

LIDs: A Light-Weight Approach to Experience Elicitation and Reuse

Kurt Schneider

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

Abstract. Building common ontologies, setting up measurement programs, and conducting interviews are valid techniques to start eliciting knowledge and experience for later reuse. However, they appear too expensive and too resource-demanding in many industrial environments. This paper presents a light-weight approach to capturing important reusable material, including experiences. The LIDs approach includes defined process steps and templates to create reusable material for different kinds of users. It has emerged pragmatically from our long-standing process improvement work with different business units.

1 Introduction

At DaimlerChrysler, there is a long-term initiative to foster systematic learning from experience in the software domain (SEC: Software Experience Center). This initiative is coordinated by Corporate Research and Technology. It has started in 1997 and has since established working relationships with several business units and application projects. A major goal is to establish local SEC groups that are supposed to implement, spread, and act as a catalyst for experiential learning.

When the initiative started in 1997, its initial concepts were mainly those of the experience factory [1]. Our early adoption and interpretation of this concept first led us to

- a focus on GQM-based (Goal Question Metric [3]) measurement programs as the primary source of input;
- development of role concepts and organizational models for a future DaimlerChrysler Experience Factory [11];
- development of mechanisms and prototypes of an Experience Base. According to Basili [1], an Experience Base is the persistent storage location or device that acts as organizational memory for an experience factory.

However, we also identified substantial differences between the situation at NASA-SEL [2], where the experience factory concept had been first applied, and our situation at DaimlerChrysler business units [10]. Among those differences were

- different models of funding such an initiative, resulting in more limited possibilities;
- application projects are not as similar to each other as they were at NASA, so that measurement results are more difficult to compare;
- differences in attitudes towards this style of improvement, and a different degree of awareness for the importance of software.

These and other differences were identified almost at the beginning. As a consequence, we tried different variants of the experience factory concept in different business units. The most important lessons learned were reported in [10, 11]. One of the findings was the need to add several light-weight elements to the experience factory as originally developed at NASA-SEL [2]. Light-weight means: low-effort and low-cost. Our follow-up work led to some interesting new concepts that grew and keep growing in practice. The LIDs concept is a recent, and rather successful example.

Section 2 discusses why light-weight approaches need to be considered in experience exploitation. The specific tension of light-weight approaches is characterized through their goals and challenges in section 3. Section 4 uses the foundation laid by the previous sections and presents the core of this paper: the LIDs (light-weight documentation and reuse of experiences) approach. Section 5 examines two cases in which LIDs were applied in practice; section 6 concludes.

2 Light-Weight Approaches

In Jonathan Grudin's seminal paper [6], he asks: Who is the beneficiary and who has to do the work? Grudin basically argues a CSCW (computer supported cooperative work) tool will fail if it puts heavy burdens on the shoulders of some while giving benefit to others. Of course, the same observation can be made in many other domains. CASE tools, software documentation in general [19], and organizational learning are obviously subject to similar phenomena.

Grudin's thesis [6,7] can be put into an equation:

$$Utility = Benefit / Effort$$

What looks obvious at first glance has a number of non-obvious implications:

- This equation refers to single individuals as well as to teams and to business units. It seems to describe an intrinsic human pattern of reaction. Something is considered useful if its benefit is higher than the effort invested in it.
- It is not the *net* benefit or the *net* effort that counts for an individual or stakeholder - it is their *perceived individual* portion of the benefit and effort. Perception is different in different perceivers, and it is definitely not only based on monetary values.
- Altruism, force, or higher payment can have a short-term impact on the perceived benefit. However, intellectual tasks seem to require different (intrinsic) incentives in a longer run.

- There is a time-dimension to the equation. It is not even sufficient to balance *overall* benefit and effort for an individual. It must even be balanced *within a reasonable time frame*. For example, the benefit of writing an experience package may pay back only in the next project. For someone who will not be on this new project, there is no benefit in even the most perfect reuse. As a consequence, there will be less enthusiasm for (experience) documentation than if there could be (even partial) reuse during the current project.
- Obvious as it may seem, the equation is still often neglected. Poor balance between effort and benefit for any of the key participants makes approaches fail that look promising otherwise.

Most importantly, the utility equation offers two roads to succeeding in a domain like systematic learning from experiences:

- The benefit can be high. This seems to be the more popular road. Since many approaches require substantial investments and commitments [4, 9, 15, 17], they must provide high benefit in return. According to Grudin, those high benefits must be perceived and appreciated by those *same* people making the investment.
- The effort can be kept low. This second road is often overseen: utility can be high even when there is only limited benefit! I call those approaches "light-weight". In some way, light-weight approaches demand less from their users or participants. As a consequence, expectations are usually lower and can, thus, be easier met or exceeded. In fact, there often is no real chance to apply heavy-weight approaches, whatever the later benefit may be.

There are several light-weight approaches described in literature. From an e-mail based group memory system (GIMMe: [12]) to organizational concepts of experience brokers [8], they vary in complexity or size, and they exist in different domains.

3 Goals and Challenges of Systematic Learning from Experiences

The LIDs approach is tuned for systematic learning from experience. For successful experience exploitation, one needs to ask:

- What is the intended benefit?
- Rather than: What can we collect?

In the terms of McMenamin/Palmer [13], the first question touches on essential requirements, whereas the second one is only a means to an end.

3.1 Experience Activity Cycle

When dealing with experiences, there is a cycle that comprises the principal activities (Figure 1). Use cases of a Software Experience Center can be embedded into this model. Arrows indicate dependencies and the direction of support between activities.

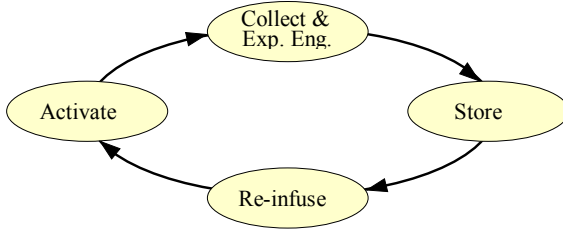


Fig. 1. Experience activity cycle

Activate indicates that experiences usually need to be actively elicited from people, documents, and situations. There is tacit knowledge [18] and experiences, or implicit knowledge, as Nonaka and Hirotaka [16] call it. The owners of such kinds of experiences cannot externalize it without assistance: the owners may consider it trivial, insecure, difficult to verbalize - or it may be so deeply ingrained in procedures they master unconsciously.

There is not one unique starting point in this cycle, but each "experience transaction" may start at a different activity. It may seem obvious to start at *activate*, but when the Software Experience Center is already "seeded" (i.e., it already has some experiences and information gathered), the next transaction may start by re-infusing the seed. This will in turn raise and activate new experiences, and so on.

Collecting experiences just means to identify experiences, take them into consideration and add them to the body of experiences regarded "in" the Software Experience Center. **Experience engineering** is a discipline supported by experience factories [1, 2]. Experience engineering adds value to the previously raw experiences, it refines them and is, therefore, at the core of systematic exploitation of experiences. Typically, context influences are analyzed, experiences are compared and condensed. Conclusions are drawn that can be reused elsewhere. A Software Experience Center is not a passive shelf to just store stuff - nor is the Experience Base. However, benefit and effort must be carefully balanced also for experience engineering. This is reflected in the LIDs approach below.

Storing refers to the technical aspects of making experiences persistent (e.g., on paper, as electronic documents, on the web). Before the technical aspects even come into play, experience-related material needs to be structured for easier reuse. This is an intellectual task we consider part of experience engineering rather than storing. Storing includes providing templates for uniform presentation of contents, putting contents and documents in databases, organizing file structures, formatting files for printing and web viewing, and the like.

Re-infusion is again a neglected (or under-estimated) activity. The cycle does not end when, for example, a best practice description is "available" on the Intranet. Many of our early experience base prototypes were technically sound. Potential users just did not care to look at them every time they could have reused material that "was available" there.

In a sense, we were facing what could be called

The Early-Web-Problem:

When there is very little useful information compared to very many problems seeking help, and when there is no way to tell how reliable any given information is - why would anybody bother to search the experience base? In most cases, nothing (useful) will be detected anyway, and in all cases, effort will be spent.

We have seen the web to reach critical mass in the meantime. There is enough useful information spread over thousands of servers. In our environment, however, an experience base will not be able to reach critical mass over a wide range of topics. The alternative is to provide a limited focus in which it can achieve good coverage of what users need. The borders of this focus must be very clearly communicated to avoid wrong expectations. This is particularly important for light-weight approaches. In short, the re-infusion mission stays the same: to actively provide the right experience-based information at the right time and in the right way to the person who needs it [20].

3.2 Who Are the Users?

In the beginning of our work, we tried to build one big experience base for a heterogeneous community of potential users. Each user would have access to the full set of experiences and related documents. Sophisticated search mechanisms would assist to detect those elements a user could best apply in his or her actual problem [9,14].

We abandoned the One-Big-Base approach when we realized that there are some intrinsic problems:

- There is always sensitive information that should by no means become visible to outsiders of a subgroup of users. In a single big base the danger to violate privacy is ubiquitous.
- Even the best search mechanisms will always deliver many things that are not appropriate for a particular user. Worse than picking a slightly *wrong topic* is picking a presentation for a different audience and perspective: Depending on a user's role, a more or less abstract, concrete, or detailed format must be used. We see people react strongly when they get an *inappropriate style* of information.
- The One-Big-Base approach assumes there are many rather small experience packages around. They may be linked, but when a user receives a list of packages, it is the user's responsibility and job to weed out the jungle of links, and to synthesize what remains of it. Few users want to shoulder this burden.

As a consequence, we decided to go for several small sub-bases, each of which would represent one topic area (like reviews, test planning etc.). To provide focused information, we differentiated between user roles that are to be supported in each of the sub-bases (Figure 2):

- **Naive or novice users** browsing through the experience base. Naive and novice users want to know what the topic is about and where it fits into their world. A *shopping window* is the right way to address these users: they get a first idea of a topic, including expected benefits. If they are interested, they need to enter the store and talk to a person. A shopping window can be implemented as paper flyer or brochure, or as an easy-to-read set of web pages. Readers are encouraged to contact an SEC representative.
- **Mass customers** looking for ready-to-use templates and guidance. Mass customers know the basics and are looking for immediate support in a limited area. They typically search for templates; sometimes a best practice description might be welcome. Mass customers require no personal service. They prefer a *self-service shelf* where they can grab what they need.
- **Long-standing customers** requiring tailored information and in-depth coaching. Long-standing customers expect best practices tailored to their domain and environment. They should also be offered active feedback opportunities, which will then trigger best practice revision.
- **Experience engineers, project supporters, and librarians** who support all of the above. Experience engineers structure the contents, librarians put the documents in place technically (database, web, etc.). Project supporters are SEC coaches that help people to adopt an experiential working style. They all need to look under the hood. They can handle unpolished documents and isolated experiences.

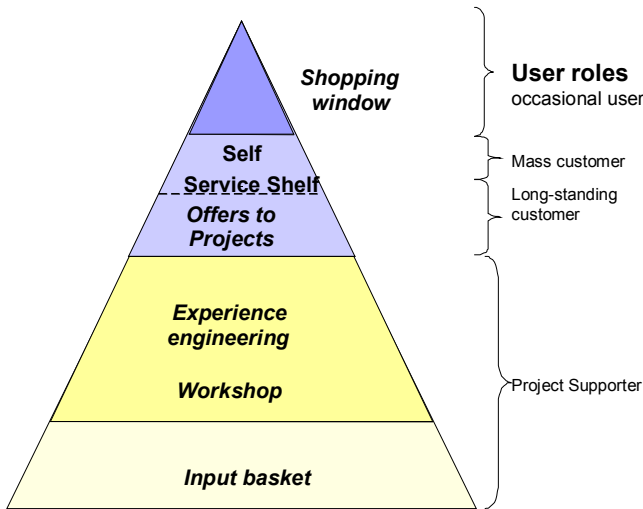


Fig. 2. User perspectives on an experience base (width indicates amount of information)

Others have provided one source of information that is filtered for different user groups [4, 5]. Filtering seems to work well in technical information bases, where the level of detail is the distinctive property between different user groups. In an experience base on software quality management, however, it is not so much the *level of detail* that makes the difference, but the *style and purpose* of an information.

Novice users need more motivational style, while customers want to know What to do, and How to do it. Experience engineers must know in addition How a best practice was derived and How it can be tailored. Although similar in content, there is almost no overlap in actual texts or documents.

Whether a company decides to go for filtering or for separate sub-bases is a technical question. We found it essential, however, to avoid mixing the user perspectives *conceptually*. It is crucial for an experience-base (and the whole Software Experience Center) to provide appropriate, tailored information to the users: better not to provide a piece of information at all than one that is tuned for a different perspective and purpose!

3.3 A Common Misconception

In most environments, there is a lot of experience residing in people. At first glance, the problem just seems picking up all that experience so it does not dissolve into vague rumors or oblivion. However true the general observation, 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. This is in line with the original experience factory and GQM concepts [3]: find a goal first, then start collecting. Not vice versa.

When one starts to search for experiences in a less rigorous way, one can take advantage of the experience-loaded material that has already been produced:

- In the course of working, there are many informative documents produced and many observations made. A lot of effort and intellectual energy went into them, but they may be difficult or impossible to interpret and to put in perspective for an outsider. These assets are spread over several servers, file systems, data bases, and printouts.
- There is also a lot of garbage: old versions, unfinished documents, double copies with slightly different names, wrong models and dead-end plans. An outsider has not the slightest chance to distinguish garbage from assets.
- For documents that have not been touched within two or three months, *not even insiders and participants* understand the meaning or status any more - nor can anybody at that point of time evaluate the merit of a document. This ability decreases (roughly in exponential order) over time.
- The above three observations hold not only for documents, but also for what people consider and are able to easily verbalize as their experiences. Doing complex work may bring even tacit experiences to bear, and sometimes can make them conscious - for a while. After a much longer time than a few weeks, they sink back into unconsciousness or even oblivion. Then, experience elicitation will need much more activation effort.

3.4 Challenges for Light-Weight Approaches

In the above sections, all foundations have been laid and the motivation has been given to attack the inherent challenges of systematic learning from experience in a light-weight way. The general challenge is to support one or more defined user roles at low cost and effort. Concepts to achieve that are

- little and inexpensive activities that serve experience elicitation only;
- active elicitation by doing useful things for the customer: Most valuable practical experiences come from *doing* real work in practice;
- engineer experiences modestly first - just enough to save the material and prevent it from being forgotten.
- Further experience engineering towards higher degrees of reusability can be added off-line, and later. Select carefully and focus on the most promising topics.

4 LID: Light-Weight Documentation of Experiences

During the last sections, the background has been sketched before which experience engineering must take place. This has also set the scene for the core of this paper, the light-weight experience exploitation approach called LIDs. LID stands for Light-weight Documentation of Experiences. "Lid" is also the English word for the cover on a pot. The associations of putting things together in a pot and then to cover it up with a common lid, is also close to what LIDs do. Hence, the name.

4.1 The Ideal Case for LIDs

LIDs biggest potential is in capturing experiences and supportive material shortly after a one-to-three month activity or project phase *of importance* is over.

Examples include

- organizing and carrying out a large workshop or expert meeting that has remarkable or unusual conditions, features, or outcomes (see section 5, case A);
- introducing a new technology of modest size into a working group (reviews rather than object oriented design);
- coaching a project activity like writing test plans and test cases, or tailoring and starting risk management in a business unit (see section 5, case B).

We have also tried to apply the LIDs approach to longer phases of projects. This seems possible but requires the more patience and energy the longer the activity takes. In general, LIDs work best under the ideal conditions sketched here. The more conditions hold, the better it will work.

On the most abstract level, the LIDs approach is as simple as this:

1. Shortly after an activity (see above) is over, hurry to make some of the key participants tell you the story. Use a brief checklist of questions and things you want to learn (see below).
2. Write the story down together with them on only a few pages, and collect all the newest versions of those documents (e.g., slides, plans, reports) that were mentioned in the story, and that had some relevance.
3. In particular, look for templates and close-to-template documents.
4. Put all of that in a directory (like in a pot), link the story to the documents it refers to and use the story as the (only) access mechanism to the ingredients in the pot (the lid covers the pot).

The rationale behind the above steps is

- It is essential to capture experiences when they are still fresh. People usually like to talk about their recent adventures, so they will not consider it effort. The checklist is important to avoid getting lost in war stories. It is advantageous to have an SEC project supporter to do the on-line write-up.
- The story should remain just this: a (non-fiction) story that is good to read and not too long. Any details must be deferred to the attached documents. The story must stay comprehensive and should avoid inside slang.
- Templates are among the most reusable documents. Therefore, they deserve special attention. Even documents that are not real templates, but still so general that they could be easily turned into templates, should be marked as "low-hanging fruits" for reuse.
- Putting everything together in one storage location makes it easier to compress, copy, and transfer the material in one piece. Readers will find all related material in one place. They will not have to worry about versions and garbage.
- The lid also protects the "pot contents" from being modified or deleted by others. Therefore, the lid must be a restrictive access mechanism. According to Figure 3, it provides searching and output operations, but not modification or direct input. This makes a LID a read-only access.

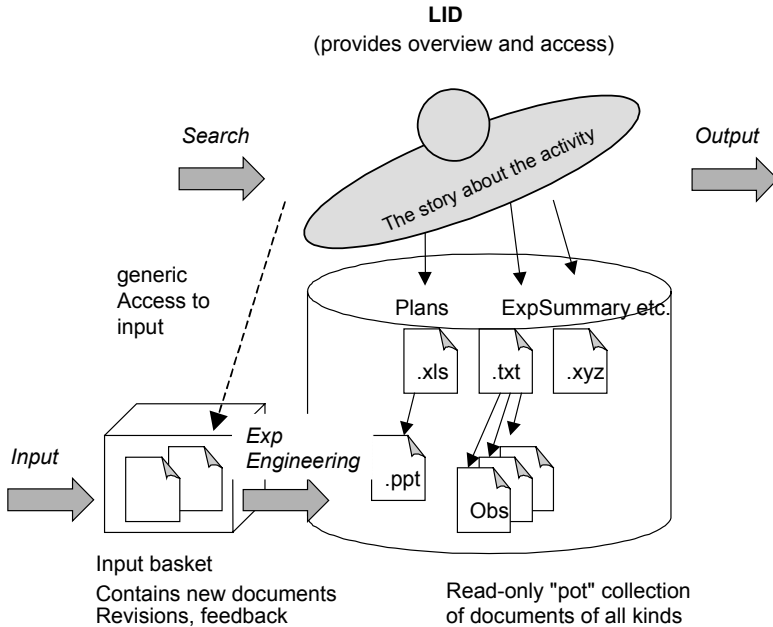


Fig. 3. A LID as overview and access mechanism to structure documents and experiences

Figure 3 shows the principal elements behind the LIDs approach, and the operations supported (gray arrows):

- The pot holds and saves all kinds of existing documents, brief experience snippets, and documents related to the activity.
- An input basket will hold any future reactions to the activity covered under the lid. It also will contain reports on similar activities, before those reports have been merged with the pot material.
- The lid itself, which provides access for searching and reusing material out of the pot. The lid is only lifted when experience engineering adds refined ingredients to the pot, or when someone wants to pick a document mentioned in the lid. For activities that are over, this restriction is more protection than annoyance.
- Essential operations around the base are: *Input* into the input box; *Experience Engineering* boils down new material to things that relate to the pot; *Searches* are carried out on the lid and lead to *Output* of existing material from the pot.

4.2 The Bigger Picture: LIDs in Process Perspective

Experience elicitation according to the LIDs approach follows a process outlined in Figure 4. Some of the process steps are original tasks of the activity itself (gray), whereas others are carried out only when systematic elicitation and reuse of experiences is taking place (white).

LIDs themselves occur only in step six of this process; all previous steps prepare for it. This process is again the ideal case of a LIDs experience exploitation process. This process explicitly includes (re-)using existing LIDs – directly or in the form of best practices.

Some steps may be omitted: Often, there will be no official evaluation of any kind (e.g. post-mortem; step 5); best practice and shopping window presentation (steps 7, 8) are results of more intense experience engineering. They are crucial to fully exploit experiences from LIDs, but they will occur only rarely, and off-line.

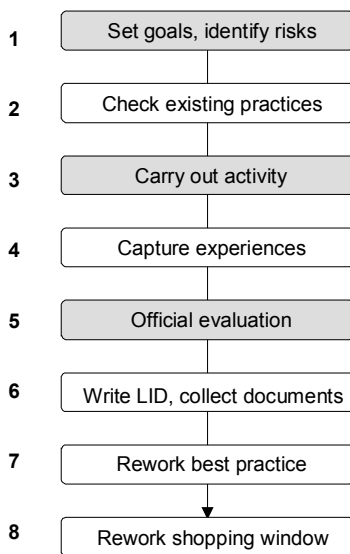


Fig. 4. Experience exploitation using LIDs

Essential steps are capturing experiences (in a light-weight way), writing a lid and collecting documents, and checking existing best practices and lids for reusable stuff. Checking existing practices is crucial! It offers benefit. All effort will be spent in vain if the existing material is not reused. Lids are supposed to make reuse easier.

- Steps 1 and 2 occur before the activity takes place. It is good practice to think about goals and risks, anyway. As a side-effect, they also provide highly effective handles for later experience elicitation.
- Step 4 refers to an immediate opinion poll right after the activity. This may be carried out in a short session one week after everything is over. Before everybody deletes the files and cleans up their desks, the group should spend an hour to collect and document the most prominent observations, experiences, warnings and opinions. A relaxed atmosphere is helpful.
- Step 5, if carried out at all, is more systematic and takes more time and effort than the opinion poll should. If there is any formal evaluation (e.g., based on a questionnaire), it is an excellent source of experience. All results go into the experience base *Workshop* layer of Figure 2.

- In step 6, the core of the LIDs approach is reached. The section below explains the checklist that can be used as guidance in this step. This step is usually fun for people. Results are always helpful for *project supporters*; some LIDs may be appropriate for *long-standing customers* (Figure 2).
- Steps 7 and 8 are follow-up steps that distill the results of the lid-and-pot step 6 into material that is more directly reusable. Up to step 6, every information captured refers to what happened in the activity, how, why, and when. At step 7, there are conclusions drawn to describe what *should* be done in similar situations in the future. A rule of thumb makes things much easier again: if the activity was not a failure, and if there is no other "good practice" available, just use the observed practice as a "best-so-far practice". Of course, modifications due to identified weaknesses must be made. The result is a Best Practice document for *customers*, sometimes for the *self-service shelf* of Figure 2. Step 8 is rare: only fundamental changes ripple up to the shopping window. The surface level is rather stable.

Non-trivial experience engineering for a best practice starts when there are several lids around one topic. As Figure 5 indicates, best practices will then try to take all those lids into account. This is how a good practice actually grows into a best practice. Once lids are engineered into best practice, they are still needed as real-world examples to illustrate that best practice. An example from their own environment is an element many people miss in general process manuals.

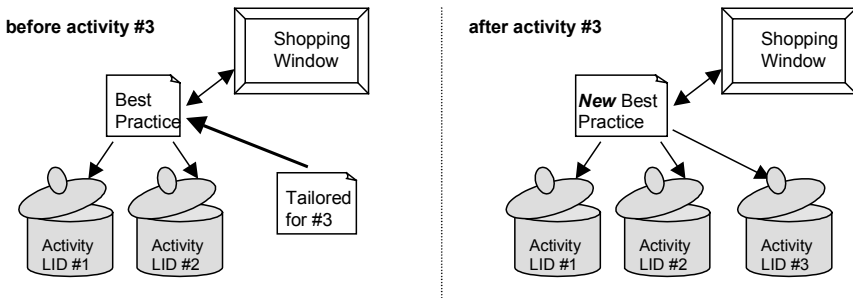


Fig. 5. Using lids and pots during an activity (#3). Arrows indicate access links

As the above discussion of steps and their results has shown, user perspectives in Figure 2 are supported bottom-up: First, project supporters - naive users last. It is important to stress that it will in many cases be sufficient to support only project supporters. LIDs is all they need. Everything else is welcome, but not mandatory.

4.3 Contents and Structure of a LID

It has been most useful to start writing a LID from a pre-existing LID. We use the example of case A which is our best example so far. It contains the structure shown in Figure 6. In the actual LID writing session, participants will go through the document and fill in their story. The existing example helps them to adjust their style,

granularity, and amount of information given. If there are more than three people involved (as is often the case), a computer projector is the best way to write the LID in a team effort. Document names mentioned in the story are underlined first. They can be linked off-line (e.g., via the MS Word or HTML linking mechanisms).

Standard structure of a LID: to be used as checklist and as table of contents	
1	What was the reason, trigger, and the responsible customer or initiator for this activity?
2	Do activities like this occur repeatedly, or even on a regular basis?
3	In general: was it a success?
4	Links to goals, expectations, and risks
5	As-is process ("the story"): how did the activity unfold?
5.1	Overview: Phases, Results, People
5.2	Preparation
5.x	...
5.n	Follow-up activities
6	In hindsight: observations and expectations
6.x	With respect to the above phases or subactivities, or results
	...
7	Conclusions: „From what we have seen here, we will recommend ...“
8	Reusable documents (e.g., templates)
<i>(extensions and modifications are possible; this is a guideline only)</i>	

Fig. 6. What should go into a LID? A checklist

Section 1 is less than half a page long. It provides the context, stakeholders and motivation. A reader will be able to tell right away whether the following could be applicable in his or her case [14]. **Section 2** is again less than half a page long. It asks the crucial question: is it worth while writing a LID? If there is little chance to ever reuse the story, why would you record it? We have never seen a team stopping at this point, but they often reframe their goals and reduce their effort. Asking for the general impression (**section 3**) before diving into details gives both authors and readers a chance to calibrate their expectations. A success story is told and read very differently from a failure report. Both reports are highly regarded. However, readers should be aware of the authors' mood. Most real contents are described in **sections 4 and 5**. First, authors are reminded of the goals and risks. *Sections 1-4 together are usually under two pages long.*

Section 5 is the center piece, and should be around 3-5 pages long. It usually contains a graphical or textual description of what happened (as-is process). Subsections are devoted to the steps or details of this process. This is also the section

with usually more than 80% of all links to documents. **Section 6** uses the observations and experiences captured immediately after the activity, and relates them to goals, expectations, risks - and to the story. The conclusions in **section 7** are the attempt to capture what many people are eager to convey: "Next time, I would ...": their conclusions, tips, and tricks. **Section 8** is just for convenience, to put all templates together for the impatient reader.

5 Applying LIDs in Practice

LIDs have been applied in a number of practical situations. Two of them are examined to provide an impression on the kind of information captured, the amount of effort spent, and the results achieved.

5.1 Case A: A Community-Building Workshop

Activity and challenge. In the first case examined here, SEC had organized a very special workshop for the business units. It was about a technical discipline that is becoming more and more important in high-end cars; and it has a lot to do with software. Before the workshop, several business units had started working in this direction, but had not met. The main purpose of this one-day workshop was to bring key people together, focus their attention on topics of common interest, and to offer initiating a company-wide community of practice.

There were several challenges involved, like identifying the key people from several business units; scheduling the workshop on a day and in a place that would allow most of them to participate; find interesting external and internal speakers who could give a "keynote" or start the discussion by providing concrete insights into their business units. SEC ended up giving the keynote, which again required substantial effort to understand enough about this new technology to be able to gain participants' interest.

Applicability of LIDs. From the decision to organize this workshop to the wrap-up immediately afterwards, it took us about three months of (part-time) work. Learning from others' experiences requires a certain kind of community. Therefore, a workshop like this can be a good (although comparably heavy-weight, high-effort) tool to support our other SEC activities. Potential for reusability and the conditions ideally fit the LIDs application conditions. We followed all steps of the Figure 4 process down to writing LIDs. A best practice has also been derived. In the meantime, a similar workshop for a different community was held. The LID produced in the first workshop was used, lessons learned in the second workshop were recorded.

LIDs effort and benefits. All steps before actually writing the LID were so much intertwined with the activities to organize, moderate and evaluate the workshop that they were not perceived as extra effort. Then, one person prepared a raw sketch of the story. In a two-hour session of seven people it was filled and turned into almost the final LID. Existing documents were identified in the session. Afterwards, they were collected and linked, which took half a day more.

As an unexpected by-product, participants were very much in fond of writing a LID as a wrap-up. They kind of "discharged" that way. They provided the documents, cleaned their directories, and had the impression that the lessons they had learned (sometimes the hard way) had been saved. Their struggles had not been in vain. LIDs had provided a psychological benefit even before its result had ever been used. There is yet not quantitative basis for measuring benefit. Qualitatively, reuse has occurred and is likely to occur on a regular basis.

Effort 19 hours	Steps before writing LIDs	0 extra
	Raw sketch of the story	1 Person * 2 hours
	LIDs writing session	7 people * 2 hours
	Collecting and linking documents	3 person hours
Benefit	Participants liked LID wrap-up: saved what they had learned	motivation up
	Eight-page LID with over a dozen documents attached, 5 reusable	reused in later workshop

5.2 Case B: Introducing Risk Management

Activity and challenge. In the second case from a business unit, the "activity" was our attempt to introduce risk management. The mission was to get all participants informed and motivated; to provide them with the templates they could use for carrying out day-to-day risk management tasks; and to coach them to the point where a continuous loop of *risk identification - mitigation - control and replanning* was reached. This activity took a little longer than three months. Three presentations were given to different groups of stakeholders. A risk questionnaire was tailored from a template, and was used to elicit initial risks. Results were evaluated, prioritized in the respective project group, and risk mitigation activities were defined. The interaction of all stakeholders with the risk manager for the continuous phase was defined and explained. After that we withdrew from the most active part and offered on-going exchange of experiences.

Applicability of LIDs. Again, this kind of activity is very experience-relevant: Introducing risk management is in high demand in countless projects and business units. As we have learned, most of the effort needs to go into good tailoring of standard processes (whereas *providing* a standard process can easily be delegated to a best practice manual). In order to become more efficient, we need to constantly improve our tailoring efforts, based on experiences in similar projects or business units. Unlike case A, risk management introduction was not a stand-alone activity. It was embedded into a larger effort, continuous risk management, which again was only part of project management. It were also not so much our own experiences that had to be captured, but those of several other stakeholders (project leader, participants, etc.) However, the interface of our activity to the rest of the project was sufficiently clear so that also the LID could be written without too much forward and backward references.

LIDs effort and benefits. The entire LIDs process (Figure 4) was followed. Since we produced and used several templates during the activity, there is now a nice set of

templates together with a *use case* for the templates: the story in the LID. Since stakeholders were not co-located, it took us three short occasions to elicit "stories". We did that during regular meetings, so it was just perceived as an agenda item, not as an extra session. Currently, there is some more effort invested to use the LID together with other experience reports (from the time when LIDs had not been "invented") to bring up a first practice description. This resembles very much the situation in Figure 5. The integration of those new risk elements and the writing of a Management Summary has been made much easier, but savings are difficult to quantify since comparable tasks did not occur before.

Effort 17 person hours	Steps before writing LIDs	0 extra
	Raw sketch of the story	1 Person * 4 hours
	LIDs writing sessions	3 occ. *3 people* 1 hour
	Collecting and linking documents	4 person hours
Benefit	Best practice with templates and use case (example)	ready for reuse
	Easy to integrate new risk-management elements (defined proc.)	Done within a week
	Management summary on current practice found all material ready	Within 4 days

6 Conclusions

There are several heavy-weight approaches to experience elicitation and reuse [4, 9, 11, 17]. Despite their interesting benefits, they will fail in many situations, and in many environments. When people are not willing or not able to invest the amount of effort required, heavy-weight approaches never get a chance to provide the expected benefits.

We have tried to focus on the most highly regarded experiences that can be easily elicited after an experience-loaded activity. The LIDs approach is supposed to be as much in line with the usual procedures as possible, and to require only very little extra effort. A simple process, a checklist and the background information in this paper support LIDs. Technically, it requires not more than the Microsoft-Word Hyperlink mechanism, or an equivalent linking tool (e.g., based on HTML). Since rather large amounts of information and documents are linked in an intrinsically meaningful way (along a real story), there is little need for extensive search mechanisms. This makes *using* LIDs easier. In most cases, lids will end up as illustrating examples to a best practice. Deriving the best practice is supported by LIDs, but there is still the need for analytical and creative thinking.

LIDs is not the one and only solution for experience elicitation. It is tuned for a specific situation (see section 4.1) that also determines the granularity and the approximate size of results. We have encountered several situations in practice that were not quite the ideal case for LIDs. In some cases, it may be worthwhile stretching

the concepts a little. However, if basic preconditions do not hold, LIDs reaches its limits (e.g., if no one remembers the situation well; if the focus is not an activity, but an organizational unit; or if the assumed “experience” is too shallow or too tacit). One should then consider alternative elicitation techniques – and stay aware that *no* technique can elicit deep insights from shallow opinions.

Since we take the LIDs approach seriously as a pragmatic way of experience engineering, its application in a number of cases has been highly encouraging. This gives reason to hope it could also be applicable in other companies, where experience-based process improvement sometimes needs to start small and experience reuse has to be a light-weight add-on to regular activities. The LIDs approach is one more example of making experiential learning work in practice. It can be complemented and combined with other (e.g., heavy-weight) approaches, it is particularly well-suited during the crucial starting phase of experience-based process improvement, and it maintains all options for more sophisticated experience engineering, storing, or searching for reuse.

References

1. V. Basili, G. Caldiera, and D. H. Rombach. *The Experience Factory, Encyclopedia of Software Engineering*. John Wiley and Sons, 1994.
2. Basili, V.; Caldiera, G.; McGarry, F.; Pajersky, R.; Page, G.; Waligora, S. (1992): The Software Engineering Laboratory – an operational Software Experience Factory. In *Proceedings of the 14th International Conference on Software Engineering (ICSE-14)*, pp 370-381.
3. V. Basili, G. Caldiera, and H. Rombach (1994a): Goal question metric paradigm. In J. J. Marciniak, editor. *Encyclopedia of Software Engineering, volume 1*. John Wiley & Sons, New York, 1994. pp 528-532.
4. Fensel, D.; Decker, S.; Erdmann, M.; Studer, R. (1998): Ontobroker or How to enable intelligent access to the WWW. In Proc. *11th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98)*, Banff, Canada.
5. Flor, Th.: *An Adaptive Information System for Cooperative Learning Environments*. Habilitation thesis, University of Ilmenau, Germany (in German), 1999.
6. Grudin, J.: Social evaluation of the user interface: Who does the work and who gets the benefit. Proc. of INTERACT'87. *IFIP Conference on Human Computer Interaction*. Stuttgart, Germany, 1987.
7. Grudin, J.: Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. Proc of the Conference on Computer-Supported Cooperative Work (CSCW 88), 1988.
8. Johannson, C.; Hall, P.; Coquard, M.: Talk to Paula and Peter - They are Experienced. Proceedings of the Workshop on Learning Software Organizations. Kaiserslautern, June 16, 1999.

9. Houdek, F.; Kempter, H. (1997): Quality patterns – An approach to packaging software engineering experience. In: Harandi, M. (ed.): Proceedings of the Symposium of Software Reusability (SSR'97), *Software Engineering Notes* 22(3), pp. 81-88.
10. Houdek, F; Schneider, K: Software Experience Center. The Evolution of the Experience Factory Concept. *International NASA-SEL Workshop, Dec. 1999*.
11. Houdek, Frank; Schneider, Kurt; Wieser, Eva: Establishing Experience Factories at Daimler-Benz. *Proc. of the International Conference on Software Engineering (ICSE-20)*, Kyoto, April 19-25, 1998.
12. Lindstaedt, S. *Group memories: A knowledge medium for communities of interest*, Ph.D. Dissertation, University of Colorado, Boulder, 1998
13. McMenamin, S.; Palmer, J. F.: *Essential Systems Analysis*. Yourdon Press, Prentice Hall, Englewood Cliffs, 1984.
14. Minto, B. (1987): *The Pyramid Principle - Logic in Writing and Thinking*. Minto International, London, 3rd edition.
15. Moran, T.; Carroll, J. (1996): *Design Rationale: Concepts, Techniques, and Use*. Lawrence Erlbaum and Associates, Mahwah, NJ.
16. Nonaka, I.; Hirota, T.: *The Knowledge-Creating Company*. Oxford University Press, 1995.
17. Pirle, T.; Studer, R. (1995): An environment for reusing ontologies within a knowledge engineering approach. In *International Journal of Human Computer Studies* 43(5/6), pp. 945-965.
18. Polanyi, M. *The tacit dimension*. Doubleday, Garden City, New York, 1966.
19. Schneider, K.: Prototypes as assets, not Toys. Why and How to extract Knowledge from Prototypes. Proceedings of the 18th *International Conference on Software Engineering (ICSE-18)*, Berlin, 1996.
20. Stolze, M. Visual critiquing in domain oriented design environments: Showing the right thing at the right place. In Gero, J.S. and Sudweeks, F. (eds.): *Artificial Intelligence in Design '94*; Kluwer Academic Publishers, pp. 467-482, 1994.