An Experience Management System for a Software Engineering Research Organization

Victor Basili, Patricia Costa, Mikael Lindvall, Manoel Mendonca, Carolyn Seaman, Roseanne Tesoriero, and Marvin Zelkowitz
Fraunhofer Center for Experimental Software Engineering - Maryland

Abstract

Most businesses rely on the fact that their employees possess relevant knowledge and that they can apply it to the task at hand. The problem is that this knowledge is not owned by the organization. It is owned and controlled by its employees. Maintaining an appropriate level of knowledge in the organization is a very important issue. It is, however, not an easy task for most organizations and it is particularly problematic for software organizations, which are human and knowledge intensive. Knowledge management is a relatively new area that has attempted to address these problems. This paper introduces an approach called the Knowledge Dust to Pearls approach. This approach addresses some of the issues with knowledge management by providing low-barrier mechanisms to "jump start" the experience base. This approach allows the experience base to become more useful more quickly than traditional approaches. This paper describes the approach and gives an example of its use at the Fraunhofer Center for Experimental Software Engineering – Maryland.

1. Introduction

The business world tends to be more and more knowledge-oriented. While organizations in the past were more focused on expensive machines to produce their products, organizations today face a much more competitive environment that is highly based on knowledge. In order to be successful and to excel, businesses of today need to focus on getting the right knowledge at the right time.

Most businesses rely on the fact that their employees possess relevant knowledge and that they can apply it to the task at hand. The problem is that this knowledge is not owned by the organization as such. The knowledge is owned and controlled by its employees. The following statements illustrate this situation:

- "Our knowledge has legs – it walks home everyday"[1]
- "Not only do you have experience walking out the door, you have inexperience walking in the door"[2]

Maintaining an appropriate level of knowledge in the organization is a very important issue. It is, however, not an easy task for most organizations and it is particularly problematic for software organizations, which are human and knowledge intensive. The employees are the main assets of software organizations. The software industry is characterized by frequent technology changes, which calls for a continuous stream of new knowledge. In addition, the turnover rate in software organizations is often high, which makes the problem of maintaining knowledge worse.

The problem of maintaining an appropriate level of knowledge consists of many different sub-problems. Examples of these sub-problems are:

- Loss of knowledge, for example when employees leave for other opportunities or for retirement.
- Lack of knowledge, for example when new people are hired they need to get up to speed by acquiring new knowledge, which often takes a long time.
- Lack of time to share knowledge, for example new experts need to share their knowledge with new employees, but seldom have time to do so properly due to a heavy work load. If they spend time sharing knowledge, they will be less productive.
- Location of knowledge is not known. A common problem is to locate knowledge needed to solve a particular problem. The knowledge is probably already present in many organizations; the problem is to identify where it is or who has it.

Knowledge Management (KM) is a relatively new area that addresses business problems by capturing and sharing

[1] Leif Edvinsson, Brain of the year '98, director of intellectual capital, Skandia Insurance Co.

[2] Scott Eliot, Director Knowledge Management Product Groups, Lotus (KMWorld 2001)

knowledge. There are many software packages on the market that help organizations with this task [6]. Knowledge Management can be applied by any kind of organization that needs to manage its knowledge. The Experience Factory [2] is an example of a knowledge management approach especially designed for software organizations. The Experience Factory recognizes that organizations need to learn from their past experiences in order to deliver products faster, cheaper, and with higher quality than before.

If knowledge management is the answer to the problems of maintaining knowledge, why aren't all organizations doing knowledge management?

The answer is that it is hard!

The payoff takes time and it is a hard sell.

When management is approached to invest in knowledge management they will ask the critical questions: (i) what is the return on the investment? and, (ii) what is the payoff time? The problem is that knowledge management is sold to management mainly as a long-term investment, meaning that the organization invests now and harvests later. Even if there are early benefits, these are often not evident because they are hard to see and measure. This is mainly the same problem that the software process improvement activities face: the benefits from these activities are often intangible and it is hard to assess.

The critical question that individual employees will ask is: what do I get out of changing my work processes and doing additional work? Knowledge management is based on the contribution of the employees. It is built on the fact that knowledgeable employees capture and share their knowledge with less knowledgeable employees. When knowledge management is sold to individual employees the message they hear is: You invest now, someone else might harvest later. This answer is often not satisfactory enough to motivate them.

Because it takes time to get the benefits from knowledge management and because it is hard to measure the benefits, knowledge management can be seen as a risky activity. Knowledge 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.

These risks manifest in many ways. A classical example is when the organization spends a long time developing an enhanced taxonomy before worrying about the content of the experience base. This can be compared to designing a library system for classifying books without having books to fill the library. When the organization realizes that content is needed it starts to author books for the electronic library. If it takes long to develop the taxonomy, it will take even longer to populate the experience base. Eventually, when the organization has spent all its efforts on authoring electronic books, the intended users have already lost interest because they perceive the experience base as limited in providing them the support they expected.

In our work at Fraunhofer Center for Experimental Software Engineering - Maryland (FC-MD) we have experienced similar problems when applying knowledge management to our own organization and to our clients. FC-MD is a not-for-profit applied research organization in the area of experimental software engineering. The center is based on the ideas of the Experience Factory [2]. The authors of this paper represent a team within FC-MD that specializes in the development of FC-MD's Experience Management System, Experience Bases, and Knowledge Management for Software Engineering in general. The work presented in the paper is based on experience from knowledge management and reports on our new approach: the *Knowledge Dust to Pearls* approach.

2. The Knowledge Dust and Pearls Approach

Based on our experience we set out to define a less risky approach that would be more appealing to both management and individual employees. The new approach is influenced by the ideas of the Quality Improvement Paradigm (QIP) [7]--- 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 leads the organization to knowledge management gradually and improve step by step. In short, it allows the organization 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 individual workers see benefits much sooner and allows the overall direction of the initiative to be adjusted quicker.

Our main approach is the Experience Factory (EF), 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 [1].

The AnswerGarden (AG) approach was developed due to the needs of sharing knowledge in customer support organizations and helpdesks. Such organizations receive requests for help from customers and users of a system and the organization helps them overcome the problems encountered with that system. The nature of this business is highly repetitive; the same request for help occurs frequently and the same help strategy can be used many times. The first challenge help-desk organizations face is how to capture and share the knowledge so that known answers can be quickly dispatched to customers. The second challenge is to establish a process that allows experts share their knowledge with each other and with novices in an efficient way. The AG approach lets employees store and organize questions and answers as they are received and answered by the organization. By storing questions and corresponding answers in a common repository, the knowledge can easily be spread throughout the organization. By making the repository open to customers, they can even try to serve themselves first, by searching for existing questions and answers. Only when they do not find an answer, they need to take one step further and contact the helpdesk. Examples of such approaches, including ones where customers help each other can be found on many companies' web sites[3].

Our approach, the Knowledge Dust to Pearls, combines and make use of benefits both from the AnswerGarden (which represents *Knowledge Dust*) and the Experience Factory (which represents *Knowledge Pearls*). The AG serves short-term needs; 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 knowledge, i.e. there is a need and demand for knowledge. The EF serves long-term needs, 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 EF group.

The new approach that we have implemented captures the knowledge dust that employees 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. The new Experience Factory Model

Figure 1 shows the Experience Factory (EF) Model and its main feedback loop. The EF group collects data produced by the Project Organization. The data goes through an extensive analysis phase and is synthesized into higher levels of knowledge and packaged in the form of experience packages. The experience packages are stored in the Experience Base and are made available to the project organization mainly in the form of business support provided by the EF group.

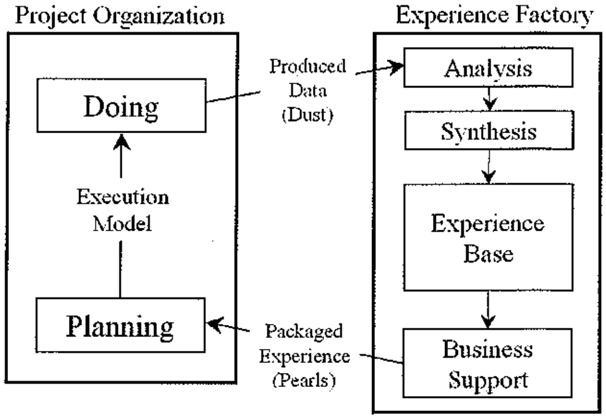

Figure 1 The Experience Factory Model.

The full feedback loop generates valuable experience packages, but due to the sophisticated analysis and synthesis processes, it takes a relatively long time to generate experience packages and make them available for the Project Organization. In order to deliver benefits from the EF sooner, we defined a way to speed up this feedback process.

The new approach adds a new and shorter feedback loop to the EF, which can be seen in Figure 2.

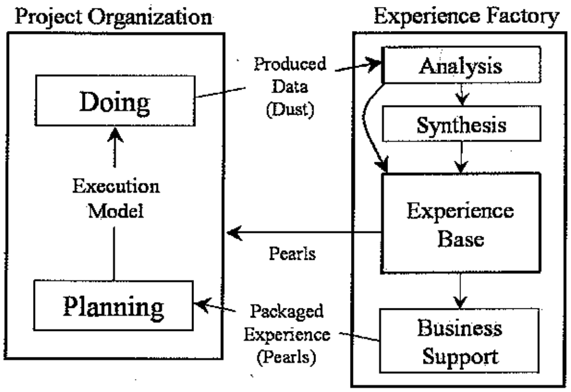

Figure 2 The Experience Factory Model with the new feedback loop.

[3] The support site for CodeCharge including FAQ's where the helpdesk helps customers: http://support.codecharge.com/. CodeCharge's community pages where customers can help each other: http://www.gotocode.com/community.asp

In the new approach the EF group defines a number of ways to let the project organization collect data about itself while it is conducting its core activities. This data, the dust, goes through a minimal analysis phase turning it into a mini-pearl. The mini-pearl entirely bypasses the synthesis phase and is stored in the experience base. The mini-pearls in the experience base are made available to the project organization almost immediately after collection. In this way, the organization receives benefits from the EF as soon as the dust collection process is established.

4. Knowledge Dust

The main characteristic of knowledge dust is that individuals produce it during their core activities. Thus knowledge dust is a nice side effect produced during activities that have to be done anyway. This does not mean that it takes no time to produce the dust or that it is worthless in another context. The point is that by collecting the information that is produced already and applying it to a broader context, widely extends its utilization. In essence, collecting dust requires that the organization asks employees for a minor change in behavior. An example of such a small behavior change is to capture tacit knowledge while they are being exchanged between two employees,. or by making explicit but private knowledge available to a larger group of employees. On other occasions, the organization must ask the employees for more information. An example is to ask for more information to be added to bug reports so that they can be more easily analyzed later. Knowledge Dust is much smaller and more fine-granular from a knowledge management perspective. Each dust particle may not convey a lot of information, but becomes useful when gathered, published and made searchable.

5. Knowledge Pearls

Knowledge Pearls are larger sets of dust that are analyzed and evolved into higher levels of knowledge. Knowledge pearls often start out as smaller pearls that grow larger and larger over time. Cultivating knowledge dust into pearls is a continuous improvement process. As more dust is collected, more extensive analysis can be conducted, which results in more complete and better-organized pearls. Cultivating knowledge pearls can be a relatively sophisticated activity that can take time. Therefore, it is conducted by the Experience Factory group which specializes in analysis and synthesis techniques, such as qualitative and quantitative analysis, statistical methods, and models based on empirical data. The pearls are fed back into the experience base for future use by both the project organization and the EF group. As the pearls are used by the organization, feedback about this usage is collected and analyzed and the pearls are improved based on this feedback.

Figure 3 shows the stages of the knowledge dust evolution until it turns into knowledge pearls. The cultivation of dust to pearls is very inexpensive early in the process. The project organization provides the dust with a minimal effort. The EF group spends minimal amount of time on the analysis of dust, turning it into mini pearls and then making them available to the project organization.

More sophisticated pearls require more sophisticated analysis. This more sophisticated analysis that is done by the EF group is more costly and takes more time, but the result is also much more useful to the organization.

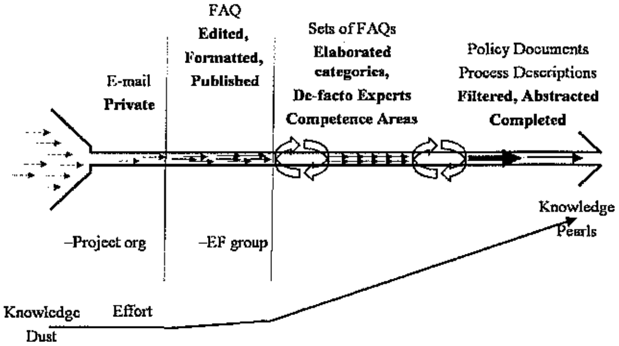

Figure 3 Cultivating Knowledge Dust into Knowledge Pearls.

6. Cultivating Dust into Pearls at FC-MD

The knowledge dust to pearls approach has been applied at FC-MD in several different ways, for various different projects. This section describes some of these activities and method for cultivating pearls.

Many times, employees communicate over e-mail when asking and answering questions. These questions and answers are stored in our FAQ system that can be used by all employees. The FAQ puts structure and makes public questions and answers exchanged by employees. Periodically, the EF group analyses the FAQs and identifies pieces that are rich enough to be turned into pearls. The related FAQs are put together in a more refined document, for instance, a process description that is stored in the experience base to be later used by the project organization.

When an incident occurs, employees are encouraged to record them in our Lessons Learned database. This database can be used by employees to search for such incidents and to apply the lessons learned to the task at hand. When the EF group realizes that a critical mass of lessons learned are collected on a certain topic, the EF group analyses and synthesizes the lessons learned into a best practices document that is stored into the experience base and is provided to the project organization.

We have a bug-tracking tool for documenting and tracking the defects during software development. The tool can be accessed by all employees and can help identify who are the experts on certain technologies. The bug-tracking tool provides fields where developers and designers document their decisions and issues, thus keeping a history of the facts. Based on the reports generated by the tool, it is possible to identify problem areas and proposed solutions. The EF group analyzes the bugs reported and establishes development practices that can be used by the project organization.

We are using a chat tool that for conducting guided discussions that are captured and analyzed. We already used the tool to conduct two eWorkshops for the CeBASE project on the area of defect reduction [3]. During an eWorkshop, a real time analyst classifies the statements to be analyzed later. There is also a scribe that generates summaries of the discussion, posts them in one of the chat tool's windows and immediately receives feedback from the participants. The chat logs and summaries can be used by the project organization immediately. Also, the EF group makes a detailed analysis about both the content of the discussion and the process to generate Best Practices and Lessons Learned documents that are stored in the experience base and can be used by the project organization.

One method that we are investigating for synthesizing knowledge dust into pearls involves the use of qualitative data analysis methods. Qualitative data is data in the form of text and pictures, as opposed to numbers or discrete categories, and can be either objective or subjective. Qualitative analysis techniques are generally designed to find trends, commonalities, and "stories" in textual data. Some of the more commonly used qualitative data collection techniques include interviewing and observation, while qualitative analysis techniques include the constant comparison method and cross-case analysis. The qualitative analysis method we have chosen to model our dust-to-pearls technique is the constant comparison method (first proposed by [4] and is described in more practical terms by [8]). The constant comparison method was designed to generate grounded theory, in other words to reveal hypotheses (based on trends, patterns, etc.) that are grounded in the qualitative data collected. The basic process is to first code a set of qualitative data (a textual narrative resulting from an interview, an observation, etc.) by assigning codes, or labels, to passages of text indicating the subject or theme of that passage. Then similarly coded passages are grouped together, in context, to reveal trends and patterns. A field memo is then written, describing the finding and referencing the data that supports it. Writing and refining field memos eventually leads to the formation of hypotheses. This process is repeated in parallel with continued data collection, thus refining and shaping the hypotheses as patterns in the data emerge.

Our process for synthesizing knowledge dust involves examining a set of experience packages for similar, contradictory, or related experiences, resulting in a comprehensive "story" or description of a phenomenon or a higher-level statement that is supported by experience. For example, suppose the experience repository contains 15 project history reports on COTS projects (i.e. development projects in which commercial components were integrated into the final system). Suppose further that eight of these cite problems working with "young" component vendors, e.g. the vendor going out of business, a vendor being unable to properly support the product, low quality components, etc. The vendors cited in these reports ranged in "age" from 6 months to 4 years. The remaining 7 projects, whose reports did not mention problems with vendors, worked exclusively with "older" vendors. Synthesizing the dust from this set of textual experience would result in a higher-level package that says, "Working with COTS vendors who have been in business less than 4 years is problematic" and then would go on to describe the types of problems that typically arise, discuss various contextual issues of these problems, and finally cite the original experience packages from which this information came. Such a higher-level package would be useful to future COTS projects not only because it constitutes an experience pearl, but also because it saves the future project personnel time in reading through all the previous project history reports (which they would be unlikely to do anyway).

The proposed process for qualitative analysis of experience dust is as follows. First, a set of experience packages (suspected of containing important experience dust) is identified. In the example above, this would be the set of project history reports. Then, a set of keywords is chosen that describe the topic or issue that needs to be investigated (in the above example, such keywords might be "vendor" or "problem"). Then the text in the experience packages would be searched for occurrences of the keywords and the passages found would be coded accordingly. The next step would be the intellectual task of grouping the coded passages and looking for trends, resulting in a "story", or a hypothesis, called a pearl. Finally, a new experience package would be created to summarize the newly created knowledge. Since this whole process would benefit greatly from tool support, our work to date has concentrated on identifying appropriate tools. The Visual Query Interface, described later, can be helpful in identifying and organizing the experience packages to be used as the basis of the synthesis. Tools like Nvivo™ (from QSR International; more information at http://www.qsr.com.au/products/nvivo.html) can be used to automate the searching and coding, as well as facilitate searching for trends and packaging the results.

7. Tool Support

The Knowledge Dust to Pearls approach is best-supported by software tools that establish a sense of low-barrier use and a modern user interface. All the applications mentioned above have the same basic architecture. The Experience Base is physically a database that stores experience packages. Figure 4 illustrates one example, the FAQ system. The other systems follow the same pattern.

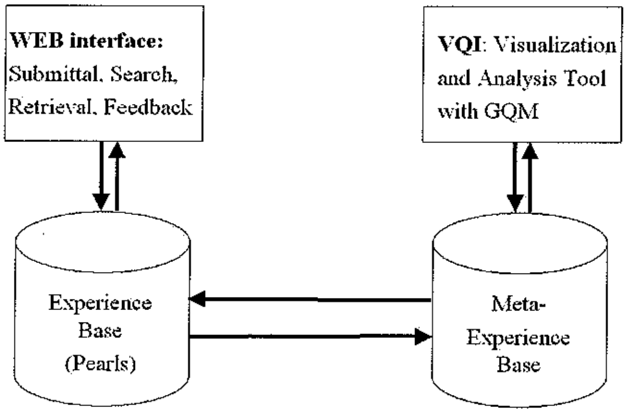

Figure 4 Tool Support.

In the FAQ system, the experience consists of frequently asked questions and answers.Dust is collected and immediately turned into mini-pearls. Integrated with the experience base that stores the knowledge dust is the Visual Query Interface (VQI), a software tool developed at FC-MD that is based on the ideas of the Starfield display [5]. VQI is used to analyze and draw conclusions about the content and growth of the experience base, feedback from users, and the usage of the tool. This analysis is used in the process of synthesizing the knowledge dust into checklists, best practices, guidelines, processes, and other forms of derived knowledge.

The Knowledge Dust Collector in the FAQ case is implemented as a web application with one part for end-users (the front-end) and one part for topic experts and content managers (the back-end). The front-end is a self-contained application, but it can be customized easily to fit into already existing applications for maximal utilization. The front-end is designed in such a way that it minimizes the learning curve for new users as much as possible. Following the principles of the Experience Factory and its theories for organizational learning, the front-end also allows the users to provide feedback that is used to improve the system as well as evolving the content. The back-end is used by content managers mainly. They manage the experience base, the network of experts, and the assignment of questions to be answered. In order to make it as easy as possible for topic experts, we implemented an email interface that enables experts to populate the experience base from within their email system. The most complex part of the system is the VQI, which allows for extensive analysis of the experience base. This Knowledge Dust Collector has been in use at FC-MD for more than six months and covers all major content areas found in such an environment. The experience base is used on a daily basis and contains (as of this writing) almost one hundred experience items supporting new employees as well as experienced ones. The feedback from users on the tool has been positive. One reason for the positive experiences is due to its way of capturing and sharing Knowledge Dust without interfering with the normal work processes.

8. Conclusions

Knowledge is increasingly becoming the main asset of many organizations. This is particularly true for software organizations. Knowledge Management is the proposed solution to the problem and is built on the capturing and sharing of employees' knowledge. Capturing and sharing knowledge is hard and risky. The time it takes to capture, organize, package and distribute knowledge is often to long for managers to be willing to invest in knowledge management. Employees hesitate to participate because they have to change their work processes and do additional work while their personal benefits are intangible and unclear.

We address these problems by defining a new approach called the Knowledge Dust to Pearls approach. This approach adds the short-term-oriented features from the AnswerGarden approach to the long-term and sophisticated features of the Experience Factory approach. The result is a new, shorter feedback loop that "jumpstarts" the creation and evolvement of Experience Factories in general, and Experience Bases in particular. The Knowledge Dust to Pearls approach is a web-based, low-barrier, and less-risky approach that helps organizations quickly capture, share and use daily knowledge, and evolve it into more complex knowledge over time. The approach gives the organization a starting point for building an Experience Factory and a Learning Organization. It helps change the corporate culture into a sharing one by making visible how sharing knowledge adds value sooner. The approach provides support for the project organization as well as for the experience factory organization. The project organization gets a system that enables them to share knowledge and help each other in a peer-to-peer manner. This peer-to-peer sharing is helpful for the EF group as well because it relieves them from the pressure to deliver immediately complex experience packages, which take time to develop. As time goes by, dust is collected in the experience base. This dust becomes the basis for the analysis and synthesis of the EF group, which turns the dust into pearls and delivers them to the project organization.

Acknowledgments

We would like to thank Ioana Rus for her helpful comments and Keith Blurton for supporting the project.

References

[1] Ackerman, M. S. "Answer Garden: A tool for Growing Organisational Memory", *In Proceedings of* Conference on Office Information Systems, COIS90, Cambridge, Mass, 1990.

[2] Basili, V. R., Caldiera, G., and Rombach, D. H., "The Experience Factory," *Encyclopedia of Software Engineering - 2 Volume Set,* pp. 469-476,1994.

[3] Basili, V. R., Lindvall, M., and Costa, P. "Implementing the Experience Factory concepts as a set of Experience Bases", Knowledge Systems Institute, *In Proceedings of* 13th International Conference on Software Engineering & Knowledge Engineering, pp. 102-109,2001.

[4] Glaser, B. G. a. A. L. S., *The Discovery of Grounded Theory: Strategies for Qualitative Research.,* 1967.

[5] Jog, N. S. B. "Starfield information visualization with interactive smooth zooming", *In Proceedings of* IFIP 2.6 Visual Databases Systems Proceedings, 1995.

[6] Lindvall, M., Rus, I., Jammalamadaka, R., and Thakker, R. "Software Tools for Knowledge Management
157", *In Proceedings of* DACS State-of-the-Art-Report, 2001.

[7] McGarry, F. and others, 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*, 94.

[8] Miles, M. B. a. A. M. H., *Qualitative Data Analysis: An Expanded Sourcebook.,* 2nd Edition ed. Sage Publications, 1994.