Journal of Universal Computer Science, vol. 8, no. 6 (2002), 570-580
submitted: 4/2/02, accepted: 3/5/02, appeared: 28/6/02 © J.UCS

What to Expect from Software Experience Exploitation

Kurt Schneider
(DaimlerChrysler AG, Research Center Ulm
P.O. Box 2360, 89013 Ulm, Germany
Kurt.Schneider@DaimlerChrysler.com)

Abstract: Software quality management and quality assurance are disciplines that require substantial knowledge of the methods and techniques to be applied. More important than a solid knowledge of methodology, however, is the ability to judge feasibility of approaches, and to tailor activities to the business unit culture and constraints. Software quality activities must be carefully integrated into an existing company or business culture. Making informed decisions requires more than knowledge – it calls for experience of what works and what does not work in a given environment. Experienced quality agents are a scarce resource. Exploiting a scarce resource - like experiences in software quality - more effectively is a straight-forward concept.

Five years ago, DaimlerChrysler set up a large research project with business units, called SEC (Software Experience Center). Its purpose was to explore opportunities for learning from experiences within and across different business units. Unlike more general approaches of knowledge management, SEC was entirely devoted to software processes: software development, software acquisition, and in particular software quality in both development and acquisition settings.

However, not all expectations that are often related to experience exploitation are realistic. In SEC, some of our initial expectations were met, others were not. This talk reports and reflects on our attempts to capture, engineer, and reuse experiences in the realm of software quality and software process improvement.

Keywords: Process improvement, systematic learning from experience
Categories: K.6.3, H.3

1 Introduction: Why Experiences?

Software quality management and quality assurance are disciplines of computer science that require a rare personal profile [Schneider 2001b]: A software background, good moderator and negotiation skills, ability to handle conflicts and a good standing with developers are some of the elements of an ideal quality agent (i.e. quality manager or quality assurance person). Of course, familiarity with QA techniques and experience as software developer are also desireable background aspects.

It is quite obvious that such a combination of qualifications is always difficult to find. One solution could be to hire newcomers, or graduates from other disciplines (business administration, electrical engineering, physics, biology etc.).

With training and transfer of experiences, they could fill vacant software quality positions [Siedersleben 2001]. Experience plays a crucial role: Software quality is one of the disciplines that relies most on experience**.**

In general, experience is in highest demand, where technical(software) knowledge needs to be combined with human judgement and the ability to make informed compromises.

If there is no clear set of criteria for "good" and "bad" compromises, experiences are often the only basis to decide. Therefore, the goal to "transfer experiences" is a cornerstone of a plan to transfer software quality qualifications. At DaimlerChrysler, we started a project in 1998 whose mission was to develop, try out, and apply techniques and mechanisms to make best use of (to "exploit") existing software engineering experience. Software quality was soon identified as an area to focus on.

Section 2 describes the reasons and goals for starting DaimlerChrysler´s SEC project. Section 3 provides a sample of our attempts to activate, capture, analyze ("engineer") and spread experiences and experience-related support is described. Typical expectations are grouped and analyzed by taking a closer look at our experiences. In section 4 we summarize our lessons learned by sketching what role experience exploitation will probably play in the future in one of our business units.

2 Goals for Experience Exploitation in SEC

In 1998, DaimlerChrysler set up the Software Experience Center research project (SEC). Its purpose was to explore opportunities for learning from experiences in different business units. Unlike more general approaches of knowledge management [Nonaka and Hirotaka 1995], SEC was entirely devoted to software processes: software development, software acquisition, and in particular software quality in both development and acquisition settings.

SEC was supposed to work intensely with several business units. In addition, we had envisaged three layers of systematic learning from experience:

- From project to project within one business unit;
- from one business unit to another;
- among an international consortium of companies.

Corresponding to this research mission, SEC had to be active in several places at a time. To compensate for this diversity of partners, quality assurance and quality management were always on the list of learning issues. Learning from experience requires *recurring* situations. With too many different topics and too many environments, a critical mass could not have been reached. Over the years, we tried many approaches to capture and exploit experiences.

At face value, experience exploitation is supposed to provide benefit for new projects based on the insights gained in past projects. However, there are several more expectations and assumptions, some of them hidden or implicit. During the years of the SEC project, we could often identify those expectations only when we faced them the *second* time. We are convinced that DaimlerChrysler is not the only environment in which those expectations tend to occur. Whoever tries to benefit from experiences should be aware of them.

It should be mentioned that there were many more activities on all levels – only those were picked for presentation that allow to make an interesting point, or that are transferable to other environments. Activities come from more than one business unit.

3 Software Quality Experience Exploitation

A first set of experiences with realistic and unrealistic experiences was presented in [Schneider 2001c]. Below, typical statements are sorted in four categories (subsections) that are relevant for making experience-based knowledge management work. By relating the statements – representing expectations or assumptions – to each other and to one of the core aspects of experience exploitation, our respective lessons learned will be easier to reuse.

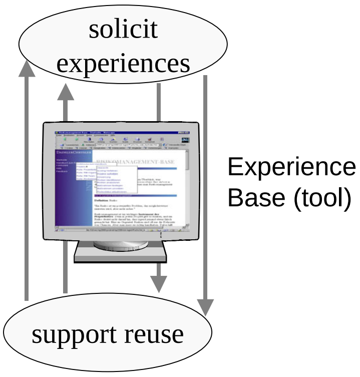

Figure 1: Core aspects of experience elicitation: tools, experience solicitation, and effective reuse. Arrows depict flow of experiences

We found the three aspects depicted in Figure 1 most influential to our SEC project. Similar projects should be aware of expectations related to each of the aspects. Experience exchange among the SEC consortium of companies is added as fourth subsection.

3.1 The role and importance of tools

Experience exploitation is often associated with tools. As in general knowledge management, there is a wide variety of functions that can be supported with tools. In many conversations, we encountered the mental image of an "experience database" as the core of all experience exploitation activities.

"The core of experience exploitation is the usage of an experience database (tool)" (wrong)

We started building tools, too. A structure for a piece of experiences was defined and called "quality pattern". Quality patterns [Houdek and Kempter 1997] were supposed to be the (highly complex) "data" in our experience database. Quality patterns define a document structure consisting of eleven headings that should be filled for each piece of experience (Figure 2). It was the purpose of this structure to capture all important aspects of an experience.

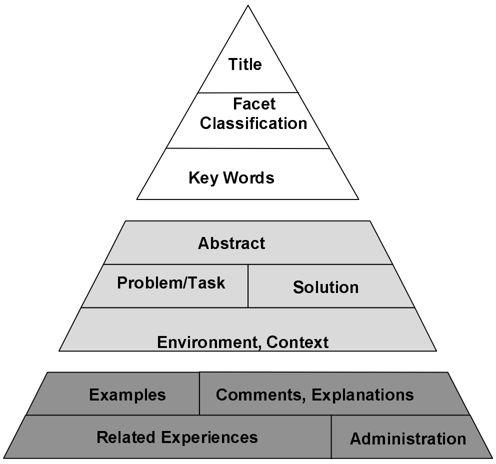

Figure 2: Pyramid structure of a quality pattern (acc. to [Houdek and Kempter 1997]*)*

Storing and searching were carefully implemented. However, as we report in [Schneider and Schwinn 2001] in detail, the database filled up slower than we expected. We had held a tacit assumptions that turned out to be not only wrong but a serious risk to SEC:

*"People are willing to defer gratification for sharing their experiences" (*wrong*)*

Researchers and managers both tend to assume that business unit employees see the need for putting experiences into a repository before they can – later – get something back from there.

This is sometimes true, but not in general. We met a few employees who had embraced the concept of experience exploitation and who really accepted to talk to SEC people (not write anything into a database!) several times only to *share* their information. Most expected to get something back in return. The above expectation is usually tacit [Polanyi 1966] and it can endanger the success of an experience exchange project: SEC needed to face the fact that any delay between "giving" and "getting" could mean to lose potential SEC participants. The patience of many employees is easily overestimated.

*"One can start with an empty experience repository and let people fill it" (*wrong*)*

Whenever experience databases, repositories, or experience bases (with many qualitative entries) are built, their designers tend to start with an empty sheet of paper. Managers rarely think about this question at all and expect experience bases to start quickly – which can only mean to start *empty*.

We learned the hard way how frustrating an empty repository tends to be for the few ambitious visitors: one can lose their support, too. This is a great risk for experience exploitation at large. As a general conclusion, any experience element that expects to be filled by "productive software employees" needs to provide a useful “seed” [Fischer 1998] from the very first hour of usage. SEC personnel needs to provide this seed, for instance by deriving it from other sources. An experience element includes not only explicit repositories, but also FAQ sections etc.

From the beginning of the SEC project, a computer-based collection of experiences (Experience Base, according to Basili´s terminology [Basili, Caldiera et al. 1994]) was considered an important component of a learning software organization. Our perspective and requirements for experience bases changed drastically during the project, and we ended up with an Intranet-based experience base for each process topic. We refined both the structure of those bases and the process to derive an adequate process.

It took us more than a year to realize what kind of seed was appropriate for an experience base. We narrowed the contents of a base down to everything related to one single specific quality sub-process (review and inspections; risk management; requirements engineering activity). We characterized the format of the contents as best practices and explicitly listed the components of a best practice (e.g., centering around a process description, containing templates, checklists etc.). We packaged all that material, discussed it with business unit representatives (still on paper!), and only afterwards put it into an Intranet-based experience base as seed.

In our experience bases, there are feedback mechanisms (contextualized email, FAQ, questionnaire, phone numbers) to sustain the flow of experience. In a business unit with several hundred users, this seeding procedure seems to work.

We conclude from our failures and the final successes that this choice of contents, format, and presentation indeed leads to a working seed – and this seed can systematically be constructed. A tool like an experience base is still considered crucial, but less essential than eliciting useful experiences. Elicitation is a highly manual task since people do not tend to just "provide" experiences, as has been pointed out above.

Politics and support is far more important than any tool can be in experience exploitation:

*"Management commitment and support is indispensable" (*correct*)*

It is sometimes difficult to convince higher management to devote money and valuable resources for the sake of what they perceive as “background, cross-functional” activities like experience exploitation. For lack of top-down support, our business unit partners more than once asked us to start with bottom-up activities. Supporting quality circles with rather simple activities, comparing quality manuals, and especially doing some more quality coaching were among the resulting activities.

Most of those basic support activities were even rather successful (produced benefit for the business unit). However, they hardly contributed to our overall goal of spreading systematic learning from experience. A business unit in which there was more management commitment, resulting benefits were much higher. In essence, there seems to be a critical mass of investment in experience exploitation. With a too small number of potential experience re-users, success is much less likely. Below a certain threshold, most activities will vanish. Above the threshold, there is a tendency for mutual reinforcement. Management commitment must, therefore, not be measured only in words or in funding, but also in concrete action.

3.2 Often underestimated: soliciting experiences

Tools are not the most decisive aspect when building an SEC. We found it far more important to solicit a sufficient number of high-quality experiences that could provide benefit early. While building tools is technically easier than we though (but less relevant), experience solicitation turned out to be more relevant, but much harder to carry out effectively. One misconception was the following:

*"Valuable experience is everywhere: it just needs to be collected" (*both true and wrong*)*

In most environments, there is a lot of experience residing in people. At first glance, the problem just seems picking up all that experience. However, it is a popular misconception that there is much value in *raw, superfluous* experience. A lot of experience elicitation effort can be wasted in only capturing the thin and unsettled spread of "experiences" on whatever comes to anybody´s mind. We learned that the assumed and the perceived values of those "experiences" often deviate drastically. Much more effort must be spent on analyzing raw experiences than on getting them in the first place. In our four-year experience thin-spread observations on many subjects have almost never been reused, whereas deep experiences going beyond rumors could effectively be turned into best practices and was actually reused.

*"Interviews help to collect experiences" (*correct, but...*)*

Several times a topic was raised that could not be answered from the given project environment. In fact, a demand in the project led to investigating a question (e.g., quality planning for the roll-out phase of a large administrative system). In this case, a couple of SEC people arranged interviews with members of one or more other projects in order to capture (and reuse) their experiences. We called this a "pull" situation as opposed to a "push" situation in which experiences were collected without concrete demand and then "pushed" into the projects [Wieser, Houdek et al. 1999]. Interviews are a highly effective way of soliciting more details from a knowledgeable and experienced person. We found them far less effective to lay the foundation of a topic (e.g. roll-out plans), since interviewers often had a hard time finding the right questions. The following approaches were more appropriate when concrete questions were missing.

*"Any gathering of knowledge workers can be a source of experiences" (*correct*)*

In order to establish a direct experience exchange, we established a quality circle: Quality agents from different projects (within one business unit) gathered once every other month. SEC supported this circle by organizing the meeting, setting up an agenda, and often by providing a first input talk to stimulate discussions. We also considered the open exchange of opinions and lessons learned a good source of experiences. We captured what we understood. The last step was to feed some of the findings in one of our experience bases (see Section 3.1 above). Again, using the culture of a quality circle and the atmosphere of sharing was more crucial than the final step of putting stuff on a computer.

*"There are patterns in documents that point to interesting experience" (*correct*)*

Besides workshops, interviews, and quality circles, we assumed to find experiences - or traces of experiences - in documents. Our technique of *delta analysis* compares two documents that are closely related (e.g., two subsequent versions of a plan; a graphical model and its textual description; a table and an excel sheet on number of errors). Differences between the files were identified. The central heuristic of delta analysis claims that a difference ("delta") could be caused by (1) conscious decision or (2) mistake, both of which need closer attention. Often, decision rationale contained a lot of experience. We investigated the reason of such a difference by other means (focused interviews, more documents, etc.). There may be other patterns that point to experiences, but delta analysis is simple and often effective. The proper solicitation of experiences needs to be carried out off-line as soon as it has been identified by a document pattern.

"The best way to capture experiences is by first sharing experiences" (often correct)

We had long-standing relationships with a business unit. As coaches or consultants, we had supported their software engineering and quality efforts. A starting idea was to use this on-going support as a source of real experiences. In practice, one SEC person was added to the initial support team. This person assisted project supporters, and at the same time tried to capture experiences and reflect on them. It is easier for many people to discuss and criticize an immature proposal than to come up with their own new one. This is why previously captured experience can be used to provoke new experiences, often in the form of feedback and criticism.

*"Meetings can be optimized for experience elicitation" (*correct*)*

We experimented with different formats of interviews and workshops to elicit experiences from a group of people. LIDs is a resulting technique that is optimized for efficiency [Schneider 2000a]. Within half a day, a group of people who have been involved in a common task (e.g., test phase of a project; organizing a fair or conference) are assisted in identifying, discussing, and documenting what they remember and consider important about the activity. LIDs helps to collect and tie in existing documents from the activity. The LIDs technique has been applied many times in different environments. It is described in [Schneider 2000a] in detail.

*"One can learn across business units by comparing documents" (*correct*)*

This expectation was met in our experience. Sometimes, text blocks or formatting concepts were adopted; more often, a difference led to discussions of goals and priorities. In a way, experiences (as rationale for both manuals) were involved. It was the conclusion derived in intense discussions that turned out to be the benefit. What did not work was an attempt to directly copy any significant amount of text, figures, or processes.

3.3 Making experience reuse happen

Only during the second half of the SEC project did we focus on the other end of experience exploitation: on spreading and using what had been captured and engineered. The above experience bases are one of the resulting elements - but only one. It turned out that effective spreading takes as much inspiration and active support as experience elicitation or tool building.

*"Lessons learned will be read and considered as soon as they are stored in a database or Web tool" (*very wrong*)*

Many researchers and authors of experience-based material consider their task completed when they have inserted their contribution to a database, experience base, or Intranet page.

This assumption is very wide-spread, but it is tragically wrong. As others have pointed out in their domains [Stolze 1994], there are currently so many sources of information on a wide variety of media that most people suffer from an information overflow. An “available” information is just not good enough. Software quality agents really need a piece of experience at the time and in the way they can best integrate into their respective work assignment [Stolze 1994]. The tragic element lies in the fact that excellent material can sleep on a wonderful experience base system – without ever being found and used by those who desperately need it. We reacted by not only advertising our experience bases, but also by tightly integrating them in a system of training courses, newsletter notes, and information briefings for the subject conveyed (e.g. risk management) [Schneider 2001a]. Once a critical mass of users and usage frequency has been gained, this assumption may turn true a little more.

This wrong expectation is based on a deeper one:

*"In principle, everyone wants to learn from others' experiences" (*often wrong*)*

Many experienced workers assume that the need for experiences is so obvious to everybody that there is almost no need to convince them. Experience exploitation seems self-supportive in nature.

It came as a surprise to us in the first year that people were often very open and giving in interviews, but never took the initiative to search, read, or apply any material we had retrieved from others. Our analysis led us to believe (a new, unproved assumption!) that deep inside they did not trust the quality of the offered experiences. As a second reason we identified the unclear benefit from searching or asking SEC. There seems to be several reasons that keep potential users from even touching an experience repository. Our conclusion is to consider “encouragement for reuse” as another active task of SEC. In this perspective, workshops, quality circles, and in particular coaching and consulting situations were ideal to re-infuse experiences, not only to capture them.

*"Training programs are still needed despite all experience repositories" (*correct, but not sufficient*)*

Most software process improvement activities ran into a phase in which we had to teach project participants about some aspects of the modified process. For example, reviews and inspections were introduced with a short talk. In the long run, it is more efficient to teach standard topics to several projects at a time. For that reason we developed a training program that was mainly influenced by the experiences collected [Schneider 2001a].

Training courses cannot meet all requirements of a business unit in reality. There are always people who miss the course, or who consider it trivial, or who would prefer a self-directed computer-based training. Others would prefer personal exchange. Daily work in a complex situation, such as quality management in a software business unit, also confronts quality agents with tasks they have not yet learned to deal with. In those situations, we need to open more than one channel of communication and collaboration for solving problems; collaborative learning on demand is a typical mode of working in knowledge-intensive domains. In an EU-funded project (Coronet: IST-1999-11634) we try to learn more about this delicate system of learning and working [Schneider 2001a].

3.4 Learning between companies

*"Experiences from other companies are highly useful and easy to transfer" (*partly correct*)*

Large companies have many software problems and constraints in common. Therefore, it could be rather easy to exchange experiences.

The international SEC consortium of companies exchanges experiences at two meetings per year. It soon became obvious that effective experience exchange depends on good preparation. Since then, we are experimenting with several ideas of making those meetings more effective, and producing tangible benefit. Among the approaches used in the consortium was LIDs [Schneider 2000a] and defined processes to submit a topic to discussion, to prepare for a session beforehand, and to document its results. In SEC meetings, software topics like component-based development or quality gates are addressed, and aspects of practical relevance are discussed – based on experiences. As an intended side-effect, all participating company representatives learn what works for experience exchange in such a multi-company group (and what does not work so well).

Large companies have their own company cultures, and experiences must always be seen in the light of those cultural constraints. A major obstacle to effective experience exchange first was a different attitude towards failure and success. While some companies prefer to report successes as media for their "positive" experiences, others insist on collecting "negative" experiences, too. They argue analyzing and avoiding a mistake in the future can be more beneficial than denying problems. In their view, common problems are among the most promising areas for exchanging experience. A second issue is the format of exchanging experiences effectively. It took SEC several meetings to slowly evolve processes and techniques that are tailored for the situation. We have seen that international companies can effectively exchange experiences - but they need time and imagination to bridge different cultures. And it is not easy.

4 Conclusions for Our Future Projects

The purpose of the SEC project was to explore opportunities of experience exploitation. Business projects and our own future projects will build on the findings, some of which have been reported above. We are going to spread the basic insights into all our research projects, and there will be a number of documented experiences. Earlier findings were reported in [Houdek, Schneider et al. 1998], [Landes, Schneider et al. 1999], [Houdek and Schneider 1999], [Schneider 2000b], [Schneider and Schwinn 2001], [Schneider 2001c].

In an upcoming project that focuses on specific aspects of software quality in agile software projects, we plan to employ the concept of Experience Bases that has been outlined above. Most effort should go into defining and discussing the process and its elements in the light of collected experiences. Those core discussions should be supported by techniques like LIDs [Schneider 2000a]. On all levels, LIDs has found amazing acceptance. It makes little assumptions about application environments, and it is a light-weight technique. Whenever it comes to capturing experiences of a team or group, LIDs will be our first choice. Analogues to quality circles usually meet expectations, as well as delta analysis in all its variants.

We also hope to continue our experience exchange with other companies – now focusing on those software topics that are dominant in our new projects, or in our business units. We expect to get more and more dividends from all our investments into the consortium.

We have learned some lessons the hard way, and we have worked hard to resolve numerous details of approaches that were expected to work right away. Some expectations may be realistic in an ideal environment; in several real settings, tough, constraints make those expectations unrealistic. Therefore, our lessons learned need to be read with a grain of salt. An honest analysis of company culture, experience exploitation goals, and (explicit and implicit) constraints is indispensable in order to avoid traps. At the same time, we are convinced that the lessons learned about wide-spread expectations in experience exploitation contain a general truth that does not depend on our particular environment. Hopefully, others will thus be able to recognize and avoid known traps and pitfalls. Experience exploitation is a high-potential area of knowledge management – but it is not as easy as it may first seem.

References

[Basili,, Caldiera, et al. 1994] Basili, V., G. Caldiera, et al.: Experience factory. Encyclopedia of Software Engineering. J. J. Marciniak. New York, John Wiley & Sons. 1: 469-476.

[Fischer 1998] Fischer, G.: “Seeding, Evolutionary Growth and Reseeding: Constructing, Capturing and Evolving Knowledge in Domain-Oriented Design Environments.” Automated Software Engineering 5(4): 447-464.

[Houdek and Kempter 1997] Houdek, F. and H. Kempter: Quality patterns - An approach to packaging software engineering experience. Symposium of Software Reusability (SSR'97).

[Houdek, Schneider 1999] Houdek, F. and K. Schneider: Software Experience

Center. The Evolution of the Experience Factory Concept. International NASA-SEL Workshop.

[Houdek, Schneider, et al. 1998] Houdek, F., K. Schneider, et al.: Establishing Experience Factories at Daimler-Benz. International Conference on Software Engineering (ICSE-20), Kyoto.

[Landes, Schneider, et al. 1999] Landes, D., K. Schneider, et al.: “Organizational Learning and Experience Documentation in Industrial Software Projects.” International Journal on Human-Computer Studies (IJHCS) 51(Organizational Memories).

[Nonaka, and Hirotaka 1995] Nonaka, I. and T. Hirotaka: The Knowledge-Creating Company, Oxford University Press.

[Polanyi, M. 1966] Polanyi, M.: The Tacit Dimension. Garden City, NY, Doubleday.

[Schneider 2000a] Schneider, K.: Active Probes: Synergy in Experience-Based Process Improvement. Product Focused Software Process Improvement (PROFES 2000), Oulo, Finland, Springer.

[Schneider 2000b] Schneider, K.: LIDs: A Light-Weight Approach to Experience Elicitation and Reuse. Product Focused Software Process Improvement (PROFES 2000), Oulo, Finland, Springer.

[Schneider 2001a] Schneider, K.: Experience-based Training and Learning as a Basis for Continuous SPI. European SEPG, Amsterdam.

[Schneider 2001b] Schneider, K.: Qualitätsmanager/in: Wunschprofil und Erfahrungsaufbau. SQM 2001 congress, Bonn, Germany, SQS AG.

[Schneider 2001c] Schneider, K.: Realistic and Unrealistic Expectations about Experience Exploitation. Conquest 2001, Nürnberg, Germany, ASQF Erlangen.

[Schneider and Schwinn 2001] Schneider, K. and T. Schwinn: “Maturing Experience Base Concepts at DaimlerChrysler.” Software Process Improvement and Practice 6: 85-96.

[Siedersleben 2001] Siedersleben, J.: Sidestep - die Informaik-Initiative von sd&m. SEUH 7: Software Engineering im Unterricht der Hochschulen, Zürich, dpunkt.verlag.

[Stolze 1994] Stolze, M.: Visual critiquing in domain oriented design environments: Showing the right thing at the right place. Artificial Intelligence in Design'94. J. S. Gero and F. Sudweeks, Kluwer Academic Publishers: 467-482.

[Wieser, Houdek, et al. 1999] Wieser, E., F. Houdek, et al.: Push or Pull: Two Cognitive Modes of Systematic Experience Transfer at DaimlerChrysler. 11th International Conference on Software Engineering and Knowledge Engineering (SEKE 99). Workshop on Learning Software Organizations., Kaiserslautern, Germany, Springer.