Smolander, Kari; Rossi, Matti; Pekkola, Samuli

Collaboration change in enterprise software development

Published in:
Proceedings - 9th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2016

DOI:
10.1145/2897586.2897590

Published: 14/05/2016

Document Version
Peer reviewed version

Please cite the original version:
Smolander, K., Rossi, M., & Pekkola, S. (2016). Collaboration change in enterprise software development. In Proceedings - 9th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2016: CHASE 2016 (pp. 68-74) https://doi.org/10.1145/2897586.2897590

Collaboration Change in Enterprise Software Development

Kari Smolander
Aalto University
Department of Computer Science
P.O.Box 15400, FI-00076 Aalto
+358 40 546 3493
kari.smolander@aalto.fi

Matti Rossi
Aalto University
School of Business
P.O. Box 21220, FI-00076 Aalto
+358 50 383 5503
matti.rossi@aalto.fi

Samuli Pekkola
Tampere University of Technology
Dept. of Information Management
P.O.Box 541, FI-33101 Tampere
+358 40 586 0791
samuli.pekkola@tut.fi

ABSTRACT

Enterprise software development is a complex effort that may last years. Enterprise software is often developed by a systems integrator that makes modifications to a pre-made package or builds tailored software for the specific purpose. The development may include many developer organizations, the user organization, and their different departments and sub-units. Their collaboration evolves through project incidents, phases and even crises. The practices of project management, communication, contracts, and ultimately personal relationships change intentionally or unintentionally. These changes may cause uncertainties and discontinuities for the development. This study observes changes during enterprise software development and their influence on collaboration practices in different situations. During twenty years of development both internal and external crises and changes in the business environment triggered changes in collaboration. The collaboration practices are classified with four modes of collaboration (contract, cooperation, personified, and process) that illustrate emphasis in collaboration in different circumstances.

CCS Concepts

Software and its engineering → Software creation and management → Collaboration in software development

Information systems → Information systems application → Enterprise information systems

Keywords

Collaboration; collaboration change; enterprise software; project management;

1. Introduction

Enterprise software is developed in complex networks of parties. A new system is needed by the adopting organization, and at least its IT department and a business unit. The software is typically developed by another organization with the adequate technical expertise and infrastructure for the development [1]. The enterprise system may also be based on a pre-made software package that needs to be integrated to the adopting organization by an integrator organization [2], [3]. Furthermore, the network can include any number of subcontractors and other simultaneous projects with a variety of stakeholders.

CHASE'16, May 16 2016, Austin, TX, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4155-4/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2897586.2897590

The collaboration practices and relationships between adopters and developers are not static or rigid. Various crises, incidents and project phases may change the dynamics and practices of the development, including for example project management practices, communication patterns, contracts, and personal relationships. These changes may cause uncertainties and discontinuities in the development. However, quite little is known about these changes, their triggers, and their consequences. Some examples include Newell et al. [4] and Nandhakumar et al. [5], but their focus is rather on short-term changes and survival strategies than on evolving long-term relationships.

Our study explores how collaboration between an enterprise software developer and a client has evolved over twenty years, both in response to external shocks and to internal issues in the project organization. We have identified patterns of collaboration in the development of an enterprise system over its whole life-cycle, from concept to near retirement. Our findings illustrate four different modes of collaboration (contract, cooperation, personified, and process), each being the main driver of development under different circumstances. This knowledge is of value when improving inter-organizational practices for enterprise software development.

2. Research area

Enterprise systems aim at integrating information flows across the organization to increase the organization's competitiveness [6]. The implementation and continuous use of enterprise systems are studied for example through different types of critical success factor (CSF) studies (see for example [7] and [8]). In these studies development is often perceived as a linear process with certain starting and ending points and a set of separate phases, where certain conditions have to be met. However, already Robey et al. [9] suggested that instead of linearity, the development is a continuous dialectical learning process, especially in large-scale enterprise software development.

In software engineering collaboration has been studied for a long time, at least since the pioneering study by Curtis & al. [10]. Collaboration has been approached as a general phenomenon, "people factor" [11] for example from the viewpoint of processes [12], communities or other social units [13], global software engineering [14], and agile development [15]. Despite of this general interest to the "people factor", there is little evidence available about specifics of collaboration, such as its change over time and its different modes in large software development efforts including enterprise software development. The collaborative practice of enterprise development can be easily disturbed [16], but still collaborative patterns and their changes have been rarely studied [17]. In this study, we refer to collaboration as a practice where at least two parties work together to achieve a common goal or co-create value in the form of enhancing the software-based enterprise system [18].

Previous research has assumed that the development model is decided in the beginning and then it stays more or less as planned until enterprise software is released, transferred to maintenance, and abandoned. However, the experience reports of large-scale development give a more complicated view of the messy process. Understanding how the development process and developer relationships evolve over time is an important topic to study.

3. Research process

This paper presents an inductive case study using Grounded Theory (GT) as the research method [19]. This kind of inductive approach is suitable for approaching the essence of a complex organizational phenomenon [20]. Enterprise software development includes both a technical and a social component and it requires the understanding of the stakeholders and their interactions. We have investigated an enterprise software development process in one very large and long project in a global manufacturer that has built a custom ERP system for the management of its value chain.

An inductive case study with Grounded Theory is an iterative and systematic process [19]. The data analysis is based on Strauss and Corbin's version [21] of GT that consists of three coding procedures: open, axial and selective coding. In open coding, the data is first given conceptual labels with descriptive codes. Open coding also includes constant comparison between the pieces of data (other open codes), so that similarities and differences can be found. In axial coding, the codes may be classified to categories. Connections between the codes and categories are also created. This includes interpretation of the codes and categories, and their properties. The purpose of this refining activity is to make the constructs more abstract and theoretical [22]. In selective coding, an overarching theoretical viewpoint to the data is selected. The observations are then explained from this viewpoint, also referred to as a core category.

3.1 Data collection

Factory is a global manufacturer of materials and common goods. Its annual turnover is over 8 billion euros. It has operations, manufacturing and sales in all continents. In the beginning of the 1990s, Factory realized that it needed to renew its sales and logistics systems. *Birdie*, a fully-customized ERP system for sales and logistics was decided to be built to replace legacy systems and to overcome the year 2000 problem.

The data was collected with theme-based interviews between February and May 2013. The data collection started by discussions with our contact person from upper management in the user organization. The research goals were briefly presented in order to identify the right persons to interview. The snowballing technique [21], in which the next interviewee is a referral from the previous one, was used. In selecting the interviewees we especially emphasized persons who had important responsibilities and experiences in various parts of the long enterprise software development project and maintenance period.

The interview questions were open-ended, focusing on the interviewee's experiences in the development project. They included only a few general questions, which then led to a more detailed discussion, depending on the interviewee and her/his background and answers. All interviewees were very willing to describe their experiences and history in the project. We thus received vivid stories about events and incidents. The interviewees shared with us their interpretations of their meaning and importance. Their stories included also the timings and orders of the events, and their interpretations of causal connections between the circumstances, events and their consequences. This enabled us to form a triangulated and combined narrative of the project and the changes that had happened in collaboration.

We interviewed 17 persons (Table 1) from *Factory* (the user organization), *Integrator* (the developer organization) and *Middleware consulting* that were involved in the development of the system, *Birdie*. The interviews lasted from 26 to 73 minutes, the average being 45 minutes.

Table 1: Roles and organizations of the interviewees

Role	Organization	Responsibility in the project
Corporate IT manager	Factory	Project leader 1995-2000
IT manager of a business area	Factory	Project and maintenance management since 1995
Program manager	Factory	Business analyst 1995-2002, business analyst in an interfacing system 2002-2010
Enterprise architect	Factory	Responsibilities of interfacing systems 1999-2010, development and maintenance 2010-
Representative of sales	Factory	Analyst at Integrator 1995-1997, area responsibility since 1997
IT support manager	Factory	Area responsibility since 1995, rollout responsibilities
Representative of logistics	Factory	Area responsibility 1995-2006, now working with a related logistics provider
Corporate IT manager	Factory	Project leader 2000-2004, interfacing systems responsibility 2004-2009
Vice president	Integrator	Birdie project manager at Integrator 1999-2005
Service owner	Integrator	Technical support 2001-2009, maintenance service responsibility 2009-
Continuous service manager	Integrator	Analyst 2004-2011,maintenance manager 2011-
Infrastructure manager	Integrator	Hardware and facility responsibility 1995-2001.
Project manager	Integrator	Birdie project manager at Integrator 1995-1997
Lean software developer	Integrator	Birdie developer since 1999
Service manager	Integrator	India-based offshore maintenance manager 2011-
Middleware manager	Middleware consulting	Middleware consultant 1996-1998
Technical consultant	Middleware consulting	Middleware consultant 1996-1998

3.2 Analysis

All the interviews were recorded and later transcribed for easier analysis. Atlas.ti was used as the coding tool, as it provides easy-to-use functionality for managing text and attaching codes to portions of text. First, we made a closer scrutiny of the incidents in the ERP development project. The particulars of each incident were coded inductively, without any a priori analysis framework, as required by the Grounded Theory methodology [20], [21]. Our inductively built coding scheme included conditions, decisions and individual events related to each project incident, in combination with the interviewees' presumptions and the effects of the incidents. This coding produced in total 200 codes.

In the next phase of coding, the relationships between the codes and categories were identified. Also some new categories based on these relationships were formed. Each project event and decision was scrutinized, and all related codes were connected to each other. For example, specific issues in the tailor-made system created performance problems, which, in turn, required a decision on architecture reorganization to be made.

Our interest fell into the collaboration between the main partners of the ERP development project. It seemed to be important to understand how the collaboration evolved throughout the project incidents. The project, described in detail below, was complex, with many crises related to the coded incidents. This led us to select "collaboration change" as the core category. With the aid of the dimensions that we had deemed as essential, processuality and cooperation, we identified four modes of collaboration that explained how collaboration changed in this enterprise software development project.

4. Modes of inter-organizational collaboration in Birdie development

We analyzed how the incidents in Birdie's development emerged, and what kind of decisions and actions the user organization made as a reaction to them. Most incidents meant major changes in the collaboration between the main parties; Factory and Integrator. We were able to identify patterns of change after and during the identified project incidents. The analysis continued by explaining why and how the collaboration change occurred and what determined the direction of the change.

The important categories related to collaboration change were contracts, cooperation, personification, and processes. The more we looked into the data, the more plausible it seemed that there were four different and independent "modes" through which collaboration took place. These four modes appeared to exist simultaneously, from the beginning of the project until the end. However, their relative emphasis changed in response to the project incidents. The four modes of collaboration are:

- **Contractual mode**: Collaboration is defined by legal contracts between the parties. The project incidents and their consequences are handled according to formal business contracts that define roles and responsibilities.
- **Cooperative mode**: Collaboration is based on mutual interests and voluntary cooperation. The parties approach incidents and their consequences by their common points of interest, searching for a solution beneficial to both parties.
- **Personified mode**: Collaboration happens between individuals. The incidents and their consequences are dealt with by the key persons and the relationships, trust and acknowledgement of expertise between the key persons have developed over years of joint collaboration.
- **Process mode**: Collaboration is a planned and designed process. The incidents and their consequences are resolved through this process that determines the parties and procedures. Typical defined processes are those related to change and quality management.

The following quotation expresses the Contractual mode in 1996-1998. It describes the status during a project crisis when one of the challenges was to get rid of the too rigid contract, and follow the Cooperative mode:

”*I think it was the most challenging point. To give up the initial mindset that you have the stuff and we have the money. You'll deliver the stuff and we'll pay you. We have this agreement, which juristically binds us to do things. You just have to obey it.*" (Corporate IT Manager, Factory).

In the development of Birdie, the incentive to be cooperative materialized in a crisis. The Cooperation mode went so far that the project organizations in Factory and Integrator were de facto merged without an explicit renegotiation of the contract:

”*The main element was that we couldn't continue as earlier. That there were two separate projects: the customer having one and the vendor having another one, both with own agendas etc. So I decided to establish a joint project. I set its steering group and I continued as its chairman, and as being in charge of this whole thing overseeing everything*" (Corporate IT Manager, Factory)

Taking the Personified mode may also mean that heroic individual accomplishments and individual expertise are emphasized. This also happened when solving the technological crisis:

”*… I had this personal relationship. I realized that we're going to ride for a fall. So I invited that guy here. He flew over here on a morning plane, we sat in [the local restaurant], and I told everything. I had all the documents and I told which is which. Then we had lunch. After this he said that we are really in trouble, but he is going to help us out there.*" (Corporate IT manager, Factory)

Later, when the crisis was resolved, the importance of recurring processes grew. The Process mode was adopted especially in change and quality management. However, the switch from the Cooperative mode to the Process mode was not easy:

"*When I came in, I thought it was chaos. Like I said, nobody knew how many change requests there were, what kind and where they were. They were nowhere, they were in different places. Then we made it systematic. We made the whole testing model, the whole change management, how to make new releases, how many weeks can we use [Integrator] and where, how much do they do, where are the acceptance criteria, how many changes we may take in.*" (Project manager 2, Factory)

5. Project incidents and collaboration

A project incident is a major event during a project, such as a crisis, an internal or external event changing the practices and/or relationships, technology change, architecture change, project organization change, and a major requirement change.

Table 2 lists chronologically the most important incidents related to the development of Birdie, which are described in detail below.

Table 2. Main project incidents

Year	Incident	Description
1995	Decision to build a custom system	Factory compares packaged ERPs with its requirements and decides to build a custom system.
1995	Decision to build a general product for the industry field	Integrator aspires new markets by a decision to build a general product that is based on Birdie.
1996-1998	Project start and technological crisis	Full-scale development starts. Very soon it reaches a crisis related to the selected technological approach.
1998	Architecture reorganization	The technological crisis is solved by a cooperation of Factory and Middleware consultants.
1998	Merger	Factory merges with its competitor of about the same size.
2000-2004	Rollouts	Birdie is installed and taken into use globally in about 50 production sites and sales offices.
2000	Abandonment of product development	Integrator concludes that it is not possible to make a general product of Birdie because of its very Factory-specific features.

2000-2002	Move to maintenance mode	New development is gradually replaced by a more maintenance-oriented development.
2002, 2008	Change of technology	The server operating systems and the database management system are changed for commercial reasons.
2006, 2010	Offshoring	Maintenance and further development are offshored first to Europe, and later to India for cost reasons.

5.1 Decision to build a custom system (1995)

The development of Birdie started in the beginning of the 1990s by an initial study of requirements, and how packaged ERP systems could support them. The conclusion was that no existing ERP package could support the desired functionality and business processes well enough. Factory decided to build a unique system fully tailored to its needs. It was evident that Integrator would deliver the system. A part of Integrator originated from the IT department of Factory. They had also built many of the operative legacy systems that the new system would replace. Thus, there were already established collaboration practices and close personal relationships between Factory and Integrator, on many levels and through many systems developed and maintained.

Dominant modes: Cooperation mode and Personified mode

Incident features essential to collaboration mode:

- Contract between Factory and Integrator based on known requirements [PROMOTES Contract mode]
- Ethos of user-drivenness between Factory and Integrator [PROMOTES Cooperation mode, PROMOTES Personified mode]
- Established ways of working between Factory and Integrator [PROMOTES Cooperation mode, PROMOTES Process mode]
- Do-it-yourself attitude especially in Factory [PROMOTES Personified mode]

5.2 Integrator wants to build a general product for the industry field (1995)

The business domain Factory operated had a strategic importance for Integrator, which aimed at a global presence. Integrator grasped an opportunity to build a general product on Birdie, so that it could be offered other global enterprises in that business domain. This increased complexity to the project and created hidden motives and agendas to the collaboration. While development was previously done with common practices, good will, and personal relationships, now a partly opaque component that speculated on future benefits entered. This obscurity reflected especially on the selection of development tools and libraries. It was perceived important that no license fees for tools and libraries were included, and that they were selected for long-term product development with portability, and not for rapid and flexible customer-specific implementation.

Dominant mode: Contract mode

Incident features essential to collaboration mode:

- Integrator wants to secure its legal position [PROMOTES Contract mode]
- Integrator's objectives were partly hidden to Factory [DEMOTES Cooperation mode]
- Integrator's technology and product experts have the major role [PROMOTES Personified mode]

5.3 Project start and technological crisis (1996-1998)

The requirements and business processes were reasonably well understood by both Factory and Integrator, but the selected technology was unstable. Software and systems development in the 1990s can be characterized by immature technologies, constant technology change, and new tools and platforms for development. Birdie was originally developed as a POSIX compliant C++ application for Windows NT clients, UNIX servers and a relational database that replicated over, at that time not-so-reliable, data communications networks. The platform included a homemade class library for fat clients and a proprietary messaging service that was planned to be used globally for transactions. The messaging service was performing well, but the client class library was still immature when the implementation of Birdie started. In addition, most of the developers were not familiar with the platform. Their earlier experience was mostly in character-based UNIX or MPE systems. All this accumulated, causing slowness in the start and confusing the developers.

The biggest obstacle was poor performance caused by wrong architectural choices. The selected architecture with fat clients and extensive interactions with database servers could not produce the required performance. The client library generated too much interaction with the database servers because of a fundamental flaw in the architecture design. This was a severe business crisis to Integrator and the awareness of crisis grew both in Integrator and in Factory. The project costs started to grow and the schedules to slip. Factory understood that the contract made with Integrator was overly optimistic and would not substantiate as such.

Dominant modes: Cooperation mode, Personified mode

Incident features essential to collaboration mode:

- Factory becomes aware of its overtrust to Integration [PROMOTES Contract mode]
- The original contracts are found to be unrealistic [DEMOTES Contract mode]
- Awareness of crisis [PROMOTES Cooperation mode]
- The technology change is driven by the customer [PROMOTES Cooperation mode]
- Integrator's business is in a crisis [PROMOTES Cooperation mode]
- Solutions are initiated through personal relationships [PROMOTES Personified mode]
- The crisis is managed in an ad-hoc manner [PROMOTES Personified mode, DEMOTES Process mode]
- The whole project is reorganized [DEMOTES Process mode]

5.4 Architecture reorganization (1998)

Factory's active problem solving contributed to overcoming the crisis. The project manager at Factory took the lead, and by using his personal connections, hired external middleware consultants to redesign the architecture. This caused another crisis in Integrator, where many of the key technology experts decided to resign.

This incident and its recovery can be characterized as a phase of improvisation, intensive cooperation and personal contribution. We heard many stories about how the crisis was solved through personal contribution and cooperation in reorganizing the code and testing the new architecture in real situations. Factory also understood that to overcome the crisis, it must provide some new benefits to Integrator. Compensation in the original contract was not viable for Integrator in the business sense. This understanding, with the very good business cycle in the field, made it possible to emphasize collaboration in the project. Both Factory and Integrator had a common goal: to finish the project successfully.

Dominant mode of the incident: Personified mode

Incident features essential to collaboration mode:

- Factory is aware of its overtrust to Integrator [PROMOTES Contract mode]
- Factory uses external solvers to core problems [PROMOTES Contract mode]
- There is competition between developers and consultants [PROMOTES Contract mode, DEMOTES Cooperation mode]
- Factory is ready to compensate Integrator losses [PROMOTES Cooperation mode]
- Problem solving is driven by the customer [PROMOTES Cooperation mode]
- Expertise is acquired through personal relationships [PROMOTES Personified mode]
- High valuation of personal "world-class" expertise [PROMOTES Personified mode]
- Factory management is confident in risk-taking [PROMOTES Personified mode]
- Some key developers resign from Integrator [DEMOTES Personified mode]
- Trial and error approaches are used extensively [DEMOTES Process mode]
- Some developer beliefs are unrealistic [DEMOTES Process mode]
- A general awareness of crisis prevails [DEMOTES Process mode]

5.5 Merger (1998)

While Birdie was moving gradually back on the track, new external and unexpected events created concerns. Factory decided to merge with another international company of about the same size. This was a source of many kinds of uncertainties that had to be handled. After the merger it was not clear which system, Birdie or its counterpart at the other company, would be selected.

The headquarters of the other company was in another country. Many mentioned that there was a feeling of competition between the countries. The process of selecting the system to handle the core value chain of the new Factory included also a political component, as both former organizations in the new Factory wanted to have a system for their interests. In this discussion, Birdie was saved by the fact that it did not have an evident and ready-to-be-used alternative. It also got some unexpected external support from another large enterprise, which selected the very same middleware solution simultaneously - despite the fact that some believed it would already be an obsolete technology.

Dominant modes of the incident: Contract mode, Process mode

Incident features essential to collaboration mode:

- Added responsibilities of merged IT [PROMOTES Contract mode]
- More competition among developers and consultants, more external parties involved [PROMOTES Contract mode]
- Use of external consultants to confirm decisions [PROMOTES Contract mode]
- Competing systems in merged organization [DEMOTES Cooperation mode]
- New fractions and parties in the organization, more stakeholders involved [DEMOTES Cooperation mode, DEMOTES Personified mode, PROMOTES Process mode]
- Cultural differences in merged parts [DEMOTES Cooperation mode]
- Issues of location and ownership in IT and systems [DEMOTES Cooperation mode]
- Decision points and decisive events with competing parties [PROMOTES Personified mode]
- Increase in the scale of the system, added requirements [DEMOTES Personified mode, PROMOTES Process mode]
- Increased awareness of process development needs [PROMOTES Process mode]

5.6 Rollouts (2000-2004)

By the end of 2004, Birdie had been installed and deployed at 33 European sites. Factory and Integrator together needed to create explicit processes for testing, change management, and rollout implementation. It was no longer possible to deploy rollouts sequentially but they needed to be made in parallel. This generated additional needs for defined processes.

The rollouts were managed and organized by Factory with experts from Integrator for their implementation, with a spirit of collaboration. At that time, Factory hired a new project manager for Birdie that observed the lack of change and quality management processes. There was a clear need to "professionalize" recurring activities. ITIL was seen as a move to professional practices in change and quality management.

Dominant mode of the incident: Process mode

Incident features essential to collaboration mode:

- A customer-driven deployment process [PROMOTES Cooperation mode]
- Clear management support for the operations [PROMOTES Process mode]
- Need to develop processes for the recurring deployment activity [PROMOTES Process mode]

5.7 Integrator abandons product development (~2000)

Roughly in 2000, Integrator came to a conclusion that Birdie will be a unique solution without any new product opportunities. This decision had an effect on processes and attitudes at Integrator. There was less need to focus on intellectual property rights, the goals were less diverse, and customer-specific processes were easier to promote.

Dominant mode of the incident: Cooperation mode

Incident features essential to collaboration mode:

- Integrator has less needs to protect intellectual property rights [DEMOTES Contract mode]
- The goals of Integrator and Factory are better aligned [PROMOTES Cooperation mode]
- Specific customer requirements are easier to implement [PROMOTES Cooperation mode]
- Customer-specific development processes are easier to implement [PROMOTES Process mode]

5.8 Move to maintenance mode (2000-2002)

In 2000, a decision to end the development project and move Birdie to a maintenance organization was made. However, there were major development efforts ongoing until 2002. As releasing new versions, bug fixes and features to a very large install base was difficult and complex, and emphasized testing requirements, ITIL [23] was taken as the basis for maintenance processes also.

Dominant mode of the incident: Process mode

Incident features essential to collaboration mode:

- Use of standard support processes [PROMOTES Contract mode, PROMOTES Process mode]
- The status of "business-as-usual". The feeling of crisis dissolves [DEMOTES Cooperation mode]
- Complex version and change management, very large install base [DEMOTES Personified mode, PROMOTES Process mode]
- Testing requirements are emphasized [DEMOTES Personified mode, PROMOTES Process mode]

1.1 Change of technology (2002, 2008)

In 2002 and 2008 the database and application servers of Birdie were changed. Both changes were made due to cost reasons. The database technology was changed after Factory felt that the provider became greedy and wanted to gain too much profit. Both changes were moves to a more impersonal direction in relation to technology. In both cases the new technology was provided as a standard product by Microsoft.

Dominant mode of the incident: Contract mode

Incident features essential to collaboration mode:

- Cost control, especially of licenses [PROMOTES Contract mode]
- Standard technology, less specific expertise needed [DEMOTES Personified mode]
- Greediness of the technology provider [PROMOTES Contract mode, DEMOTES Personified mode]

5.9 Offshoring (2006, 2010)

The costs were also a determining factor when Factory decided to offshore the maintenance of Birdie. Integrator nearshored the maintenance to a cheaper country in Europe in 2006 and later offshored it to India (in 2010). Both changes added new requirements for the processes for managing changes and quality. The old way of maintaining personal relationships between Factory and Integrator was no longer possible.

Dominant modes of the incident: Contract mode, Process mode

Incident features essential to collaboration mode:

- Focus on costs [PROMOTES Contract mode, DEMOTES Cooperation mode]
- Scaling down of maintenance organization [PROMOTES Contract mode]
- Frequent personnel changes [PROMOTES Contract mode, DEMOTES Cooperation mode, DEMOTES Personified mode, PROMOTES Process mode]
- Longer distances between parties [DEMOTES Personified mode]
- Offshoring requires clear and defined processes [PROMOTES Process mode]

5.10 Current status

At the time of the interviews, Birdie was used widely in Factory. It was also considered as a success, in spite of problems. However, many considered it probable that Birdie will be replaced with a standard packaged product in the future.

6. Discussion

The history of Birdie can be explained with the alteration of four modes of collaboration; Contract, Cooperative, Personified and Process modes. Each of these was emphasized differently in and after each project incident. Figure 1 characterizes these four modes by defining their essential features.

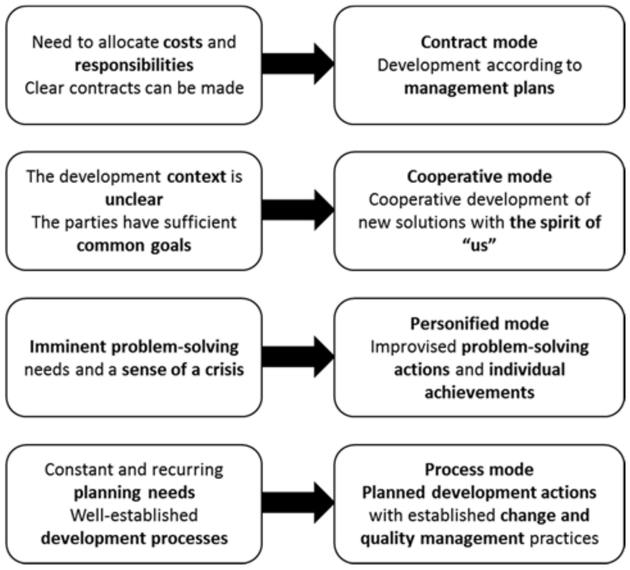
Need to allocate costs and responsibilities
Clear contracts can be made
Contract mode
Development according to management plans
The development context is unclear
The parties have sufficient common goals
Cooperative mode
Cooperative development of new solutions with the spirit of "us"
Imminent problem-solving needs and a sense of a crisis
Personified mode
Improvised problem-solving actions and individual achievements
Constant and recurring planning needs
Well-established development processes
Process mode
Planned development actions with established change and quality management practices

Figure 1. Summary of the collaboration modes

A mode can be seen as a reaction. The Contract mode is a reaction to the need to define the division of costs and responsibilities as early and as clearly as possible. The cooperative mode is taken when the context is somehow unclear, such as in the technological crisis of Birdie. The Personified mode is often a reaction to imminent problem-solving needs requiring improvisation and ad-hoc actions. Any long-lasting "normal" situation emphasizes the planning needs, which can be solved by taking the Process mode.

Each mode also emphasizes different things in development. The Contract mode sees systems development as a management-induced action emphasizing plans and commitments. Cooperative mode requires extensive cooperation for realizing new solutions. The Cooperative mode emphasizes the spirit of "us" in cooperative action. In the Personified mode, improvised problem-solving actions are regular and individual achievements are emphasized. The Process mode is business-as-usual, where planned development actions, such as change and quality management, are executed.

Finally, each mode has different pre-requisites. The Contract mode requires clear contracts. It is not possible to adopt the Cooperative mode without identifying at least some common goals. The Personified mode requires influential persons who are committed and able to use their influence, skills, and personal relationships. Naturally, defined and implemented processes are required for the Process mode.

The four modes existed simultaneously in the development of Birdie, although their relative emphasis varied a lot over the history. During and after any incident, the project organization may start to emphasize another mode. This transfer may happen also without any deliberate decision. A reaction to an incident may follow with a changed awareness of new kinds of requirements related to collaboration in the project. From the history of Birdie, we could explicitly identify moves between all four modes.

There are potential validity threats when studying things in the past. Our study is in a sense longitudinal, but the data has been collected during a limited time period. However, time in our study is a social construction, where "critical is not just events but the underlying logics that give events meaning and significance" [24, p. 273]. Our approach does not require an exact definition and measurement of time units, boundaries, periods and timelines. Instead, essential are the causes and consequences of events which we were able to triangulate over the multiple viewpoints, perspectives and experiences in the interviews.

The produced classification is not directly generalizable as such. However, the four identified collaboration modes provide a general-level explanation of how collaboration changed in and after project incidents in large-scale enterprise software development. This finding does not provide direct testable propositions, but it can guide further research and practice improvement in enterprise software development.

7. Conclusions

We observed the changes in collaboration during large-scale strategic software development. In order to succeed in a prolonged project and collaboration, the collaboration practices between the parties have to be able to evolve in response to different project incidents. We identified four modes of collaboration, Contract, Cooperation, Personified and Process modes, which differ in their emphasis, requirements and ways of working, and are reactions to different situations. Furthermore, we identified changes to the main collaboration mode over the course of the project.

Collaboration in large enterprise software projects is never static and rigid. Instead, different incidents force project parties to adjust their mode of collaboration dynamically to a new situation and its requirements. The identification of these modes and guidance on when to apply them is valuable in practice.

8. ACKNOWLEDGMENTS

This research was funded by Academy of Finland grant #259454.

9. REFERENCES

[1] J. Y. T. Chang, E. T. G. Wang, J. J. Jiang, and G. Klein, "Controlling ERP consultants: Client and provider practices," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1453–1461, May 2013.

[2] S. Sawyer, "A market-based perspective on information systems development," *Communications of the ACM*, vol. 44, no. 11, pp. 97–102, 2001.

[3] S. Sawyer, "Packaged software: implications of the differences from custom approaches to software development," *European Journal of Information Systems*, vol. 9, no. 1, pp. 47–58, Mar. 2000.

[4] S. Newell, C. Tansley, and J. Huang, "Social Capital and Knowledge Integration in an ERP Project Team: The Importance of Bridging AND Bonding," *British Journal of Management*, vol. 15, no. S1, pp. S43–S57, 2004.

[5] J. Nandhakumar, M. Rossi, and J. Talvinen, "The Dynamics of Contextual Forces of ERP Implementation," *Journal of Strategic Information Systems*, vol. 14, no. 2, pp. 221–242, 2005.

[6] T. H. Davenport, "Putting the Enterprise Into The Enterprise System," *Harvard Business Review*, vol. July/ August, pp. 121–131, 1998.

[7] H. Akkermans and K. van Helden, "Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors," *European Journal of Information Systems*, vol. 11, no. 1, pp. 35–46, Mar. 2002.

[8] C. P. Holland and B. Light, "A Critical Success Factors Model For ERP Implementation," *IEEE Software*, vol. 16, no. 3, pp. 30–36, 01-May-1999.

[9] D. Robey, J. W. Ross, and M.-C. Boudreau, "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change," *Journal of Management Information Systems*, vol. 19, no. 1, pp. 17–46, Jul. 2002.

[10] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Communications of the ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[11] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131–133, Nov. 2001.

[12] I. Richardson, V. Casey, F. McCaffery, J. Burton, and S. Beecham, "A Process Framework for Global Software Engineering Teams," *Information and Software Technology*, vol. 54, no. 11, pp. 1175–1191, Nov. 2012.

[13] D. A. Tamburri, P. Lago, and H. van Vliet, "Organizational social structures for software engineering," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–35, 2013.

[14] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "An empirical study of global software development: distance and speed," in *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Ontario, Canada: IEEE Computer Society, 2001, pp. 81–90.

[15] T. Dingsøyr and N. B. Moe, "Research challenges in large-scale agile software development," *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 5, pp. 38–39, 2013.

[16] S. Sarker and A. S. Lee, "Using a case study to test the role of three key social enablers in ERP implementation," *Information & Management*, vol. 40, no. 8, pp. 813–829, 2003.

[17] Y. Dittrich, S. Vaucouleur, and S. Giff, "ERP Customization as Software Engineering: Knowledge Sharing and Cooperation," *IEEE Software*, vol. 26, no. 6, pp. 41–47, 2009.

[18] S. Sarker, S. Sarker, A. Sahaym, and N. Björn-Andersen, "Exploring value cocreation in relationships between an ERP vendor and its partners: a revelatory case study," *MIS Quarterly*, vol. 36, no. 1, pp. 317–338, 2012.

[19] B. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chigago: Aldine, 1967.

[20] K. Charmaz, *Constructing Grounded Theory*. Sage, 2014.

[21] A. Strauss and J. Corbin, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA: SAGE Publications, 1990.

[22] C. Urquhart, H. Lehmann, and M. D. Myers, "Putting the 'theory' back into grounded theory: guidelines for grounded theory studies in information systems," *Information Systems Journal*, vol. 20, no. 4, pp. 357–381, 2010.

[23] A. Hanna, J. Windebank, S. Adams, J. Sowerby, S. Rance, and A. Cartlidge, *ITIL V3 foundation handbook*. The Stationary Office, Norwich, UK, 2008.

[24] A. M. Pettigrew, "Longitudinal Field Research on Change: Theory and Practice," *Organization Science*, vol. 1, no. 3, pp. 267–292, Aug. 1990.