

A Light-Weight Process for Capturing and Evolving Defect Reduction Experience

Victor R. Basili
*Fraunhofer Center for
Experimental Software
Engineering, Maryland*
vbasili@fc-md.umd.edu

Mikael Lindvall
*Fraunhofer Center for
Experimental Software
Engineering, Maryland*
mlindvall@fc-md.umd.edu

Forrest Shull
*Fraunhofer Center for
Experimental Software
Engineering, Maryland*
fshull@fc-md.umd.edu

Abstract

Selecting technologies for developing software is a crucial activity in software projects. Defect reduction is an example of an area in which software developers have to decide what technologies to use. CeBASE is a NSF funded project that has the role of improving software development by providing decision support on the selection of techniques and tools. The decision support is based on empirical data organized in experience bases and refined into high-level models. Empirical data is collected through various activities, for example through eWorkshops in which experts discuss important issues, and formalized using the lightweight knowledge dust to knowledge pearl process.

1. The Need for Capturing and Evolving Defect Reduction Data and Experience

When software defects (i.e. faults in the requirements, design, or code of a software system) are allowed to propagate to subsequent development phases, the effort necessary for detecting and correcting them tends to increase. In the worst case, the defects are never caught and result in faults in the delivered product. At the very least, uncaught defects increase the cost of software by causing wasted effort in development, or just through the time required for tracking down the bug and removing it. Thus, an effective approach to software defect management is needed to produce software of desired quality, within time and budget constraints.

To create a coherent defect management approach, software developers have a plethora of development tools, techniques, and methods from which to choose. In order to develop software on-time and within-budget, meeting the correct functional and non-functional requirements, they need to make informed decisions about which of these technologies to select. Such decisions should be

based on an understanding of the effects of those technologies on software quality (both alone and in combination with other development technologies) and their associated cost. Example questions are:

- What types of inspections are best for catching defects of omission in the requirements analysis phase?
- For reviewers at an average level of experience, is a procedural or non-procedural approach to code reviewing more effective?

Too often, such decisions are based on anecdote, hearsay, and hype. Many software developers are still surprised to learn that 25 years of empirical research activities in software engineering have not yielded answers to such questions. One problem is that the results of such activities are hard to summarize in an actionable way. Results come from different environments; often introduce subtle differences in the way the same technology is used; are reported in different ways, in different publications.

Abstracting a coherent and useful story from so many independent data points is not easy, but it is necessary if our understanding of the essential phenomena of the field are to progress and are to be continuously tested against common, real-world experience. This is necessary for ensuring the continued relevance of research results, and for getting the fruits of this research in a form where they can be useful for the people developing software.

For this reason, the United States' National Science Foundation funded the Center for Empirically Based Software Engineering (CeBASE) starting in 2000. CeBASE has, as part of its mandate, the role of improving software development by communicating to developers what heuristics and models exist to support decision-making based on empirical results; that is, on observation, experience, and measurement, not belief, theory, or hype. To support these goals, CeBASE has been experimenting with processes for abstracting practical lessons learned

across multiple sources of information about a technology, ranging from rigorously-controlled empirical data to results obtained under the vicissitudes of a live development project to time-tested, expert experience. CeBASE has researched methods for abstracting and modeling the needed information for decision support across multiple studies, and collaborates on further empirical studies where necessary to support that research.

2. The Need for a Lightweight Process

One of the cornerstones of CeBASE has been the building of experience bases in the various areas that are covered by the project. Experience bases are based on repository technologies in which experience and knowledge is stored, organized, and disseminated to users. Building an experience base faces the challenge of where to start and how to build and support its user community while there is a limited amount of content in the experience base. The general approach to building experience bases is to build and provide an infrastructure for sharing of information without emphasizing enough the question of what will drive people to use the experience base.

Experience bases are based on the contribution of experts. They are built on the fact that knowledgeable individuals capture and share their knowledge with other individuals. When the concept of sharing experience is presented to experts, the message is often interpreted as invest now, and *someone else* might harvest later. This is often not satisfactory enough to motivate these experts to share their knowledge. Experts need immediate gratification or return on their efforts in order to value experience sharing.

Another aspect is that because it takes time to receive the benefits from sharing experiences and because it is hard to measure these benefits, experience management can be seen as a risky activity. Experience management requires a relatively large investment and a fundamental commitment to change the organizational culture to a sharing one. The risk lies in the fact that it takes a long time to notice if the wrong approach was selected or to find out that another direction would have been more successful.

We have been experimenting with a lightweight process to building experience bases called the *Knowledge Dust to Pearls [1]* approach that addresses these problems and has shown promising results.

Based on our experience we set out to define a lighter and less risky approach that would be more appealing to both experts and novices. The new approach is influenced

by the ideas of the Quality Improvement Paradigm (QIP) [2]-- a model for process improvement in software organizations. QIP uses the notions of continuous improvement and iterations as the main vehicle for planning, executing, evaluating, and improving processes. These concepts led us to define an approach that lets organizations, including CeBASE, define and grow experience bases gradually and improve step by step. It allows organizations to invest less now and also harvest some now. It enables the organization to evaluate the approach, and improve based on the results. When the organization is ready to advance, it can invest more and harvest more. This leads to a situation where experts see benefits much sooner and allows the overall direction of the initiative to be adjusted quicker.

Our main approach is the Experience Factory (EF) [3], which establishes a learning organization. The EF approach is beneficial for software organizations that need to learn from their past experience. The EF is a sophisticated approach that satisfies an organization's long-term needs of sharing experience. We searched the literature for a complementary approach that would satisfy the short-term needs of an organization. The complementary approach we selected to base our new approach on was the AnswerGarden [4].

The AnswerGarden (AG) addresses two challenges; the first challenge is how to capture and share the experience so that known answers can be quickly dispatched to the ones who need them. The second challenge is to establish a process that allows experts to share their knowledge with each other and with novices in an efficient way.

Our approach, the Knowledge Dust to Pearls, combines and makes use of benefits both from the AnswerGarden (which represents Knowledge Dust) and the Experience Factory (which represents Knowledge Pearls). First, it serves short-term needs as it uses an ad-hoc methodology; and, it enables the collection of fine-granular items that lead to organic growth. Organic growth is a desirable property of an experience base as it lets the experience base grow in areas where employees search for experience. Second, it serves the long-term needs as it is based on a sophisticated analysis and synthesis methodology; it uses feedback loops; and recognizes the need of a separate organization that is responsible for the analysis and synthesis.

The new approach captures knowledge dust that experts use and exchange on a daily basis and immediately, with minimal modifications, makes it available throughout the organization. This process is accomplished by creating a system that supports peer-to-peer activities; i.e., the employees of the organization help

each other and fulfill the short-term return goals of a knowledge capturing and sharing approach. In parallel, the knowledge dust is analyzed and synthesized and transformed into knowledge pearls, which represent more sophisticated, refined and valuable knowledge items that take longer time to produce. This work is often complex and needs to be done by a separate organization: the EF group.

3. One Application: eWorkshops

An application of the concept of knowledge dust developed by CeBASE relies on electronic workshops or “eWorkshops” to gather, analyze, package and further test such information in a time-efficient manner [5]. EWorkshops are run over a standard web interface, allowing experts in geographically diverse locations to interact with one another for a common purpose in a relatively short time period. Actually, the name “eWorkshops” may be a bit misleading because although the online workshop is the centerpiece of experience collection, and the activity most visible to participants, there are associated processes and support roles in place “behind the scenes” regarding preparation, conducting, running, and analysis of the meeting. These processes and roles provide valuable focus and make sure the specific goals of the activity are addressed. This is an application of our approach in the following way: The experts find value in discussing important issues with their peers while at the same time their statements are automatically captured. These statements are the knowledge dust and are immediately useful for the participants of the eWorkshop as well as other people. Furthermore, the statements are analyzed in real-time resulting in a real-time summary of the meeting. A more extensive post-analysis is conducted after the meeting in order to turn the baby-pearls into real knowledge pearls. All of this material now constitutes the experience base with references to other published material as well as to experts in the field. The experience base is thus quickly populated with high-quality experience packages that attract experts to both share more experience and to use and learn from other peoples’ experience.

In order to achieve desirable results, the organization of the workshop follows a strict protocol:

1. *Choose a topic of discussion.* The topic under discussion is first determined.
2. *Invite participants.* Participants are invited and instructed to log into the eWorkshop using a Web browser at the appointed time.
3. *Distribute Pre-meeting information sheet.* To direct the discussion, preliminary information about the topic is presented to the participants, who send in pre-

meeting information. This is used to guide the discussion during the meeting. For the eWorkshops on defect reduction, for example, we used a “top 10” list describing 10 common beliefs about how defects behave, where they are found, and what works for removing them, to seed the discussion.

4. *Establish meeting codes – for meeting analysis.* The workshop organizers analyze the information sheets to develop a taxonomy of issues to be discussed.
5. *Publish synthesized info from pre-meeting sheets.* An analysis of the information sheets are given by the eWorkshop team and distributed to each participant before the meeting.
6. *Schedule pre-meeting training on tools.* A preliminary work session is scheduled to give meeting participants a chance to try out the software so that the meeting can proceed smoothly.
7. *Set up control room.* Several individuals (described later) actually run the meeting. While most participants are in their own offices looking at a computer screen, the meeting organizers need to coordinate their activities among several roles.
8. *Conduct meeting.* At the appointed time, the participants use their Web browser to log into the chat tool and the meeting is underway.
9. *Post-meeting analysis and synthesis.* A script of the meeting is kept and analyzed to extract knowledge for the knowledge base.

Roles for effectively achieving the above process include:

- *Lead discussants* interact with the international group of invited participant experts, to help direct the conversation.
- A *moderator* is responsible for focusing the discussion and maintaining the agenda.
- A *director* is responsible for assessing and setting the pace of the discussion.
- As the discussion evolve, a *scribe* capture and organize the results displayed on the whiteboard area of the screen. When participants reach a consensus on a particular item, the scribe updates the whiteboard to reflect the outcome.
- An *analyst* codes the responses according to a pre-defined taxonomy. The analyst enters one or more codes to categorize responses as they are entered.
- Tech support is responsible for handling any technical problems that might occur.

4. Results so far

Three eWorkshops have been run on defect reduction, specifically focused on heuristics for:

- Describing the cost and effort due to software defects;
- Describing the impact of defects on software;

- Evaluating effective methods for removing defects.

The general consensus has been that they have been not only enjoyable for participants, but also an effective way of quickly describing the state of knowledge of a field. For example, there were some mature technologies discussed that were clearly effective. Several participants described confirmatory evidence in the pre-meeting feedback and during the discussion concerning the effectiveness of software inspections. Although numbers varied, most sources reported that reviews caught more than half of a product's defects regardless of the project domain, level of maturity of the organization, or lifecycle phase during which they were applied. Several factors that could raise the defect detection rate even higher were mentioned. Given the consensus about review effectiveness, the discussion focused more on defining measures and refining the heuristic. Finally, some reasons were proposed to help understand why, in the face of so much data showing their effectiveness, peer reviews are not a more common practice in industrial software development. This collected information may be helpful for facilitating the dissemination of an effective practice.

For other, newer technologies, their degree of practical dissemination could be judged by the number of experiences cited for their use. Few participants could submit data to describe the effects of using disciplined personal practices, such as the Personal Software Process (PSP). Participants felt that the effectiveness of disciplined practices was related to a number of issues – defect introduction, removal, and cost-to-fix rates – across multiple stages of the lifecycle, and that without a framework to relate such numbers no global estimate of effectiveness could be reached.

Finally, the eWorkshops were perhaps most useful for describing heuristics about defect behavior that seem to hold across development domains. For example, several participants contributed data supporting the rule-of-thumb that about 80% of the defects come from 20% of a system's modules, although the exact relationship varies based on environmental characteristics such as development processes and quality goals. The implication this supports is that attempts to target the high-defect class of modules can have a very worthwhile payoff.

Full results of the eWorkshops, including the summarized expert discussion, are available online at <http://www.cebase.org/www/researchActivities/defectReduction/index.htm> as well as in [6]. (The process has since been applied to other areas, such as COTS-based development¹ and agile development methods², see [7].)

¹ <http://www.cebase.org/www/researchActivities/COTS/index.html>

References

- [1] Basili, V. R., Costa, P., Lindvall, M., de Mendonca Neto, M. G., Seaman, C., Tesoriero, R., and Zelkowitz, M. V. An Experience Management System for a Software Engineering Research Organization. The 26th Annual NASA Goddard Software Engineering Workshop. 2001.
- [2] McGarry, F., Pajerski, R., Page, G., Waligora, S., Basili, V. R., and Zelkowitz, M. V. Software Process Improvement in the NASA Software Engineering Laboratory. *CMU/SEI-95-TR-22*. Department of Computer Science, University of Maryland, College Park, MD 20742. 1994.
- [3] Basili, V. R., Caldiera, G., and Rombach, D. H., "The Experience Factory," *Encyclopedia of Software Engineering - 2 Volume Set*, pp. 469-476. 1994
- [4] Ackerman, M. S. Answer Garden: A tool for Growing Organisational Memory. Conference on Office Information Systems, COIS90, Cambridge, Mass. 1990.
- [5] Basili, V. R., Tesoriero, R., Costa, P., Lindvall, M., Rus, I., Shull, F., and Zelkowitz, M. V. Building an Experience Base for Software Engineering: A report on the first CeBASE eWorkshop. Bomarius, Frank and Komi-Sirviö, Seija. 110-125. Proceedings of PROFES 2001
- [6] Shull, F., Basili, V. R., Zelkowitz, M. V., Boehm, B., Brown, A. W., Costa, P., Lindvall, M., Port, D., Rus, I., and Tesoriero, R. What We Have Learned About Fighting Defects. International Software Metrics Symposium, Ottawa Canada. 2002.
- [7] Lindvall, M., Basili, V. R., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., and Zelkowitz, M. V. Empirical Findings in Agile Methods. 197-207. Springer. Extreme Programming and Agile Methods - XP/Agile Universe 2002.

² <http://fc-md.umd.edu/projects/agile/>