeholders had to be integrated by improvisation in the last iteration.

5.2.3 The technical integration pattern

The SAS case is also an interesting example of the technical integration pattern. In accordance with RUP, it was planned with a stepwise integration of software components and systems. The technical environment was a complex one. The CMI was developed as an integrated part

Table 5. The SAS project, enrolment of components and systems during five iterations

Iteration	Components and systems enrolled	Mode of enrolment	Result
1	RUP requirements specification	Project plans	The project managers attempted to follow RUP through most of the project.
2	Graphic prototype	Workshop	Unsuccessful, because basic components were not available.
3	Development tools (e.g., ASP) CMI components	Configuration programming	Successful, but integration of external components postponed because they were not ready.
4	CMI components External components Test environment	Programming Procedure calls Integration testing	Unsuccessful integration testing, because of instability.
5	External components Test environments Legacy systems Amadeus	Procedure calls Integration testing Production testing	Because of integration problems, the iteration was extended several months. But eventually successful.
Start prod.	Legacy systems Amadeus	Production environment	Successful and stable production since start.

RUP, Rational Unified Process; ASP, Active Server Pages; CMI, content management interface.

of the other e-business platform-related projects, and it also had to be integrated with the existing information infrastructure – the legacy booking systems and the international Amadeus booking system. All the projects were component-based, and the publishing solution was specified in UML and programmed in JavaScripts and Active Server Pages (ASP), using Vignette's application program interfaces. Through five iterations a large number of internal and external software components and systems were linked into the project. The enrolment of components is illustrated in Table 5.

Although the project was planned with stepwise technical integration, external component and systems integration was concentrated on the last two (and in particular the last) iterations. This is not according to RUP, and it was not intended. The reasons for the postponement were outside control of the project manager. First, the external components were delayed, blocking the integration of the product. Second, the production environments were too large and complex to be installed, and had to be partly simulated. Adding to this, the technical infrastructure was changing during the project, making configuration management difficult. The technical solution was stabilized in the last iteration. This was partly achieved by improvisation and not iterative planning. Cooperating with the experienced SAS data centre, the project group succeeded in setting the system into stable production in May 2002.

5.2.4 Managerial challenges

Stepwise technical integration is a beneficial way to reduce risk, but it comes at a price. It increases the complexity of the development process. Examples of this in the SAS project

were a risky dependence on external software components, and a complex and changing technical infrastructure. This situation created much friction in the development process, and thus also considerable pressure on the project manager. Parallel with the stakeholder integration pattern, in such situations the development project may choose to respond with project encapsulation, concentrating on internal development issues. This will increase internal project control but also risk implementation failure.

5.2.7 Combining the two patterns

With both the stakeholder integration and the technical integration patterns identified in the same case, will this not result in a socio-technical integration pattern? The answer is clearly no. In both the design and the execution of these patterns the SAS project handled the human and the technical integration issues separately. They were never coupled dynamically, in the sense that the IS development process and the organizational processes were mutually influencing each other. Thus, at the start of production at SAS, there were *two* networks – one mainly technical and one mainly social – that were integrated. We will discuss the risks of this strategy in section 6.

The following section contains an example of how these processes may be dynamically and stepwise integrated into one network.

5.3 An example of the socio-technical integration pattern: development of an audit support system

The Office of the Auditor General (OAG) is instituted by the Norwegian Constitution. Its main tasks are to audit the central government, ministries and their agencies' accounts. The OAG is based in Oslo where most of the 450 employees are located. Auditors perform three types of audit: financial audit, corporate control and performance audit. The system developed in this case study supports the financial auditing process. The OAG case is described in more detail in (Bygstad, 2005).

5.3.1 The process-oriented IT audit support project

After a feasibility study had assessed and rejected five commercial audit support systems, it was decided in 1999 to develop a tailor-made system. The main objective for the project was to standardize the auditing process through a modern IS. The project was planned and organized within the MSF (Microsoft 2004). Like RUP, MSF is an iterative and incremental process framework focusing on stepwise integration.

The project organization was designed to align the organization and the development, with a strong management and quality focus. Large resources were allocated to secure a stepwise integration. Through five iterations, a number of stakeholders and software were gradually enrolled into the project. First, a strategy of broad user involvement and long workshops in the first two iterations failed to stabilize a version. Then, in the third iteration, an organizational

Table 6. The Office of the Auditor General project, enrolment and stabilization during five iterations

Iteration	Stakeholders enrolled	Components and systems enrolled	Mode of enrolment	Result
1	Steering group Six project teams 10 test teams	Requirements specification	Workshop	Unsuccessful. Too complex process.
2	Six teams 10 test teams	Prototype	Workshop	Unsuccessful. Too complex process.
3	Methodology group Developers	GUI design Internal components	Prototyping Process redesign Testing	Successful mutual adaptation of development and auditing process
4	User departments	Internal/external components	Demos Department process redesign	Successful
5	Pilot users	Internal and external components	Courses Testing	Successful
Start prod.	All auditors	Whole system	Integrating with department work practices	Successful, but some resistance from auditors not enrolled earlier

mechanism for mutual adaptation between the development process and the organizational processes was introduced. This was the Audit Methodology Group, which had been given two mandates: to provide requirements input to the process-oriented it audit support (PROSIT) project, and to change the auditing process. Seeing that the two mandates reinforced each other, they were able to use the learning from the PROSIT project to restructure the auditing process, and also structure the PROSIT software system according to the new auditing process. The organizational implementation of the resulting IS was done stepwise. After redesigning the auditing process, all departments were asked to plan their own PROSIT implementation. This was successfully done during the fourth and fifth iteration. The steps are illustrated in Table 6.

5.3.2 Managerial challenges

The PROSIT project illustrates that the socio-technical integration pattern has the great advantage of coupling stakeholders and software components dynamically. It allows for organizational experimentation. An important lesson learned is that large user workshops were unsuccessful, while a small, mediating group with a clear mandate including both the development project and the auditing process was very successful.

The complexity of the integration strategy of the project was considerable. It was well supported by the iterative structure and team model of MSF, and by the organizational

construct of the Methodology Group. An important aspect was that the government agency was not under strong external pressures. It had considerable organizational and technical flexibility, with a reasonable slack in time and resource pressures. This allowed for the necessary experimentation at the expense of the project schedule.

6. DISCUSSION

This section discusses implications for research and practice, and assesses the limitations of the suggested patterns.

6.1 Implications for research

Integration in a socio-technical sense is a key issue in IS research, particularly because it implicitly addresses the relationship between technology and organizations. Several central research contributions have investigated this relationship from an ontological and epistemological point of view (for example Orlikowski, 1992; Hanseth & Monteiro, 1996), drawing on social theories such as structuration theory or actor-network theory. Most IS researchers would agree that the key questions on the nature of the relationship and of 'technology agency' are not fully resolved (Rose *et al.*, 2005).

By focusing on IS project management theory and patterns, we acknowledge that our ambition is more modest, but our contribution is also more specific. Patterns are concerned with 'malleable' variables, i.e. such variables that managers can influence on (Majchrzak, 1984). The point of departure for this paper was that the IS project management field is in need for more knowledge on integration.

Our contribution, the four integration patterns and the analysis of these, addresses two important aspects of integration, namely the socio-technical and temporal dimensions. The patterns represent an extension to the research on dynamics of socio-technical integration (Kling & Scacchi, 1982; Mumford, 1983; Coakes *et al.*, 2000; Avgerou, 2002). This research emphasizes that a new IS must *link into* an existing socio-technical web of mutually dependent elements that cooperate to create value. The four patterns describe the managerial trade-offs for achieving these links. As illustrated in the SAS case, the integration process is non-linear and only partly controllable. Enrolment of stakeholders and technology from the social networks external to the development project is often outside the control of the project manager, who has to exploit windows of opportunity when these appear. Managing this very situated and context-dependent challenge is underrated in the IS project management literature.

The patterns were designed and investigated in the context of iterative SE. Our analysis shows that support for integration in SE frameworks is varying; in RUP (and other iterative frameworks) there is support for both stakeholder integration and technical integration patterns, but not for the combined socio-technical integration.

The temporal nature of the four patterns concerns a more subtle phenomenon; that critical events are linked to each other in ways that create trajectories that tend to continue once they have been established. In the SAS and OAG cases the patterns were never decided, but emerged after a breakdown situation. In the SAS case, the SAS business project manager engaged directly in software development after a breakdown of communication between developers and users. This made his priorities change from stakeholder integration to technical integration, thus changing – for a period of time – the pattern of the whole project. In the OAG case, the critical event was the establishment of the Methodology Group that triggered a basically unplanned dynamics in the project, described as the socio-technical integration pattern. These findings are related to the research on the importance of breakdowns (Ciborra, 2000), but they are also congruent with the findings of Newman and Robey who documented that the user-analyst relationships tended to create stable patterns of behaviour once they were established (Newman & Robey, 1992).

We also aim to bridge a gap between IS research and SE research. Acknowledging the complex non-linear and political nature of integration, the patterns incorporate existing knowledge in IS implementation research. But at the same time, the patterns address the practical challenges inherent in SE, thus relating directly to modern iterative process frameworks like RUP and MSF. While the normative project management literature (e.g. Cadle & Yeates, 2004) and SE frameworks (e.g. Jacobson *et al.*, 1999) treat risk management as a key discipline, integration challenges are underrated. Thus, the four integration patterns are also a contribution to the management of integration risks in SE research.

6.2 Implications for IS project managers

For practitioners, this research has two important implications. First, the four integration patterns together represent an analytical framework to be used in planning iterative software development projects. As the cases show, integration comes neither free nor easily and our analysis shows that there are both advantages and disadvantages for each pattern. The framework may help the practitioner to analyse his or her integration challenge and choose an appropriate pattern.

However, the patterns cannot be applied uncritically and care should be taken to evaluate the situations in which the patterns will be useful. Selecting the right integration pattern requires sensitivity to context. In Table 7 we summarize the criteria on which project managers should base their choice of integration pattern as analytical tool.

While the socio-technical integration pattern may appear most attractive, it might be dangerous to use in a setting with strong time pressure. As the SAS case shows, the stakeholder integration and the technical integration patterns require careful managerial trade-offs to balance project complexity against (internal) project schedule. To reduce complexity, the project manager may choose to concentrate on internal project tasks, postponing external integration issues. Sometimes the big bang pattern may prove beneficial in situations where the patterns for stepwise integration are simply not feasible. But an organization choosing the big bang pattern should also be aware of the organizational implementation risk.

Table 7. Integration pattern selection criteria

Integration pattern	Suitable context	Managerial trade-off
Big bang	Stepwise integration not feasible	Prioritizing project control, with the risk of user resistance and integration problems at start-up
Stakeholder integration	Simple technical solution in a complex organizational environment	Managing stakeholder complexity while maintaining project schedule
Technical integration	Technically complex solution in a stable organizational setting	Managing technical complexity while maintaining project schedule
Socio-technical integration	Organizational and technical flexibility	Allowing process dynamics, but at the risk of losing project control

Second, the four patterns may help IS project managers assess and manage the risks of the project. With the big bang pattern the internal development project risk is low, but organizational implementation risk is high. In the stakeholder integration and technical integration patterns, the internal development project risk increases, while there is a lower implementation risk. The socio-technical integration pattern lowers the organizational implementation risk, but increases greatly the complexity of the development project and hence also increases the risk associated with managing this complexity. As an example, the SAS project attempted to apply both the stakeholder integration pattern and the technical integration pattern in parallel. The criteria in Table 7 show that this is a risky approach, as the two integration patterns have a rather opposite selection criteria and apply in quite different contexts. This also explains some of the experienced problems in the SAS case, where stepwise integration became too complex to handle in the turbulent environment.

6.3 Limitations

Are the integration patterns also relevant for other non-iterative IS development projects? The research approach, LPR, tend to produce results that are strong on accuracy, but weak on generalization (Langley, 1999). This indicates that the primary area of usefulness of the socio-technical patterns is in-house IS development projects in large organizations, using iterative and incremental SE methodologies, with a clear business objective and an experienced development team. The patterns were designed and researched in this context, and only empirical research can answer whether this area can be expanded. As a point of departure, however, we can see no reason why the area of usefulness of the patterns should be limited to this context.

From a SE perspective one may criticize the four patterns for expanding the object of study too much. One might argue that it is unreasonable to expect that the IS project manager – in addition to the formidable task of producing high quality software – should also solve the integration challenges of the organization. In line with this, some authors have

argued that these challenges cannot be solved at project level, and suggested that they should be the responsibility of a high-level *organizational architect* (Sauer & Willcocks, 2004).

While we do agree that the integration challenge calls for complex interventions, we disagree that it can be solved at the top management level. As the cases illustrate, successful integration is closely linked to knowledge creation in development projects, not only decisions. The emergent nature of integration makes it an ongoing challenge which the IS project manager must address, whether he feels competent or not.

From the perspective of information infrastructure (Ciborra, 2000), the four patterns may be criticized for being too optimistic on the question of control. According to Ciborra, IS projects tend to drift, i.e. to deviate from plans. Trying to control this often leads to side effects, which make control even harder. Thus, from this perspective we cannot choose an integration pattern at will. Instead, we should build on existing structures and practices.

We agree that integration is hard to plan, and we have earlier in the paper argued that the patterns are context-dependent. However, the integration patterns are not planning templates ready to be used in IS development projects. They are primarily analytical constructs aimed at understanding the dynamics of integration, and thus helping the project manager to assess the risks of integration.

7. CONCLUSION

This paper started out by asking: what are the main patterns of integration in IS development projects? The theoretical approach was to view integration as the emergence of a socio-technical network.

The main finding is the suggested typology of the four generic integration patterns: big bang, stakeholder integration, technical integration and socio-technical integration. The patterns represent ideal types for how and when a development project should enrol and adapt stakeholders and technology, and stabilize the behaviour of the network. For IS research, the four integration patterns contribute to a socio-technical vocabulary of integration. Since integration is an issue for both SE and IS research, the four patterns also seek to bridge the gap between the two disciplines.

Second, careful analysis of the patterns in IS implementation and SE research and in the three large longitudinal cases shows that integration always comes at a price. Stepwise integration lowers the implementation risk, but increases greatly the complexity of the project. Thus, for practitioners, the four patterns represent a tool to analyse some of the risks of integration and to assess the managerial trade-offs.

The patterns are context-dependent. They were elicited and studied in the context of iterative systems development, where the success of the projects were associated with the integration of the organizational processes. Further research should study more closely how managers achieve successful integration, particularly in projects using other methodologies

than iterative. It could also test the validity of the patterns in other projects, especially where the context is not software development, but adapting and implementing business software packages.

ACKNOWLEDGEMENTS

An earlier and shorter version of this paper was presented at the European Conference of Information Systems (2005) in Regensburg. The extensive comments from the anonymous AE and reviewers of the *Information Systems Journal* were of great help in improving the paper. We also thank Jon Lanestedt for his work on the University case.

REFERENCES

- Alter, S. (1999) A general, yet useful theory of information systems. *Communications of the Association for Information Systems*, **1**, 1–69.
- Alter, S. (2000) Same words, different meanings: are basic IS/IT concepts our self-imposed Tower of Babel? *Communications of the Association for Information Systems* **3**, 2–87.
- Ambler, S.W. (1998) *Process Patterns. Building Large-Scale Systems Using Object Technology*. Cambridge University Press, Cambridge, UK.
- Avgerou, C. (2002) New socio-technical perspectives of IS innovation in organizations. In: *ICT Innovation: Economic and Organizational Perspectives*, Avgerou, C. & LaRovere, R.L. (eds), pp. 141–161. Edward Elgar, Cheltenham, UK.
- Avison, D. & Fitzgerald, G. (2003) *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill, London, UK.
- Avison, D. & Wood-Harper, A.T. (1990) *Multiview*. Blackwell, London, UK.
- Avison, D., Wood-Harper, A.T., Vidgen, R.T. & Wood, J.R.G. (1998) A further exploration into information systems development: the evolution of Multiview 2. *Information Technology & People*, **11**, 124–139.
- Beck, K. (2000) *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston, MA, USA.
- Blomberg, J. (2001) Using the RUP for enterprise e-business transformation. *The Rational Edge*.
- Boehm, B.W. (1988) A spiral model of software development and enhancement. *IEEE Computer*, **21**, 61–72.
- Bostrom, R.P. & Heinen, J.S. (1977a) MIS problems and failures: a socio-technical perspective, part I: the Causes. *MIS Quarterly*, **1**, 17–32.
- Bostrom, R.P. & Heinen, J.S. (1977b) MIS problems and failures: a socio-technical perspective, part II: the application of socio-technical theory. *MIS Quarterly*, **1**, 11–28.
- Bygstad, B. (2004) Controlling iterative software development projects: the challenge of stakeholder and technical integration. In: *Proceedings of the Thirty-Seventh Annual Hawaii International Conference on System Sciences*, 5–8 January, Big Island, HI, USA.
- Bygstad, B. (2005). Managing the dynamics of mutual adaptation of technology and organization in IS development projects. *Software Process: Improvement and Practice (SPIIP)*, **10**, 341–353.
- Bygstad, B. & Nielsen, P.A. (2005). Understanding and managing process interaction in IS development projects. In: *Proceedings of IRIS 28*, 6–9 August, Kristiansand, Norway.
- Cadle, J. & Yeates, D. (2004) *Project Management for Information Systems*. Prentice Hall, Harlow, UK.
- Chang, H.H. (2006) Technical and management perceptions of enterprise information system importance, implementation and benefits. *Information Systems Journal*, **16**, 263–292.
- Ciborra, C. (1997) De profundis? Deconstructing the concept of strategic alignment. *Scandinavian Journal of Information Systems*, **9**, 67–82.
- Ciborra, C. (2000) *From Control to Drift*. Oxford University Press, Oxford, UK.
- Coakes, E. & Elliman, T. (1999) The role of stakeholders in managing change. *Communications of the AIS*, **2**, 1–29.
- Coakes, E., Willis, D. & Lloyd-Jones, R. (2000) *The New SocioTech. Graffiti on the Long Wall*. Springer, London, UK.

- Davenport, T.H. (1993) *Process Innovation*. Ernst & Young, Boston, MA, USA.
- Giddens, A. (1987) *The Constitution of Society*. Polity Press, Cambridge, UK.
- Hanseth, O. & Lyytinen, K. (2004) Theorizing about the design of information infrastructures: design kernel theories and principles. *Sprouts: Working Papers on Information Environments, Systems and Organizations*, **4**, 207–241.
- Hanseth, O. & Monteiro, E. (1996) Inscribing behaviour in information infrastructure standards. *Accounting, Management and Information Systems*, **7**, 183–211.
- Hsiao, R.L. & Ormerod, R.J. (1998) A new perspective on the dynamics of information technology-enabled strategic change. *Information Systems Journal*, **8**, 21–52.
- IBM (2002) Rational Unified Process. 22 December 2006. [WWW document]. URL <http://www-306.ibm.com/software/awdtools/rup/> (accessed 22 December 2006).
- Jacobson, I., Booch, G. & Rumbaugh, R. (1999) *The Unified Software Development Process*. Addison Wesley, Reading, MA, USA.
- Kappelman, L.A. & McLean, E.R. (1994) User engagement in the development, implementation and use of information technologies. In: *Twenty-Seventh Annual Hawaii International Conference on System Sciences*, Maui, HI, USA.
- Klein, H.K. & Myers, M.D. (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, **23**, 67–94.
- Kling, R. & Scacchi, W. (1982) The Web of computing: computer technology as social organization. *Advances in Computers*, **21**, 1–90.
- Kruchten, P. (2000) *The Rational Unified Process*. Addison Wesley Longman, Reading, MA, USA.
- Lamb, R. & Kling, R. (2003) Reconceptualizing users as social actors in information systems research. *MIS Quarterly*, **27**, 197–235.
- Lanestedt, J. & Bygstad, B. (2003) Infrastructure for ICT-Enhanced Learning in Higher Education: a Generic Architecture. In: *E-Learn 2003: World Conference in Corporate, Government, Healthcare and Higher Education*, 7–11 November, Phoenix, AZ, USA.
- Langley, A. (1999) Strategies for theorizing from process data. *Academy of Management Review*, **24**, 691–710.
- Larman, C. (2004) *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley, Boston, MA.
- Latour, B. (2005) *Reassembling the Social*. Oxford University Press, Oxford, UK.
- Laudon, C.K. & Laudon, J.P. (2004) *Management Information Systems*. Pearson Education, Upper Saddle River, NJ, USA.
- Leonard-Barton, D. (1988) Implementation as mutual adaptation of technology and organization. *Research Policy*, **17**, 251–267.
- Lloyd, A.D., Dewar, R. & Pooley, R. (1999) Legacy Information Systems and business process change: a patterns perspective. *Communications of the AIS*, **2**, 1–41.
- Majchrzak, A. (1984) *Methods for Policy Research*. Sage Publications, Newbury Park, CA, USA.
- Majchrzak, A., Rice, R., Malhotra, A., King, N. & Ba, S. (2000) Technology adaptation: the case of a computer-supported inter-organizational virtual team. *MIS Quarterly*, **24**, 569–600.
- McManus, J. & Wood-Harper, T. (2003) *Information Systems Project Management*. Financial Times/Prentice Hall, Harlow, UK.
- Miles, M.B. & Huberman, A.M. (1994) *Qualitative Data Analysis*. Sage Publications, Thousand Oaks, CA, USA.
- Mingers, J. (2001) Combining IS research methods: towards a pluralist methodology. *Information Systems Research*, **12**, 240–259.
- Monteiro, E. (2000) *Actor-Network Theory and Information Infrastructure. From Control to Drift*. Oxford University Press, Oxford, UK.
- Mumford, E. (1983) *Designing Human Systems*. Manchester Business School, Manchester, UK.
- Nandhakumar, J. (2002) Managing time in a software factory. *The Information Society*, **18**, 251–262.
- Newman, M. & Robey, D. (1992) A social process model of user-analyst relationships. *MIS Quarterly*, **16**, 249–266.
- Ngwenyama, O. (1998) Groupware, social action and emergent organizations: on the process dynamics of computer mediated distributed work. *Accounting, Management and Information Technologies*, **8**, 127–146.
- Orlikowski, W. (1992) The duality of technology: rethinking the concept of technology in organizations. *Organization Science*, **3**, 398–427.
- Pettigrew, A.M. (1985) Contextualist research and the study of organizational change processes. In: *Research Methods in Information Systems*, Mumford, E., Hirschheim, R., Fitzgerald, G. & Wood-Harper, A.T. (eds), pp. 53–78. North-Holland, Amsterdam, The Netherlands.
- Pettigrew, A.M. (1990) Longitudinal field research on change theory and practice. *Organization Science*, **3**, 267–292.
- Rose, J., Jones, M. & Truex, D. (2005) Socio-theoretic accounts of IS: the problem of agency. *Scandinavian Journal of Information Systems*, **17**, 133–152.

- Royce, W. (1998) *Software Project Management*. Addison-Wesley Longman, Reading, MA, USA.
- Sauer, C. & Willcocks, L.P. (2004) Strategic alignment revisited: connecting organizational architecture and IT infrastructure. In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 5–8 January, Big Island, HI, USA.
- Stapleton, J. (2003) *DSDM: Business Focused Development*. Addison-Wesley, Harlow, UK.
- Wainwright, D. & Waring, T. (2004) Three domains for implementing integrated information systems: redressing the balance between technology, strategic and organizational analysis. *International Journal of Information Management*, **24**, 329–346.
- Walsham, G. (1997) Actor-network theory and IS research: current status and future prospects. In: *Information Systems and Qualitative Research*, Lee, A.S., Liegtenau, J. & DeGross, J.I. (eds), pp. 466–480. Chapman & Hall, London, UK.
- Webster's (1986) *Webster's Third New International Dictionary*. Encyclopedia Britannica, Chicago, IL, USA.
- Improvement and Practice, Electronic Government, International Journal of Networking and Virtual Organizations and Information Resource Management Journal.*

Peter Axel Nielsen holds an MSc degree in computer science from Århus University, Denmark, and a PhD in information systems from Lancaster University, UK. He is currently Associate Professor in computer science at Aalborg University, Denmark. He is also a Visiting Research Professor with Agder University College, Norway, and a Visiting International Research Scholar with Virginia Commonwealth University, Richmond. His research interests include action research on software development practice, object-oriented methodologies, and social and organizational aspects of software development. He is a co-author of *Object-Oriented Analysis & Design and Improving Software Organizations: From Principles to Practice* (Marko Publishing, Aalborg, 2000).

Bjørn Erik Munkvold is Professor of Information Systems at University of Agder in Kristiansand, Norway. His main research interests are implementation and use of e-collaboration technologies, and information systems research methodology. He has published in journals such as *Communications of the AIS, Database for Advances in Information Systems, European Journal of Information Systems, Group Decision and Negotiation, IEEE Transactions on Professional Communication, Information & Management, Information Systems Management, and Journal of Management Information Systems*. He has authored a book entitled *Implementing Collaboration Technologies in Industry: Case Examples and Lessons Learned* (Springer Verlag, London, 2003). Professor Munkvold serves as Co-Editor of the *Scandinavian Journal of Information Systems*, and Associate Editor for the *International Journal of e-Collaboration*.

Biographies

Bendik Bygstad holds a PhD in computer science from Aalborg University and a Master degree in sociology from the University of Oslo. He worked for 15 years in the computer industry, mostly as an IT manager. He is currently an Associate Professor at the Norwegian School of Information Technology. His main research interest is the relationship between SW development processes and organizational change. Other research interests include actor-network theory, and ICT-based service innovation. He has published articles in *International Journal of Technology and Human Interaction, Software Process:*