

Know-How Transfer in Core-Banking System Implementation Projects: A Case Study

Klaus Haller, Marcel Heuberger

COMIT AG, Pflanzschulstr. 7, CH-8004 Zürich, Switzerland
{klaus.haller, marcel.heuberger}@comit.ch

Abstract. In the last years, many Swiss banks replaced their old core-banking systems with new ones. Implementing a core banking-system requires that the bank, the software vendor, and the implementation partner collaborate. Crucial for the collaboration and the project success is planned and unplanned know-how transfer. This case study examines the know-how transfer in a core-banking system implementation project for a retail bank with around 200 project members. We describe the sourcing network and different roles people have with respect to know-how transfer. We elaborate the methods used within the project for know-how transfer and present first results of a study looking on their effectiveness. Thus, this paper gives project managers a good insight into know-transfer within a project and points out how to stimulate them successfully.

Keywords: Standard Software, Core-Banking Systems Project Management, Project Organization, Know-How Transfer

1 Motivation

During the last years, many Swiss banks modernized their IT application landscape by implementing new core-banking systems [1,2]. Core-banking system implementation projects are a joint effort of the bank and an implementation partner. They customize the system to meet the bank's and its customers' needs. Whereas the first and most important (and widely investigated, see e.g. [3]) goal is to deliver the project on time and on budget, this goal alone is not sufficient. The bank staff is only able to work with the new system and the application management is only able to ensure the system's stability if know-how transfer takes place. Furthermore, there are also know-how transfer issues in the project itself. The implementation partner transfers customization know-how and process know-how gained in other projects. The bank contributes process know-how of their specific processes. In theory, the importance of know-how management and know-how transfer in today's economy is widely accepted [4]. Concrete efforts to improve the know-how transfer in a project or between projects like the one undertaken by Intel Solution Services [5] are seldom. One reason might be the inherent difficulty of measuring know-how transfer success and its benefits.

Nevertheless, COMIT decided to take a closer look on know-how management in general and know-how transfer issues specifically in its implementation methodology, LeanStream [6]. LeanStream was initially developed for and used in two of the largest implementation projects of the Avaloq core-banking system [7] with medium-sized Swiss retail banks. Nowadays, it is applied in more and more core-banking system implementation projects.

In this paper, we share our main findings with respect to know-how transfer gained in one of the projects. The project size with around 200 makes this study especially interesting, because the project size is far behind the limit where complexity explodes and becomes a major concern [3].

Section 2 introduces the technical challenges of core-banking implementation projects. Next, we presents the different organizations and groups forming the project context, i.e. all groups interacting with the project and thereby forming a sourcing network without directly belonging to the project (Section 3). We continue with questions like who requires which know-how, and who can provide it (Section 4). In Section 5, we take a closer look on the main know-how transfer patterns in the project, whereas Section 6 discusses the results of our study concentrating on the project members' experiences. We conclude our paper with a short summary and outlook (Section 7)

2 Technical Aspects of Core-banking Implementation Projects

To ease the understanding of what and why know-how transfer takes place in core-banking system implementation projects, this section elaborates the technical challenges. Core-banking systems are standard software products. Buying standard software (instead of developing new applications in-house) is appealing for many companies. It allows companies to concentrate on their core business. The core business of a bank is providing financial services to customers: it is not developing software. Next, companies with in-house software development require the right technical and project management skills in-house. Otherwise (or even then) there is a high risk that their projects fail or the costs are higher than expected. The staff aspect is especially challenging for smaller and medium banks. Finally, buying standard software gives a bank access to state-of-the-art business processes. The software vendor implements the software at many banks and thereby gains access to different processes. This know-how allows him to improve continuously the business processes implemented in the standard software by choosing the "best-of-breed" processes.

Buying standard software is the first step. The second step is to customize the software to the specific needs of a bank. Customization means adopting and configuring the software such that the software supports the bank's specific unique selling proposition optimally and such that it collaborates smoothly with the bank's existing application landscape.

The level of freedom for customization depends on the software product. Certain components might be static whereas others are adjustable. We base our discussion on our generic standard software model (*Figure 1*). Standard software comes with a data model. The data model provides object types like account types, accounts, bookings etc. The standard software might allow adding additional attributes to accounts (e.g. to support special discount models). Certain objects require a predefined set of object instantiations. An example is the object type accounts. The vendor might ship the software, e.g. with “savings accounts.” Banks typically have different names for savings accounts or special account types for certain customer types (e.g. “Midland Bank Student Saving Accounts”). We refer to such data as “domain values.”

Workflows implement the bank’s business processes. They depend on the bank’s business model. A retail bank requires a high level of automation. A wealth-management oriented bank relies on flexibility requiring manual interaction. A good example is also the loan approval workflow. The loan approval workflow of small banks has two main steps. Step one is the bank clerk who inputs the loan details into the system. Step two is the credit officer who approves the loan. Larger banks might also give loan with an amount of more than five or ten millions. Then, the loan approval workflow should be adopted, e.g. to a third person’s approval. Furthermore, standard software provides reports for internal or external purposes. The chief risk officer needs an overview of the bank’s credit portfolio whereas other reports are produced to meet regulatory demands. If a bank needs additional reports or wants to modify existing ones, reports become part of the customization efforts.

Finally, there are interfaces. Even a core-banking system does not provide all functionality needed to run a bank. Interfaces to other applications of the bank’s application landscape have to be customized, e.g. to trading systems or to external service providers like SWIFT [8].

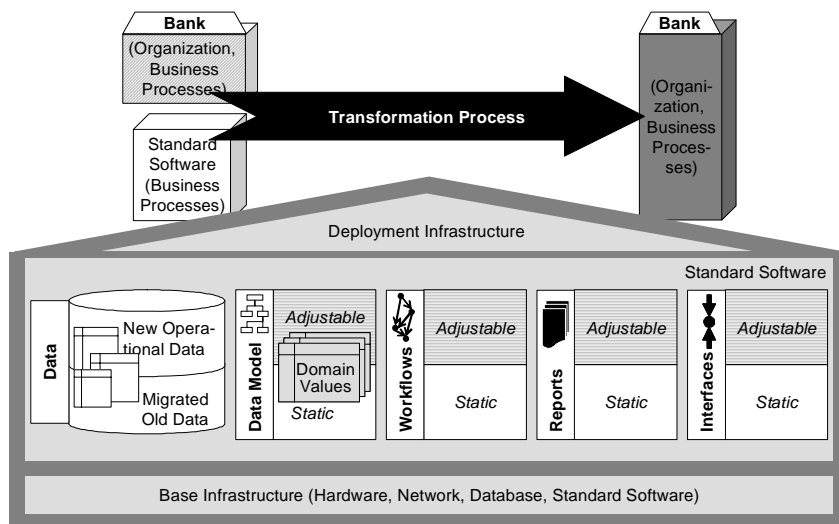


Figure 1: Generic Standard Software Model

Obviously, a core-banking system has many components. All components might be adopted during the customization. The success of the customization depends highly on the project team. The bank with its processes and organizations demands that the standard software is configured to its needs. But implementing standard software also requires the bank to adopt its organizational structure and business processes. Therefore, implementing standard software is a transformation process for the bank. One crucial success factor for this transformation is the know-how transfer in the project.

3 Project Context

The core of each core-banking implementation project is the joint implementation project team (short: joint team). It consists of consultants of an implementation partner and bank staff customizing the system together. The joint team requires certain know-how. It can get the know-how either by means of as-signing experts to the joint team or based on know-how transfer from persons outside the joint team to joint team project members. As a rule of thumb, the implementation partner provides consultants with in-depth customization knowledge (technically and business aspects). The bank sends persons with process know-how and with know-how regarding how the processes contribute to the bank's unique selling propositions.

Figure 2 compiles the different know-how groups and their interactions with the joint team in the middle. The **bank** has three main staff pools interacting with the joint project. The first pool consists of future *Avaloq users*. They need know-how how to execute (potentially adopted) business processes with the new system. There is the logistically challenging task of training each bank employee and ensuring he or she gets practical experience before the system is going live (e.g. by some training days and a few days during which the bank simulates complete working days). Users with in-depth knowledge of the processes, the business domain, and the old system are *key users*. They must ensure that the customized system can be used efficiently and provides mandatory functionality. Thus, as many key users as possible should join the project team. Secondly, the bank has an *Avaloq team*. Its size depends on the bank's sourcing strategy. In case the bank decides to outsource the application management to an IT service provider (ITSP) as in our case study, a small team for the management of the relationship to the ITSP might be sufficient. Thirdly, the bank has their own *help desk* for supporting users with basic problems (e.g. forgotten passwords) and forming the single point of contact for the users to the ITSP for severe problems. The help desk requires know-how to be able to support the users in the future. For example, the help desk must know how to change passwords or how to open new users in Avaloq.

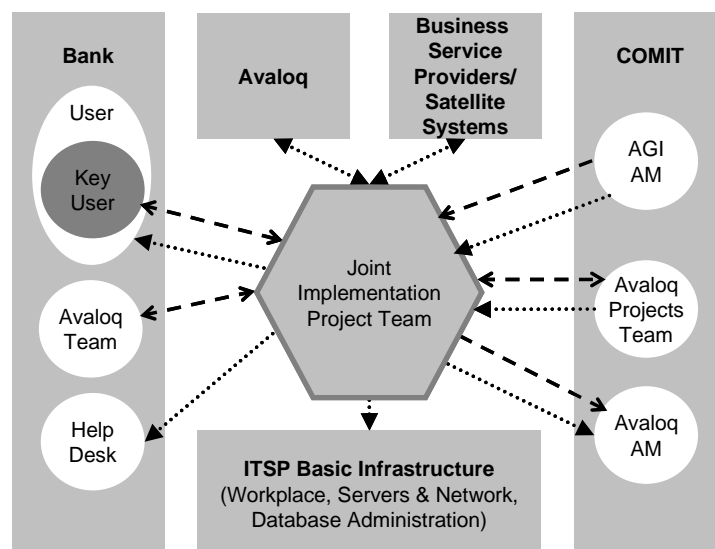


Figure 2: Project Team and Context and Know-How Transfer. Broken lines represent assignment-based know-how transfer. Dotted lines represent inter-personal know-how transfer.

The second contributor to the project pool is the **implementation partner**. The implementation partner also provides know-how for the project through assignments. Firstly, there are consultants of the Avaloq projects pool with customization know-how. They work in the project and, afterwards, move on to the next project. They also have access to other members of the Avaloq projects pool if additional expertise is needed. In this special case, COMIT has provided application management services for the old core-banking system AGI and was going to offer the same services for the future Avaloq system. Thus, COMIT could contribute persons from its AGI application management pool to the project. They undergo an intensive technical Avaloq training, work in the project, and then go to the new Avaloq application management team.

The third organization involved is the **IT service provider of the basic infrastructure**. It is responsible for the network, servers, database administration, workplace infrastructure etc. A smooth integration of their and the project's processes are important. They need at least know-how about basic processes like end-of-day processing and configuration management. Furthermore, they have to understand how Avaloq relies on the underlying database to configure it accordingly.

Also, there is knowledge exchange between the core-banking system vendor **Avaloq** and the project team. The obvious direction is from the software vendor to the project via system documentation and trainings. But there is also the way back via bug reports. In our case, the vendor implemented additional features for the bank. Thus, requirements and later details about the exact usage of the features flew from the project to the vendor. Finally, there is know-how exchange between the project and many **business service providers** like the Swiss stock exchange, Telekurs, Valorenzentrale etc.

4 Project Team Know-How

The LeanStream methodology organizes the joint team in different (sub-)teams, which can be grouped into two types [6], business teams and technical teams. Business teams customize one business domain like loans, trading, etc. Technical teams are responsible for cross-domain tasks. Examples are data migration [9,10] or the project's application management with responsibilities like configuration management or supervising the end-of-day processing. All teams have similar roles though they work on different topics. We differentiate the following ones (see Table 1 for details):

- *Team leaders (Bank)* have management experience and broad business knowledge.
- *Team leaders (Implementation Partner)* have management experience and a sound Avaloq know-how including at least a four-week vendor training.
- *Consultants (senior Avaloq profile)* have many years of customization experience and a track record in their application area. They know the possibilities and limits of the core-banking platform.
- *Consultants (Avaloq project pool)* have sound Avaloq and banking knowledge. In contrast to consultants with a senior Avaloq profile, they do not have a track record in the area they are working in *this* particular project.
- *Consultants (application management profile)* previously worked for the same bank in the application management of the old banking core-banking system. They contribute their knowledge about the customer, the way IT supported the customer's business and personal relationships with bank employees.

Function	Company	Management Experience	Intensive Technical Avaloq Training	Broad Avaloq Knowledge	Business Expert	Bank Specifics
<i>Team Leader</i>	Bank	+	?	?	+	+
<i>Team Leader</i>	COMIT	+	+	?	?	?
<i>Avaloq Experts</i>	Bank	?	+	-	+	+
<i>Avaloq Consultant</i>	COMIT	?	+	-	?	-
<i>Avaloq Consultant/ Senior Expertise</i>	COMIT	?	+	+	?	-
<i>Avaloq Consultant/ Senior AM Profile</i>	COMIT	?	+	-	+	+
<i>Key Users</i>	Bank	?	-	-	+	+

Table 1: Project Member Functions and Required Know-How. (“+” high level of experience, “-“ not necessary, “?” might have know-how)

- The bank's *Avaloq experts* usually worked for many years for the bank, i.e. they know the business and the staff very well. They participated in the intensive vendor training. In our specific situation, some already had previous Avaloq project experience.
- *Key Users* know the bank and their processes very well. They are the only pure business-staff in the project.

Certainly, it is not possible to assign only persons with a senior profile to the team. There are different reasons why this is not possible and even not desirable. Firstly, not all tasks are demanding enough to be acceptable for senior staff in a longer project. Secondly, many consultants are eager to learn new things. Assigning them in a team with a previously not well known domain is an efficient realization of job rotation. Thirdly, the focus of projects can shift. Retail-banking projects need more expertise with loans; private-banking projects have more sophisticated needs in asset-management problems. Finally, the implementation partner has to have the possibility to extend its staff pool and empower consultants for future project and application management work. Nevertheless, the project's success depends heavily on having enough senior experts. Thus, COMIT followed an *injection model* approach (Figure 3). Senior experts are strategically infused into the project to 'infect' the others with their know-how.

The overall team organization is as following: each team is lead by one person from the bank and one from the implementation partner. COMIT usually assigns at least two consultants with a senior profile plus consultants with an application management profile (they have to have knowledge of the customization to maintain it later) plus additional Avaloq consultants. The bank's assignment of Avaloq experts depends on the importance of the area for their business. Furthermore, the bank provides key users for all business teams.

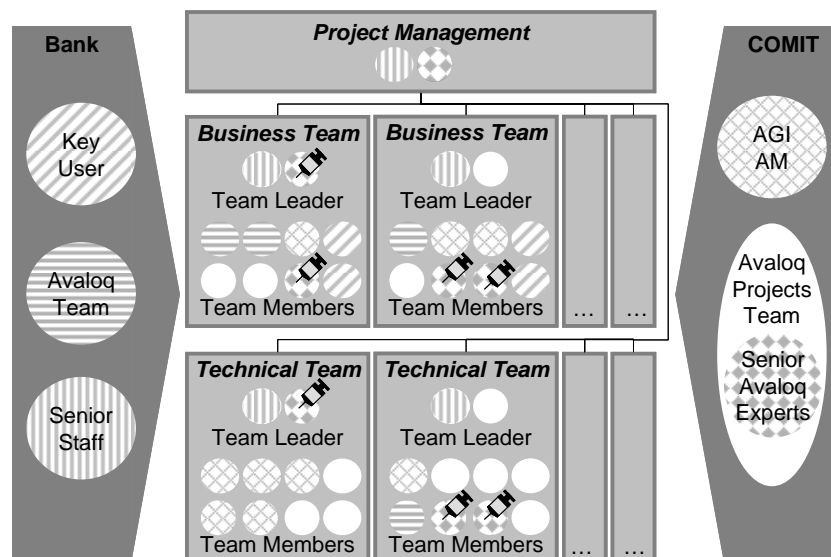


Figure 3: Injection Model containing a simplified project organizational chart and the bank's and COMIT's staff pools. Senior experts are marked with a needle.

5 Know-How Transfer Patterns

Know-how transfer can be organized in different ways, but the underlying process is always the same: tacit know-how has to be brought in an explicit form and transmitted to the receiver. The receiver has to understand and internalize the know-how [11]. The process can be looked at in two ways depending on whether knowledge is seen from an information logistics perspective or from the learning perspective [12]. In the latter case, you are interested in how stimulating and supportive the project environment is for individuals having to learn. The information logistics perspective focuses on how one can ensure that persons in need of know-how get the know-how from a person having the know-how. In our work, we concentrated on the latter aspect. Thereby, we observed the following patterns in the project:

- **Avaloq customization training.** The software vendor Avaloq offers intensive, four-week trainings focusing on the technical aspects of customization.
- **Key user training.** Key users get a one or two days' initial Avaloq user training by the implementation partner. The training continues within their business team, where they learn mostly in one-to-one trainings how to execute 'their' business processes with Avaloq.
- **Expert coaching.** No matter whether consultants or bank staff only got four weeks Avaloq training or they had already worked with Avaloq for several years, if they are new in a specific business domain, they need support from senior experts of the actual domain. The support might include specific domain know-how including details and pit falls of the customization of the business or technical domain.
- **Senior developers' meeting.** Team members with senior expertise of each team meet once per week to discuss technical problems and possible solutions. Thereby, know-how transfer takes place.
- **Inter-team coaching.** In principal, each team customizes an application domain independently. In reality, there are always interdependencies. The interdependencies require know-how of a different business domain or at least about the interface. Good examples are end-of-period processing like end-of-day, end-of-month, end-of-year etc. The project's application management team starts and oversees the processing which includes aspects of nearly all topics like calculation of interests for loans, saving accounts, or the evaluation of the performance of wealth management mandates. Overseeing the process and contacting the business teams only in case of major mistakes requires that the application management team has basic know-how of all domains, meaning that it must be transferred by inter-team coaching to them.
- **Delta specification.** The idea of a delta specification is to analyze the gap between the bank's expectations and the standard implementation of the core-banking system. The identified gap named *delta* is each business team's road map for the customization. A senior Avaloq expert and a key user together execute each process of their business domain in Avaloq to identify which adoptions the bank needs. Though know-how transfer is not the primary aim of the delta specification process, it is an obvious side effect.

- **Project intranet.** Each team has a project team page, which can be found via the global project homepage by the team's members or by other teams. The teams decide independently which information they put on their homepage to share with others.

6 Study and Lessons Learned: Success Factors for Knowledge Transfer

To improve work in future projects, companies often rely solely on the project managers' impressions or on lessons learned formulated spontaneously by some project members. In contrast, as part of COMIT's LeanStream methodology improvement, we took a broader perspective and interviewed nine project members with diverse backgrounds: two team leaders, four members of business teams, two of them with senior Avaloq profile, and three members of technical teams. The interviews were semi-structured. Semi-structured interviews are suitable for areas with no or only limited work [13] as in our particular case.

All interviews followed the same pattern (*Figure 4*). After an explanation of the purpose and the structure of the interview, the interview partners explained their observed know-how needs. Then, they could talk about how they got the know-how within their team or from other teams. Finally, the interview partners freely expressed their experiences regarding know-how transfer in the project.

In the rest of this section, we concentrate on four main issues we identified in the interviews. We explain the general situation, the results from the interviews, and how the results correlate with previous work from literature. Finally, we formulate for each issue our "lessons learned" in an operational way, i.e. what one should consider when managing a similar project.

6.1 Face-to-Face Meetings

The project organization implied much know-how transfer in face-to-face meetings. Examples are key user training for their special business area, delta-specification, and the senior developers' meeting. Face-to-face meetings are known to be highly efficient as known from the following prominent statement [11]: "How can an organization transfer knowledge effectively? The short answer, and the best one, is

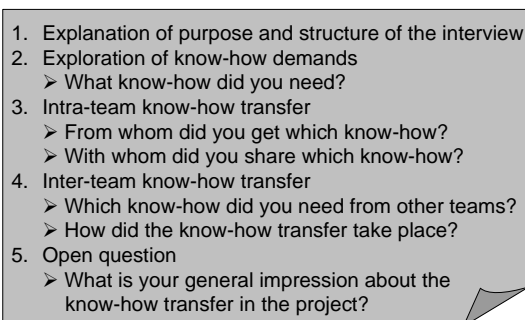
- 
- A diagram showing the structure of an interview. It is a rectangular box with a folded bottom-right corner, containing a numbered list of five items. Each item has a sub-point indicated by a right-pointing arrowhead (➤).
1. Explanation of purpose and structure of the interview
 2. Exploration of know-how demands
 - What know-how did you need?
 3. Intra-team know-how transfer
 - From whom did you get which know-how?
 - With whom did you share which know-how?
 4. Inter-team know-how transfer
 - Which know-how did you need from other teams?
 - How did the know-how transfer take place?
 5. Open question
 - What is your general impression about the know-how transfer in the project?

Figure 4: Interview Structure

hire smart people and let them talk to one another. Unfortunately, the second part of this advice is more difficult to put into practice.” In line with literature, our interviews proved the efficiency of face-to-face meetings. Consequently, the project organization should also consider many face-to-face meetings in the future and ensure this by the close collaboration of bank employees and consultants.

6.2 Distributed Locations

Most teams worked four days a week in two bank buildings within two minutes walking distance. Only very few teams were located in a third building which is a 20 minute bus-ride away. There were many possibilities for building personal networks like informal after-work gatherings for the whole project, a small cafeteria, or the fact that many consultants took the same trains or had dinner together in the hotel. Nevertheless, nearly all interview partners complained about the collaboration with *other* teams, e.g. because of the distance to them or because it was difficult to find people from other teams with needed knowledge. One explanation might be that the impression is misleading that there is “one project in one place.” The project consists of many normal, non-virtual teams, but together they form a virtual project team due to the subjectively perceived distances to other teams.¹

The consequence for future projects is that project managers should continue locating all project teams as close to each other as possible and invest in people with good communication skills. But project managers should take into account that project members might not see this effort in large projects. They might feel that distances hinder needed know-how sharing. Therefore, larger projects should consider collaborative work tools for virtual teams even if the project looks like it is being run by one team in one location.

6.3 Information Sharing Attitude

The project expected experienced consultants not only to do their usual tasks but to train (i.e. to share knowledge with) an unusually large amount of newly hired consultants. The latter ones should reach the *same* level of experience.

The interview results about the training-on-the-job are interesting and in part contradicting. We got statements like sharing knowledge is not acknowledged or that “we old ones had to learn it by ourselves, too.” These statements seem to come from experienced persons. In contrast, many others praised the willingness for information sharing and the general helpfulness. Our interpretation is that the expressed willingness for information sharing was lower than the actual one. We see two explanations. Firstly, the project pressure with its tight deadlines required to share knowledge though some persons who might not have been too pleased with that. Secondly, the experienced consultants might have shared their knowledge happily with their colleagues but got the impression that their extra efforts were not adequately recognized.

¹ See e.g. [16] for more details on virtual teams.

We learned from this analysis that one has to think carefully about motivating experienced consultants and how to recognize extra training-on-the-job efforts. Certainly, such aspects are really difficult to handle (see [14] for a broader discussion of possible approaches).

6.4 Interdependencies between teams

Interdependencies between teams require communication. Communication takes place if communication channels like meetings exist. If communication channels are missing, quality problems arise and might not be noticed until a late project stage. Sosa et al. describe this phenomenon in detail and present an example from aircraft engineering [15]. Our interviews proved the project set-up to be very good. Most needed communication channels were planned and only a few were missing. We want to give a short example. The bank account opening and the cost and fee calculation are closely linked but fell into the responsibility of different teams. Both overwrote each other's work a few times until they noticed they had to talk with each other. The key for solving the problem was, like in the other situations, that someone took the initiative and responsibility for the coordination. In contrast to the experiences Sosa describes for complex aircraft engineering, self-organization still works for our project with around 200 persons. But certainly, this requires consultants and bank employees with high self-motivation and commitment to the project's success.

We see two possibilities for improving future projects. Firstly, it is not sufficient that the project set-up guarantees information flow about planning and organizational issues simply by project management meetings. Inter-team exchange of technical issues has to be organized, too. For example, the participants of regular senior developers' meetings praised them in interviews. Also "ordinary" team members of different teams should know each other. Thus, they can detect missing communication channels earlier and set them up more quickly. Secondly, an implementation partner should have one list as a corporate memory to which all projects contribute "problematic" situations/subjects where communication channels have been missing.²

7 Summary and Outlook

Many companies contribute their knowledge to make a core-banking implementation project a success. The three main contributors are the bank itself, the implementation partner, and the software vendor. A concrete implementation project

² One might be surprised that we do not suggest preventing such problems completely. This is not realistic for two reasons even though COMIT (as the biggest Avaloq implementation partner) realizes many projects. Firstly, core-banking systems continuously add new functionality e.g. due to regulatory requirements. Secondly, banks want to be involved in the definition of a project's organizational structure. The bank influences which business teams are set-up and which tasks are assigned to them. Different organizational structures and new functionalities can both imply that suddenly a new dependency between two business teams appears in a new project the first time.

requires banking know-how, IT know-how, and the knowledge how to use IT as an enabler of the business processes. Consequently, bank employees and consultants have to work together as one large team. The project success depends heavily on the success of their know-how exchange. We identified the following five main topics in the project:

- (1) sufficient face-to-face meetings
- (2) adequate collaboration tools for larger projects even if the team seems to be working “at one physical location only”
- (3) adequate acknowledgement of efforts for training less experienced colleagues
- (4) maintaining a list on the company level for collecting possible dependencies between teams
- (5) awareness for the necessity of know-how exchange between different teams on a technical level

We see these topics as catalysts for future, sustainable project management process improvements ensuring project out-performance and the ultimate goals of customers’ and employees’ satisfaction.

The authors are grateful that all their interview partners took time to answer the questions though they were involved in a project with demanding deadlines. Furthermore, we want to thank Marcel Ottiger for its valuable feedback to this paper and for the trustful working atmosphere he created for his project, which was a prerequisite for being able to perform the study.

8 References

1. C. Gabriel: Plattform-Wechsel: Parforce-Übung mit weitreichenden Folgen, in: Schweizer Bank (6/2007), Zürich, 2007
2. M. Minetti: Ein aktuelles Bild der Schweizer Banken IT, www.inside-it.ch, 6.12.2007
3. Ch. Sauer, A. Gemino, B. H. Reich: The impact of Size and Volatility on IT project performance, Communications of the ACM, Volume 50, No. 11, November 2007
4. Th. A. Stewart: Intellectual Capital – The New Wealth of Organizations, Bantam Doubleday Dell Publishing Group, 1997
5. K. Pugh, N. Dixon: Don’t just capture knowledge – Put it to work. Harvard Business Review, May 2008
6. LeanStream®, Version 3.0, Comit AG, Zürich, 2007
7. Avaloq Banking System, <http://www.avaloq.com/>
8. Society for Worldwide Interbank Financial Telecommunication, <http://www.swift.com/>
9. K. Haller: Data Migration Project Management and Standard Software – Experiences in Avaloq Implementation Projects, DW 2008 Conference, Lecture Notes in Informatics, St. Gallen, Switzerland, 2008
10. K. Haller: Towards the Industrialization of Data Migration: Concepts and Patterns for Standard Software Implementation Projects, 21st International Conference on Advanced Information Systems (CAiSE), 2009, Amsterdam, The Netherlands
11. Th. H. Davenport, L. Prusak: Working knowledge: how organisations manage what they know, Harvard Business School Press, Boston, MA, 1998
12. Ralph Trittman: Wirtschaftlichkeit von Wissenstransfers in der Softwareentwicklung – Ein kostenorientiertes Gestaltungskonzept, Shaker-Verlag, Aachen 2004

13. S. Albers, D. Klapper, U. Konradt, A. Walter, J. Worf: Methodik der empirischen Forschung, 2. Auflage, Gabler Verlag, Wiesbaden, 2007
14. K. Bartol, A. Srivastava: Encouraging Knowledge Sharing: The Role of Organizational Reward Systems, Journal of Leadership and Organization Studies, Vol. 9, No. 1, 2002
15. M. Sosa, St. Eppinger und C. Rowles: Are Your Engineers Talking to One Another When They Should, Harvard Business Review, November 2007
16. G. Hertel, U. Konradt: Telekooperaton und virtuelle Teamarbeit, Oldenbourg Wissenschaftsverlag, München, 2007