

# From Software Experience Databases to Learning Organizations

Reidar Conradi\*

Norwegian University of Science and Technology (NTNU), Trondheim, Norway

## 1 Introduction

Knowledge management by *experience databases (EX-DBs)* is gradually getting into use. This applies e.g. for banking, oil production and ship building, as well as for software engineering. The goal is to create and sustain a *learning organization*, where the bottom-line criterion is satisfied customers in the spirit of TQM or ISO-9000.

## 2 Some useful definitions

**Knowledge is use of facts, truths or principles from studies or investigations.** That is, the available information must be made operational for (“learned” by) the person or group in question. Thus, information is not automatically knowledge.

*Explicit* knowledge is what can be formalized, e.g. as process models or guidelines in a quality system. *Tacit* knowledge is the operational skills among practitioners, including practical judgement capabilities (e.g. intuition).

We can also distinguish between *ease of transfer* of local vs. global knowledge, and between *ease of use* of programmable (often explicit) vs. unique (often tacit) knowledge:

Type	Local	Global
<b>Programmable</b>	<i>Easy to transfer, fits only certain cases.</i> Ex. Bill prefers Ada.	<i>Easy to transfer, fits in many cases.</i> Ex. Filling in time-sheets.
<b>Unique</b>	<i>Difficult to transfer, fits only certain cases.</i> Ex. Unix maintenance.	<i>Difficult to transfer, fits in many cases.</i> Ex. Running big projects.

In organizations we can also have **knowledge transfer** between one or many senders and one or many receivers:

\*Dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway. Phone: +47 73.593444, Fax: +47 73.594466, Email: conradi@idi.ntnu.no

	Sender	
Receiver	One	Many
One	* Talks	* Social Networks
Many	* Presentations * Books * Articles	* Experience databases

As seen, there is many types of knowledge and of knowledge transfer. Which of these facilitate or represent learning? We can at least say, that learning requires both formal training and informal information exchange.

Many theories of learning exists, see e.g. Senge [6]. Most theories operate with a learning process (cycle) as in figure 1. Here, common knowledge is first being internalized to change local behavior (“learning”). Then the modified behavior is externalized to refine the existing knowledge – etc. in a new cycle.

Much R&D effort has been being spent in the “upward, externalizing” direction, looking for valid experiences that can be analyzed, generalized and synthesized in the form of improved models or concepts.

However, the hard part is the “downward, internalizing” flow. That is, how to make an impact on current practice, in spite of that more updated knowledge may be convincingly available. Typical inhibitors are “not-invented-here”, mistrust to formal procedures (as represented in a conventional quality system), lack of time/resources, and general inertia.

We will define **organizational learning as the ability of groups to improve their results based on new knowledge.** So, how to support this?

We believe that internal “knowledge” organizations like a local *Experience Factory (EF)* [1] can structure and facilitate knowledge transfer at the Project-Organizational level. Such an EF should not assume the role of a conventional, perhaps “over-controlling” quality manager. It should rather encourage unselfish recording, synthesis and sharing of experiences.

Likewise, the *Goal-Question-Metrics (GQM)* [2] method can facilitate knowledge management at the Individual-Person level, e.g. by defining lean and relevant metrics and by insisting on regular feedback sessions.

### 3 Some examples of experience databases

In a national, Norwegian Software Process Improvement effort, SPIQ (SPI for Better Quality [4]), six companies are now designing, implementing and deploying simple EXDBs. Three of these are:

- **Company X1**, consultancy house: 140 developers mostly with MSc/PhD degrees.

X1 specializes in OO, UI, and AI technologies, and uses DSDM method for incremental development (www.dsdm.org). It has a flat and “process-oriented” organization.

X1 has developed a web-based corporate memory, storing administrative information, personnel competence profiles, overall project routines (not a full quality system), and day-to-day news and events. Very few hard data are collected and stored, except major project data.

For its limited ambition, it functions well.

- **Company X2**, telecom software house: 600 developers, owned by a large national telecom provider, now undergoing major transitions and mergers.

X2’s main profile in administrative support systems for telecom – logistics, personnel, billing, but not switching. It has developed and operates a dozen large information systems, e.g. developed by ORACLE-2000 Designer. Major organization instability.

X2 has introduced a web-based quality system (an external, “canned” process) and an advanced estimation system using data from 50 previous projects. The quality system was mostly introduced “over the head” of people, and e.g. final project reports are hardly ever picked up and used later – a rather demotivating fact.

The estimation system is aimed at project managers which have been given a one-day course in this, but this has not taken off either. However, the majority of project managers are positive to start using the system (an internal poll shows this), so a more person-to-person coaching approach will be attempted.

All in all, much synthesised knowledge has been collected and made easily available to key persons, but actual reuse of this information has been meager. However, all improvement efforts in X2 have been hampered by major reorganizations the last year.

- **Company X3**, consultancy house: 500 developers, Norway’s next largest with five branch offices.

X3 has developed an web-based **Information Well** using Microsoft Exchange. This knowledge base stores general company and personnel information, as well as guidelines for and experiences from using company

methods and tools (SELECT, UML, Process Engineer etc.) in different domains (banking, public sector, telecom etc.). Each method or domain subarea has a responsible person to maintain and quality-check the information. A better estimation capability is being built up, in cooperation with NTNU.

X3 claims that this Information Well is increasingly being used by managers and consultants. A technical drawback is that the stored documents exist in different document formats and versions of these (Powerpoint, Word etc.), and that the tools installed by each employee have incompatible versions in relation to the documents.

X3 is dedicated to improve the Information Well, but has experienced major internal reorganizations the last year.

### 4 Pre-conditions for Organizational Learning by Experience Databases

Some general observations:

- A general precursor for all organizational change and SPI is commitment and consensus, all the way from top-level management to the rank-and-file. This requires, perhaps, a rather flat organization and a democratic culture?

- Another precursor is sufficient organizational stability, so that appropriate improvement initiatives can be sustained for a sufficient time period, say 4-5 years.

In SPIQ, staff turnover and other changes in the involved companies has been serious impediment in both smaller and larger companies.

- A long-term goal is an “egoless” approach to sharing experiences, both good and bad.

However, some high-risk projects that test out new technologies should not be over-announced before we can draw valid conclusions. That is, an element of *privacy* should be respected. This also applies to the personal and organizational level.

- The web is an excellent vehicle to store and disseminate information. The danger is information overload and how to keep the stored information lean, updated and relevant.

The syndrome of “data cemeteries” is also a problem, cf. the fate of project reports in company X2. However, more sophisticated search methods, such as data mining, might help.

- To generalize, we need to compare present and previous status, i.e. to conduct valid empirical studies [3]. However, software experimentation is not easy and baselines are often fuzzy.
- An experience base is *not* a technical gadget, but a vehicle for organizational learning and process improvement.

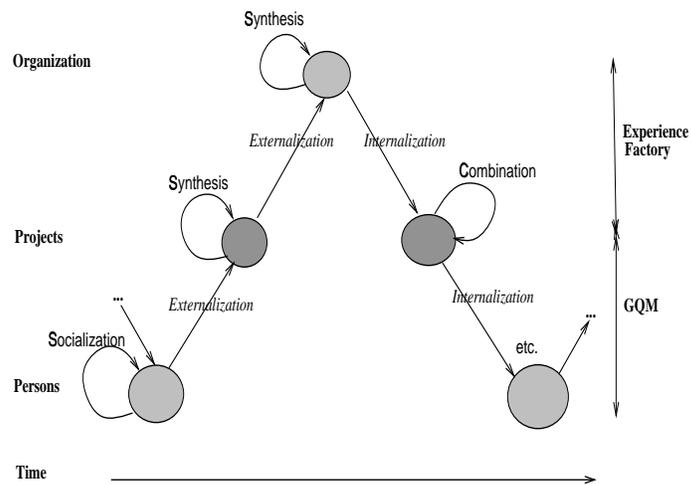
That means that organizational, not technical factors may be decisive. And do software engineers have the necessary competence for e.g. action research?

- We should start slowly: get commitment, select promising areas (e.g. estimation and risk analysis), and provide early feedback.
- The cost and benefits of EXDB itself should be regularly assessed. However, most companies do not even have an internal investment rate to compare different change efforts (new buildings, PC, furniture, or SPI).

The Experience Factory costs reported at NASA (11%) are not tolerable most places. A figure closer to 1-2% is recommended. For small companies, normal developers may constitute parts of the EF.

## References

- [1] Victor Basili, Gianluigi Caldiera, and Hans-Dieter Rombach. Experience Factory. In [5], 1994.
- [2] Victor Basili, Gianluigi Caldiera, and Hans-Dieter Rombach. Goal Question Metric Paradigm. In [5], 1994.
- [3] Victor R. Basili, Richard W. Selby, and David H. Hutchens. Experimentation in Software Engineering. *IEEE Trans. on Software Engineering*, SE-12(7):733–743, July 1986.
- [4] Reidar Conradi. SPIQ: A Revised Agenda for Software Process Support. In *Carlo Montangelo (Ed.): "Proc. 4th European Workshop on Software Process Technology (EWSPT'96)"*, pages 36–41, Nancy, France, 9–11 Oct. 1996. Springer Verlag LNCS 1149.
- [5] John J. Marciniak, editor. *Encyclopedia of Software Engineering*. John Wiley and Sons, 1994.
- [6] P. M. Senge. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Currency/Doubleday, 1995.



**Figure 1. Internalization and externalization in learning.**