Abstract
The paper studies the potential success of using software experience bases for organizational learning. First, a survey of some previous efforts in the area is presented, together with experiences from related fields. We note, that, modern network technologies like the Web provide new possibilities for distributed storage, access and dissemination of relevant knowledge. However, the main problem lies in internalizing formal knowledge (learning), not in externalizing experiences (model building).

Then we present the main results from an empirical study of four software experience bases in the Norwegian SPIQ project for software process improvement, done in 1997-99. The SPIQ project proposes a pragmatical, overall method for how to plan, establish, use and evaluate software experience bases. The hypotheses and research method are briefly presented, followed by a characterization of the studied experience bases and their actual usage and role in the associated companies. The most critical success factors are found to be incremental startup with a low ambition level, usefulness of the knowledge being offered, and, most importantly, stable company strategies and key personnel (all of this may seem obvious, but still not easy). Some advice for introducing and evaluating software experience bases are finally given.

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

1 Introduction
Knowledge management by experience bases is gradually getting into use. This applies 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 ultimate success criterion (what is “improvement”) is satisfied customers in the spirit of TQM or ISO-9000.

So, how can software organizations best systematize, organize and exploit previous experiences in order to improve their work? The learning perspective is crucial, both at the individual and organizational level. It is 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 engineers. Much too often, central process guidelines in a quality system are ignored and even sabotaged by the rank-and-file. They are perceived as “control”, and this may hinder both individual and organizational learning. To successfully convert passive information (models etc.) into active and operational knowledge, we must combine insights from software engineering and organizational sciences. Knowledge engineering technologies may be added later.

Some definitions: Explicit knowledge is 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) [22]. Not all tacit knowledge can be made explicit. Learning requires both formal training and informal information exchange. Many theories of learning exists, see e.g. [24]. Most theories operate with a learning process (cycle) as in Figure 1 below.

Figure 1: Internalization vs. externalization in learning.

Here, new knowledge is first internalized as new skills among practitioners (individual learning). This new knowledge and skills is then socialized into revised work processes and behaviour (group learning), which then is observed and externalized, and then combined to refine and extend the existing knowledge or experience base (organizational learning) – and so on in a new cycle.
The rest of the paper is organized as follows: Section 2 summarizes related work. Section 3 describes the actual study context, being the Norwegian software process improvement project called SPIQ [7]. Section 4 presents our success criteria, hypotheses, and research method, and four Norwegian experience bases and their characteristics. Section 5 presents and evaluates the results. Section 6 concludes the paper and gives guidelines for further work.

2 Related Work on Experience Bases and Related Areas

We will use the term experience base, not experience database, to avoid a too close association with traditional databases. Our experience bases may therefore be in the brains of people, on paper, on the web, in files, in spreadsheets, as well as in more classical Database Management Systems (DBMSes). However, computerized software experience bases (SEBs), especially for software process improvement (SPI), will be our focus.

As mentioned, there is a growing interest in using computerized experience bases in many fields. In part, this reflects modern information technology – like the web and other network technologies – enabling efficient and cheap storage of and access to information (knowledge) at “fingertip” range. For instance, there was a Workshop on Learning Software Organizations, related to the SEKE’99 conference in Kaiserslautern, Germany in June 1999 [5], and also a panel on this at the conference itself [8].

Existing work on SEBs spans from the one used in the NASA-SEL Experience Factory for almost 20 years [4], to ones that have been established three years ago at DaimlerChrysler [16]. Large IT companies often maintain proprietary experience bases, mainly holding summary information on previous projects. On the other hand, some SEBs have been abandoned just after start-up due to organizational turmoil, even if initial acceptance from software developers were good – see [18] about a SEB for an Australian software company for telecom.Yet other SEBs are still on the research level [14]. A central paradigm is to regard experiences including software as “capital”, that can be invested in (by generalization and refinement) and later harvested. This involves spending current effort and time for an unsecure, future return. However, especially software “rots” over time, i.e. loosing its relevance, see previous work on software evolution [19].

Other fields have introduced the term organizational or corporate memory to characterize an organization’s strategic assets, although not only from a learning point of view [1].

The knowledge engineering community has also worked on experience bases, often with emphasis on effective knowledge representations, deduction techniques etc., and towards a wide range of applications. The subfield of Case-Based Reasoning [2] has sprung up from this work, enabling simple reuse of similar, past information (“cases”) to better master new situations. We can also mention the subfield of Data Mining [13].

Social anthropologists and psychologists have studied how organizations “learn”, and how their employees make use of various information sources in their daily work. Much R&D effort has been spent on the “externalizing” flow, looking for valid experiences that can be analyzed, generalized, synthesized, packaged and disseminated in the form of improved models or concepts. For instance, to make and calibrate an improved estimation model, based on the performance of previous software projects. Explicit knowledge (written statements) may nevertheless be easily misunderstood due to lack of context and nuances.

However, the hard part is the “internalizing” flow. That is, how to make an impact on current practice, even if more updated knowledge may be convincingly available? Typical inhibitors are “not-invented-here”, mistrust (“been-burned-before”), lack of extra time/resources (“not-getting started”), or plain unwillingness to try something new or different (like adhering to formal procedures in a quality system). A study of maintenance technicians for copy machines indicated that such experts were most likely to ask their colleagues for advice, rather than to look it up in or even to follow the “book” [6]. Indeed, how many times have not computer scientists asked their office mates about commands in Word or NT-Windows, instead of directly consulting relevant documentation – although a “query” into the latter can be hard to formulate.

With a more explicit reference to our own field: the existence of software quality manuals, either on paper in thick binders (sometimes 1-2 m in the shelves) or in web documents on an Intranet, is no guarantee for their use in any form. In fact, since such manuals may dictate people on how to perform their job, traditional quality departments in many software organizations are not looked upon with high esteem by developers. So, if we are to succeed with SEBs to achieve learning, we must not carry the traditional “QA hat” of control.

Lastly, many of the ideas and techniques on quality improvement (TQM and similar) come from manufacturing, with rather stable products, processes and organizations. But information technology is characterized by rapid product innovation, not gradual process refinement [23]. One “IT” year is like a “dog” year (7 years) in other disciplines, and time-to-market seems “holy” (i.e. schedule pressure). The strength of many software SMEs (Small and Medium-sized Enterprises) lies in their ability to turn around fast and to convert next week’s technologies into radically new products and services. With reference to our SEB context, we must carefully adopt a set of improvement technologies that can function in a very dynamic environment – so how to manage constant change? Since both SPI and SEBs
assume that there is “something” stable that can be “improved”, we must pick our learning focus accordingly.

3 The SPIQ Project: Research Context and Guidelines for Software Experience Bases

The SPIQ project [7] was a Norwegian SPI project in 1997-1999, and stands for SPI for better Quality. SPIQ was supported by the Norwegian Research Council (NFR), and involved 3 research institutions, including NTNU, and 12 IT companies, mostly SMEs. These 12 companies have run over 20 SPI pilot projects, assisted by SPIQ researchers and partly by MSc and PhD students. A follow-up project called PROFIT is planned in 2000-2002.

The SPIQ goal was to increase competitiveness in Norwegian IT industry by assembling, downscaling and trying out a collection of mostly existing SPI methods. A pragmatical method handbook – in Norwegian – has been written [11], emphasizing a top-down/bottom-up approach using TQM [9] and QIP [4] ideas, an adapted ESSI process [12] to run SPI pilot projects, the Goal-Question-Metrics (GQM) method [3], the Experience Factory (EF) concept [4], and an incremental approach relying on action research [20].

In much of this, use of an EF/SEB is suggested, but underlining the need for business value with some short-term gains.

4 The Case Studies: Characteristics and Results

This section will characterize the SEBs from four SPIQ companies and discuss the results. These SEBs are in different phases of realization and deployment. The research method is mainly by participating observation and action research, and mostly qualitative data are presented, partly from questionnaires. Each of the four companies made individual SEB/SPi plans for how to improve themselves. These plans were read and commented by researchers. Goals and measurement plans were set up by the two parties together. The companies agreed to send progress and experience reports, and got rewarded economically from the SPIQ project on delivery.

4.1 Research Questions / Hypotheses

We have five general research questions/hypotheses:

H1: Cultural changes – is the SEB well-connected to a SPI program, approved by developers?

H2: Stability – has there been a stable SPI policy and stable local champions?

H3: Business value – has there been relevant cost/benefits by using the SEB?

H4: Current use/potential interest – is the SEB in actual use / are developers sympathetic to it?

H5: SPIQ feedback – have SPIQ SEB-methods been applied, with feedbacks to researchers?

Threats to internal validity: We have used standard indicators from the literature on most properties (stability, cost/benefits, “usability”, relevance), so in general we are on pre-walked ground.

Threats to external validity: the companies are very diverse, yet their problems seem classic. Still, the number and quality of the data points prevent definite conclusions.

4.2 The four studied software experience bases

Some brief company comments are:

X1: Their SEB tool for estimation tool by function points gave 10% better prediction than manual methods. It was pragmatically implemented as an Excel tool, connected to the existing, web-based quality system on their Intranet [17]. In spite of all project managers receiving half a day of training, the tool never took off, although an internal Gallup poll shows considerable interest. Their group of estimation experts is now almost dissolved, and project post-mortem reports (required by the quality system) are seldom picked up.

Main result and explanation: The SEB was never taken into active use, due to cultural barriers and due to personnel changes linked to large business changes – at least until 2001.

X2: Their SEB tool for project estimation by-analogy was made by a NTNU student using Oracle 2000-Designer and was based on company requirements. It was never populated properly with historical project data, and was in general not completed.

Main result and explanation: The tool never came into use, due to poor anchoring internally (acknowledged all the time), and internal reorganization and recent financial problems – but still informal contact about this.

X3: Their Corporate Memory is a pragmatical web-tool, containing administrative company information, developer competencies, core project data and an overall project model (a simple quality system). The company is characterized by a flat process organization, with many young and competent developers, all using the tool.

Main result and explanation: The tool is a success, due to appropriate ambition level wrt. contents and implementation. There is also a strong and stable company dedication to both organizational learning and SPI.
X4: Their Information Well is based on Microsoft Exchange repository tool. It stores strategic documents for the company [15], and is available on the Web for consultants sitting in other companies. Special responsibles will quality-check method documents related to technologies and business domains. The Information Well has been increasingly well received, as shown by annual polls, in spite of major reorganizations over the last year, where two key persons quit. A planned estimation tool extension has

Figure 2: Software experience bases at four Norwegian software companies.

<table>
<thead>
<tr>
<th>Property</th>
<th>Study 1: X1 estimation tool and experience base</th>
<th>Study 2: X2 experience base for estimation</th>
<th>Study 3: X3 corporate memory</th>
<th>Study 4: X4 Information Well</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business domain</td>
<td>Telecom application software</td>
<td>Banking software</td>
<td>Techn./admin. software</td>
<td>Admin. software</td>
</tr>
<tr>
<td># sw developers</td>
<td>600</td>
<td>220</td>
<td>140</td>
<td>400</td>
</tr>
<tr>
<td># SEB people</td>
<td>3-4</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Research partner</td>
<td>U. Oslo, NTNU</td>
<td>NTNU</td>
<td>NTNU</td>
<td>NTNU</td>
</tr>
<tr>
<td>SEB purpose</td>
<td>Better estimation</td>
<td>Better estimation</td>
<td>Assist sw dev. / SPI</td>
<td>Assist sw.dev. &amp; know.mgmt., better estimation</td>
</tr>
<tr>
<td>SEB platform</td>
<td>Web, spreadsheets. Parts of QA system</td>
<td>Oracle w/ 4GL</td>
<td>Web, files</td>
<td>MS Exchange Repository, w/ extensions</td>
</tr>
<tr>
<td>SEB contents</td>
<td>QA system (process models), estimation tool, risk assessment guidelines</td>
<td>Project exp. reports, simple estimation models</td>
<td>QA system, company info, CVs etc.</td>
<td>Guidelines, course material, project experience reports</td>
</tr>
<tr>
<td>SEB knowledge representation</td>
<td>Documents (containing models), spreadsheet entries and formulaes</td>
<td>DBMS-entries, documents</td>
<td>Documents</td>
<td>Indexed documents</td>
</tr>
<tr>
<td>SEB tool functions (more than insert, search, retrieve)</td>
<td>Estimate</td>
<td>Search by analogy</td>
<td>(Classify)</td>
<td>(Classify)</td>
</tr>
<tr>
<td>SEB processes /roles defined (“EF”)?</td>
<td>Yes</td>
<td>Some</td>
<td>Partly</td>
<td>Partly</td>
</tr>
<tr>
<td>H1: Cultural changes?</td>
<td>Some at start</td>
<td>Negligible</td>
<td>Yes</td>
<td>Some</td>
</tr>
<tr>
<td>H2a: Stable SEB strategy?</td>
<td>Until 1998, then turmoil</td>
<td>Until 1999, then turmoil</td>
<td>Stable, but modest SPI program</td>
<td>Fairly, in spite of reorganizations</td>
</tr>
<tr>
<td>H2b: Stable SEB champion?</td>
<td>Until 1998, then key person resigns</td>
<td>Until 1999</td>
<td>Yes</td>
<td>Mainly: but two key persons resign</td>
</tr>
<tr>
<td>H3: SEB business value?</td>
<td>Little (potentially very high)</td>
<td>None (never completed)</td>
<td>Yes</td>
<td>Yes (not estimation tool)</td>
</tr>
<tr>
<td>H4: Current use/potential interest?</td>
<td>(mainly dormant), Big interest via gallup pool</td>
<td>(none), Acknowledged need</td>
<td>Daily use, Positive feedback</td>
<td>Daily use, Positive Gallup poll</td>
</tr>
<tr>
<td>H5: SPIQ feedback?</td>
<td>Much interaction at start, hope to resume</td>
<td>Initial interaction, still contact</td>
<td>Continued interaction</td>
<td>Interaction up and down</td>
</tr>
</tbody>
</table>
been cancelled (these two tools “belonged” to different departments).

Main result and explanation: The tool is in daily use, due to its pragmatic relevance. There is also company dedication to organizational learning and knowledge management, less so for SPI.

5 The results and evaluation of these

This chapter summarizes and evaluates the four SEBs in light of our four hypotheses:

H1: Cultural changes – Only X3 and X4 are having a partial match here. However, “knowledge management” at X4 seems wider than SPI, and is backed by a different department.

H2: Stability – Again, the two latter companies scores highest, with X3 as best.

H3: Business value – Once again, OK for the latter two companies, since they both continue to develop it. There is also a big potential for X1.

H4: Current use/potential interest – Again, X3 and X4 have current SEB use, while X1 has signalled future interest. All the four companies have internal improvement initiatives, that do not necessarily involve a SEB.

H5: SPIQ feedback – Here, all companies must have been said to have contributed in one way or another – “failures” are not necessarily without a research interest.

H1–H5 summary: low-tech SEBs with a high business value in stable and committing organizations have the highest chances to succeed.

Some ideas on alternative hypotheses that could be tested in later studies, are

H6: What knowledge engineering (or other) technologies are most efficient to improve SEBs?

H7: What type of companies (or SPI programs) are most suitable for SEBs to be successful?

6 Conclusion

Some general observations related to our hypotheses H1—H3 are, see also [8] [10]:

H1: A precursor for all organizational change and SPI is commitment and consensus, all the way from top-level management to the rank-and-file. More training in action research may be needed.

H1: We should start slowly: get commitment, select promising areas (e.g. estimation), and provide early feedback. Using questionnaires is a practical way to assess developer's attitudes.

H1: 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 overly heralded.

The four SEBs are further characterized in Figure 2

below, with company characterization, SEB details and hypotheses summary. The results are presented in the next section.

H2: Sufficient organizational stability is needed, so that appropriate improvement initiatives can be sustained for a sufficient time period, say 4-5 years.

H3: The costs and benefits of a SEB should be regularly assessed. An overhead figure of 1-2% is recommended. In smaller companies, normal developers may constitute parts of the EF.

H3: An experience base is not a technical gadget, but a vehicle for organizational learning and process improvement. Hint: Demonstrate the usefulness of a SEB tool, and make it dynamic, e.g. show new experience items to investigate. Give feedback to the users and market the tool often. Maybe a reward system for insertion and reuse of experiences will boost the usage?

H3: The web is an excellent vehicle to store and disseminate information. The danger is, however, information overload and how to keep the stored information lean, updated and relevant.

H3: The syndrome of data cemeteries or “white elephants” is a related problem, cf. the fate of post-mortem reports in X1. Try make the SEB a communication channel, not a “dump”.

To sum up, we can say that incremental approaches are favored, e.g. using the web. The crucial part is internalization into workplace practice, not externalization as revised models in some repository tool. A pervasive, organizational commitment is ultimately needed, although we should start in the small. Lastly, stability in organizational support is a problem, like in all quality and improvement work.

Acknowledgements: We thank SPIQ colleagues in the involved companies and SPIQ co-researchers for their efforts and time. Likewise we thank Mikael Lindvall, Rose S. Pajerski, Carolyn B. Seaman, and Marv Zelkowitz at the Fraunhofer Center – Maryland for valuable comments on the paper.

7 References


Software Experience Bases: Some Results and Recommendations. Article. Jun 2000. Reidar Conradi. The paper studies the potential success of using software experience bases for organizational learning. First, a survey of some previous efforts in the area is presented, together with experiences from related fields. We note, that, modern network technologies like the Web provide new possibilities for distributed storage, access and dissemination of relevant knowledge. However, the main problem lies in internalizing formal knowledge (learning), not in externalizing experiences (model building). A Some useful recommendations for the application of the experimental process in software engineering are included. View. Show abstract. Experience-based techniques. Black Box techniques are used to derive test cases from the specification available. White Box techniques supplement the Black Box techniques and use the code to derive test cases and increase the test coverage. How do we test when there is insufficient/no documentation and also a limited time frame? What do we use to derive test cases and test results in such a scenario? Experience-based techniques come in handy in such situations. A Our Software testing trainers have more than 20 years of experience in software development, software testing and software training. Our Software testing trainers are specialists in Quality Assurance, HP UFT (QTP) & Advanced HP UFT(QTP), HP UFT (QTP) live Projects, BA, Selenium and MS Project training.