

Success Factors for Software Experience Bases: What We Need to Learn from Other Disciplines

Reidar Conradi*, Mikael Lindvall**, Carolyn Seaman***

* p.t. University of Maryland at College Park, conradi@idi.ntnu.no

** Fraunhofer Center for Experimental Software Engineering Maryland, College Park, MD, mlindvall@fraunhofer.org

*** Fraunhofer Center for Experimental Software Engineering Maryland, College Park, MD, cseaman@fraunhofer.org

Abstract

The interest for Software Experience Bases is rapidly growing as software companies recognize the need among software developers to share experiences with each other. This need is especially strong within large companies that are geographically spread out. However, there are several interesting questions connected to the implementation of software experience bases. Questions are for example: how do we best *externalize* experience (i.e. capture, package, structure, and classify experience)? When the experience is externalized the question is how to best enable *internalization* of the experience (i.e. how do we enable people to use and learn from the experience)?

The paper discusses success factors for *software experience bases* for *organizational learning* and what we need to learn from other disciplines in order to better understand how to deal with these questions. A survey of some previous efforts in the area is presented. Based on the survey, we formulate four *success factors* for software experience bases. These are: 1) a successful implementation of a software experience base requires cultural changes, 2) the business environment should be relatively stable, 3) a software experience base should offer concrete business value, and 4) the implementation and introduction of a software experience base should be done in an incremental fashion.

Keywords: Software Experience Bases, Software Process Improvement, Learning Organizations, Knowledge Dissemination, Success Factors.

1. Introduction

Knowledge management by *experience bases* is gradually coming into use. This applies to banking, oil production and shipbuilding, as well as to software engineering. The goal with these efforts is to create and sustain a *learning organization* that constantly improves its work by letting employees share experience with each other. The ultimate success criterion is satisfied customers in the spirit of TQM [9] or ISO-9000.

So, how can software organizations best systematize, organize and exploit previous experiences in order to improve their work?

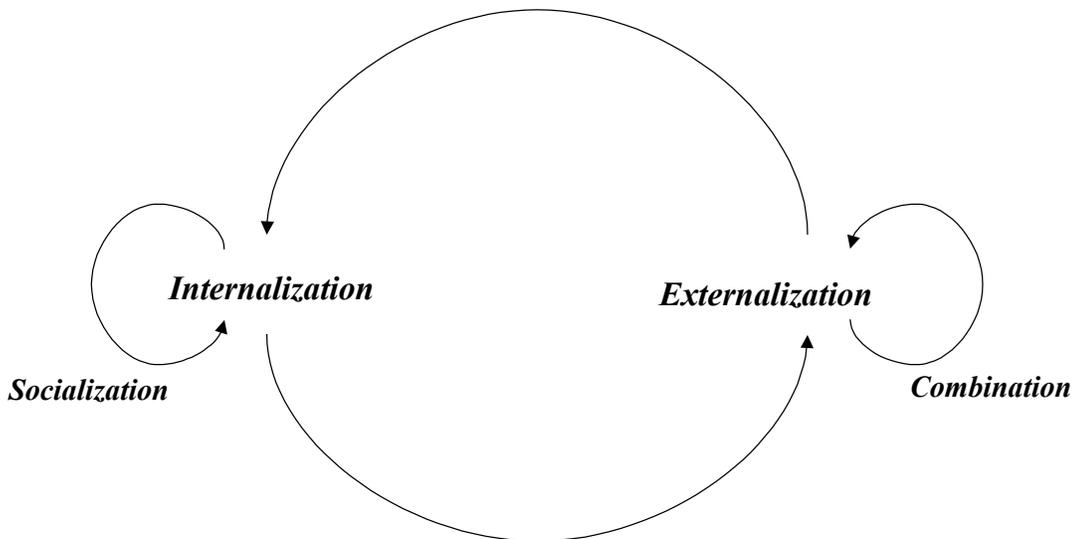
To enable learning is crucial, both at the individual and organizational level. It is, however, a challenge to make “externalized” information (stored guidelines, process models, quality estimators etc.) “internalized” as active knowledge and practical skills, i.e. to get it into real use by software developers.

We will use the term experience base, not experience database, to avoid too close of an association with traditional databases. An experience base may be in the brains of people, on paper, on the web, in files, in

spreadsheets, as well as in more classical Database Management Systems (DBMSs). However, computerized software experience bases (SEB) for software process improvement (SPI), will be our focus.

Some definitions: *Explicit* knowledge is formalized knowledge, e.g. as process models or guidelines. *Tacit* knowledge is the operational skills that practitioners possess, including practical judgement capabilities (e.g. intuition) [16]. Tacit knowledge can not always be made explicit. Learning requires *both* formal training and informal information exchange. Many theories of learning exist, see e.g. [18]. Most theories operate with a learning process (cycle) as in Figure 1 below.

Figure 1: Internalization vs. externalization in learning.



Practitioners first internalize new knowledge (i.e. individual learning). The new knowledge is then socialized into revised work processes and changed behavior (i.e. group learning).

The work processes and the changed behavior are observed, thus are they externalized. They are then combined to refine and extend the existing knowledge or experience base (organizational learning). This process continues in new cycles.

2. Success Factors for Software Experience Bases

As mentioned, there is a growing interest in using computerized software experience bases. For instance, there was recently a workshop on Learning Software Organizations [5]. In part, this growing interest reflects the advances of modern information technology, which enables efficient and cheap storage of and access to information (knowledge) at fingertip range.

Existing work on SEBs spans from the one used in the NASA-SEL Experience Factory for more than 20 years [3], to more recent efforts, for example the experience base at DaimlerChrysler [11].

Large IT companies often maintain proprietary experience bases, mainly storing summary information on previous projects. Yet other SEBs are still at the research level [10].

To succeed with the implementation of an SEB is not trivial. There are examples of SEBs that have been abandoned, even if initial acceptance from software developers was good [12]. In the Norwegian SPIQ project for Small and Medium-sized Enterprises (SMEs) in 1997-99, only two out of four industrial experience bases can be said to be successful. They are both on a low ambition level, meaning that they are based on a modest set of requirements [6].

In all this work, the central paradigm is to regard experiences, including the software itself, as assets. Further investments can be done in these assets for later capitalization. This involves spending effort and time in the present for an uncertain future return.

Based on work in the SPIQ project, from an in-depth experience domain analysis of the NASA SEL [4], and from the literature, we propose four main success factors for SEBs. These factors, later operationalized into criteria, should be used to evaluate the success of existing SEB implementations and to predict the level of success of planned SEB implementations.

F1. Cultural change: A successful implementation of an SEB requires cultural change. It is for example important that people provide knowledge to the experience base and use knowledge that is available. The SEB should also be well-connected to a SPI program and ingrained into the organizational culture towards learning is more like to succeed, than is a SEB that is perceived as an isolated entity.

F2. Stability: A stable business environment with the ability to change in a controlled way is more likely to successfully assume and make use of a SEB implementation, than is an unstable one.

F3. Business value: A SEB that brings a high amount of concrete, obvious, and demonstrable business value is more likely to be perceived as successful, than is a SEB that brings less business value, or whose value is harder to demonstrate.

F4. Incremental implementation: A SEB that is implemented in small increments in close connection with its future users is more likely to be a success than is a SEB that is developed in isolation without continuous feedback.

Similar success factors may be formulated for most improvement efforts, and illustrate that a SEB is not a freestanding entity, as F1 and F2 indicate. Criterion F3 is particularly difficult to assess, as it is usually very hard to quantify the business value of a SEB. Benefits are often intangible and the value chain -- linking the SEB to traditional success criteria of software projects such as time-to-market, cost and quality -- is very long.

Some further research questions concerning the contents and use of a SEB could be:

- What kind of experience is most useful to developers and should be stored in a SEB?
- How should experience be structured, classified, and packaged to be as useful as possible?
- Can incentives be used to create a self-sufficient and self-organizing SEB so that developers both contribute and use experience without the intervention by others?
- In what kind of environments are SEBs most successful and why?

3. Using Interdisciplinary Approaches to Improve the Usability of Software Experience Bases

Based on lessons learned from previous experience with implementing SEBs, we should lower the risks and increase the short-term benefits by adopting an incremental approach. That is, we should start implementing a simple and inexpensive SEB that brings high business value. At the same time, we should solicit support from both the lower and upper echelons of the organization. Below, we will discuss how competencies from multiple disciplines can contribute to the success factors F1-F4:

Behavioral sciences (human factors)

For F1 on cultural change, we need to look at organizational theory and social anthropology, in particular studies of organizational culture, for how to observe, understand and assess actual work patterns and how to best integrate new technology into existing organizations. Further, we need to study action research [13] where the researcher or champion becomes a driving and integral part of the actual organization, which also happens to be the study object. We also need to look at qualitative assessment approaches [15], [19].

For factor F2 on stability: this is ultimately decided by the market place and general technology development, but we believe that proper management techniques [8] can reduce the impact of instability in terms of personnel, organizational structure, and business domain.

For factor F3 on business value, using questionnaires and qualitative indicators (possibly from the education or MIS fields) can be a practical way to assess the cost/benefits, and to assess present and future expectations and needs.

Software process improvement and quality management

For all success factors F1-F3, we should have a close coupling with improvement frameworks, such as TQM, ISO-9000, or the Quality Improvement Paradigm (QIP) [3]. The latter assumes that a SEB exists in an internal Experience Factory to drive organizational learning. However, if we are to succeed with SEBs to achieve learning, we must not carry the traditional “QA hat” of control.

Classic measurement methods and analysis, cost benefit/models

For F3 on business value: In addition to above qualitative indicators and methods, we need traditional software metrics techniques (such as the Goal-Question-Metric method [2]) to assess the costs and benefits of a SEB. An approximate overhead figure of 1-2% is recommended for the level of effort expended on measurement programs. In addition, we should welcome economists to come up with a quantitative investment model for reuse items. For instance, new buildings or PCs in a business easily escape the same spotlight for financial return as improvement efforts, but how to amortize reuse investments in a sound way?

Technology transfer and introduction

For F1 on cultural change, F3 on business value, F4 on incremental development: A SEB is both a technical tool and an enabler for learning. For both functions, users must be trained and convinced of its business value. For instance, we should actively demonstrate the usefulness of a SEB tool to the users, and make it “dynamic”, e.g. show that there are new experience items to investigate. We should implement the SEB in small increments and exchange feedback with the users often. Maybe a reward system for insertion and reuse of experiences will work [17]? We believe that technology transfer skills can contribute in all this. Management science and organizational behavior may be the fields in which to look for help in this area.

Software tools (Databases, Knowledge management, Technologies)

An experience base is not a technical gadget, but a vehicle for organizational learning and process improvement. The insight from software reuse is that software repositories are not considered crucial [17]. This means that relevant SEB tools must be chosen with great care. The Web seems an excellent vehicle to store and disseminate information, while keeping start-up costs low. Perhaps also Case-Based Reasoning Techniques [1] can play a role. The danger with SEBs and similar repositories is information overload and how to keep the stored information updated and relevant. To avoid ending up with “data cemeteries”, we should try to make the SEB an active communication channel. Computer Supported Cooperative Work (CSCW) technologies are

highly relevant and should be investigated. Coupling and integration of the SEB to the daily production tools are also essential.

4. Conclusion

Incremental approaches to establishing and maintaining a SEB are favored, e.g. using the web to lower start-up costs. Important problems are internalization of the SEB into the workplace practice and externalization of models in a repository tool. A pervasive, organizational commitment is ultimately needed, although we may start in the small. The main obstacles to a successful SEB are therefore economic and cultural, not technological -- as we can see in other reuse, improvement or change efforts. The composition of the SEB team should reflect this insight. Other disciplines with decades of experience in economic and cultural issues could be consulted in order to increase the chances to success with an SEB effort.

Acknowledgements: Thanks to colleagues at the Fraunhofer Center for Experimental Software Engineering Maryland for comments.

5. References

- [1] Klaus-Dieter Althoff, Andreas Birk, Susanne Hartkopf, Wolfgang Müller, Markus Nick, Dagmar Surmann, and Carsten Tautz: "Managing Software Engineering Experience for Comprehensive Reuse", In Proc. 11th Conf. on Software Engineering and Knowledge Engineering (SEKE'99), 16-19 June 1999, Kaiserslautern, p. 10-19, Knowledge Systems Institute, Skokie, IL, USA, June 1999.
- [2] Victor R. Basili, Gianluigi Caldiera, and Hans-Dieter Rombach: "The Goal Question Metric Paradigm", In [Marciniak94], p. 528-532, 1994.
- [3] Victor R. Basili, Gianluigi Caldiera, and Hans-Dieter Rombach: "The Experience Factory", In [Marsiniak94], p. 469-476, 1994.
- [4] Victor R. Basili, Carolyn Seaman, Roseanne Tesoriero, and Marvin V. Zelkowitz. "OSMA Software Program: Domain Analysis Guidebook." University of Maryland Institute of Advanced Computer Studies and Department of Computer Science, University of Maryland Technical Report CS-TR-4003, December 1998.
- [5] Frank Bomarius (ed): Proc. Workshop on Learning Software Organizations (associated to SEKE'99), Fraunhofer IESE, Kaiserslautern, 16 June 1999, 126 p.
- [6] Reidar Conradi: "SPIQ: A Revised Agenda for Software Process Support", In Carlo Montangero (Ed.): Proc. 4th European Workshop on Software Process Technology (EWSPT'96), p. 36-41, Nancy, France, 9-11 Oct. 1996. Springer Verlag LNCS 1149.
- [7] Reidar Conradi and Torgeir Dingsøy: "Software Experience Bases: A Consolidated Evaluation and Status Report", Accepted for PROFES'2000, 20-22 June 2000, Oulu, 14 p., NTNU/Univ.Maryland.
- [8] Tom DeMarco and Tom Lister: "Peopleware: Productive People and Teams", Dorseth House, 1987, ISBN: 0-932633-05-6.
- [9] W. Edwards Deming: "Out of the crisis", MIT Center for Advanced Engineering Study, MIT Press, Cambridge, MA, 1986.
- [10] Raimund L. Feldmann: "Developing a Tailored Reuse Repository Structure – Experience and First Results", in [Bomarius99], p. 45-58, 16 June 1999, Kaiserslautern.

- [11] Frank Houdek, Kurt Schneider, and E. Wieser: "Establishing Experience Factories at Daimler Benz: An Experience Report", Proc. 20th Int'l Conf. on Software Engineering, Kyoto, May 1998, p. 443-447.
- [12] Arne Koennecker, Ross Jeffery, and Graham Low: "Lessons Learned from a Failure of an Experience Base Initiative Using a Bottom-up Development Paradigm", Proc. 24th Annual NASA-SEL Software Engineering Workshop (on CD-ROM), NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA, 1-2 Dec. 1999, 10 p.
- [13] Morten Levin and Davydd J. Greenwood: "Introduction to Action Research – Social Research for Social Change", Sage, 1998.
- [14] John J. Marciniak, editor: "Encyclopedia of Software Engineering – 2 Volume Set", John Wiley and Sons, 1994.
- [15] Matthew B. Miles and A. Michael Huberman: "Qualitative Data Analysis: An Expanded Sourcebook", 2nd Edition. Sage Publications, 1994.
- [16] Ikujiro Nonaka and Hirotaka Takeuchi: "The Knowledge-Creating Company", Oxford University Press, 1995.
- [17] J. Poulin: "Populating Software Repositories: Incentives and Domain-Specific Software", Journal of Systems and Software, Vol. 30(3), Sept. 1998, pp. 187-1999.
- [18] Peter M. Senge: "The Fifth Discipline: The Art and Practice of the Learning Organization", Currency/Doubleday, 1995.
- [19] Steven J. Taylor and Robert Bogdan: "Introduction to Qualitative Research Methods", 2nd Edition. John Wiley & Sons, 1984.

BIOs:

Dr. Reidar Conradi is a professor at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He received his MS in 1970 and his Ph.D. in 1976, both in computer science from NTNU in 1970 and 1976, and has been at NTNU since 1985. Dr. Conradi is currently a visiting scientist at Fraunhofer Center for Experimental Software Engineering Maryland. His interests are process modeling, software process improvement, software engineering databases, versioning, object-orientation, software architectures for distributed systems, and programming languages.

Dr. Carolyn Seaman is currently an assistant professor in the Information Systems Department at the University of Maryland Baltimore County. She is also a Research Scientist at the Fraunhofer Center - Maryland. She received her Ph.D. in 1996 from the Computer Science Department at the University of Maryland in College Park, her MS in 1987 from Georgia Tech, and her BA in 1986 from the College of Wooster (Ohio). Her research interests include software maintenance, organizational and communication issues in software development, development with COTS, software engineering experience reuse, and empirical studies of software engineering.

Dr. Mikael Lindvall is currently a scientist at Fraunhofer Center for Experimental Software Engineering Maryland. He received his Ph.D. in Computer Science from Linköping University, Linköping, Sweden. His interests are Empirical Studies, Experience Management, Software Process Improvement, Object-Oriented, Impact Analysis, Requirements Engineering, and Legacy Systems.