



A Review on Experience Management in Software Engineering

Navjot Kaur*
Department of Computer Science
Punjabi University Patiala, India

Navandeep Singh Tung
Assistant Professor, ME Department
Chandigarh University Gharuan, India

Gurjot Kaur
Assistant Professor, ECE Department
Guru Nanak Dev Engineering College
Ludhiana, India

Abstract: Experience Management (EM) is an area that is increasingly gaining importance. Its roots lie in experimental software engineering which is highly dynamic field in terms of research and knowledge. The goal of experience management is to reuse elements as building blocks in new software with modification occurring in a controlled way. It defines and develops methods, technologies and tools for identifying, collecting and storing experience related to software development. It helps in creating new software from existing software rather than building them from scratch. This paper highlights Experience Factory as an experience management approach in which experience is managed in a formalized way for the purpose of its reuse. It explicitly deals with continuous (organizational) learning from experience as a resource of increasing importance, merged under notion of Experience Management.

Keywords: Experience management, reusable elements, reuse approaches, experience factory

I. INTRODUCTION

Demand for new software applications is currently increasing at an exponential rate and so is the cost to develop them. However, the quality of software developed today is often low. One reason for this is the number of qualified and experienced professionals required for this extra work is not increasing commensurably. Another reason is complexity of software development tasks. With the advent of object oriented technology reusable software components now become an indispensable part of the knowledge required for software development. Experience Management (EM) is seen today as a feasible solution to this problem [1].

II. EXPERIENCE MANAGEMENT

Experience Management (EM) defines and develops methods for structuring and dealing with experience of experts for the development and advancement of its methods, tools and techniques. Software reuse has become a topic of much interest in the software community due to its potential benefits for increasing software productivity and decreasing its development cost [2]. Learning from experience gained in past projects is seen as a promising way to improve software quality at a lower cost in upcoming projects. But learning is not only limited to successful projects. Failures, especially, that occurred in past projects have to be carefully analyzed and documented in order to be avoided in the future [3]. The Software professional have recognized reuse as a powerful means to potentially overcome the situation called software crisis. Reuse of products, processes and other knowledge will be the key to enable the software industry to achieve the dramatic improvement in productivity and quality required to satisfy the anticipated growing demands [4].

Software development can often be a challenging field. It is a people- and knowledge-intensive activity; it is a rapidly changing field, and although it is slowly maturing, many activities are still ad hoc and depend upon personal experiences [5]. In order to cope with restrictions such as firm deadlines and shrinking budgets, software-developing organizations needs more reuse of software components which promises significant improvements not only in software productivity but also has a positive impact on the quality and maintainability of software product. It emphasizes strategies, techniques, and principles that enable developers to create new systems effectively using previously developed architectures and components. Number of organizational reuse approaches was introduced for the management of experience. These approaches explicitly deal with continuous (organizational) learning from experience as a resource of increasing importance, merged under the notion of experience management.

A. Experience factory:

Experience Factory (EF) is a logical and/or physical infrastructure to produce, store, and reuse experiences gained in a software development organization. It packages experience by building informal, formal or schematized, and productized models and measures of various software processes, products, and other forms of knowledge via people, documents, and automated support. It is substantial for the support of organizational learning, that the project organization and the learning organization are separated. EF is based on the Quality Improvement Paradigm [6] is the result of the application of the scientific method to the problem of software quality improvement. Experience Factory idea organizes a software development enterprise into two distinct organizations, each specializing in its own primary goals. The Project Organization focuses on delivering the software product and the Experience Factory focuses on learning from experience and improving software

development practice in the organization. Although the roles of the both are separate, they interact to support each other's objectives. The feedback between the two parts of the organization flows along well-defined channels for specific purposes. The Project Organization for each new project:

- The problem at hand is characterized.
- Goals are set.
- A suitable process is chosen.
- The process is executed and measured.
- Outputs are analyzed.
- Lessons and products are packaged and stored in the experience base for future reuse.

Experience Factories recognize that improving software processes and products requires:

- Continual accumulation of evaluated and synthesized experiences in experience packages.
- Storage of the experience packages in an integrated experience base accessible by different parts of the organization.
- Creation of perspectives by which different parts of the organization can look at the same experience base in different ways.

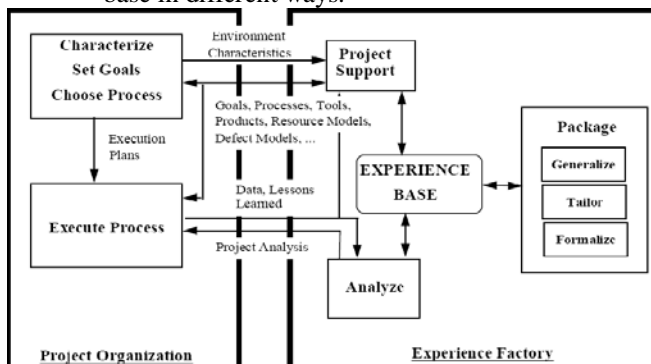


Figure 1. Experience Factory [7]

B. Experience packed for reuse:

- Process Models:** It is a description of a process being performed or to be performed. It is the base for process improvements. In the process model an organization can store its experience for how to best perform a task. As the process is being better understood and then improved. It includes
 - product models showing relations between the work products of the process,
 - ordered activities that manipulate the products,
 - roles that perform the activities,
 - methods explaining the technical and logical manipulations within the process,
 - examples and templates to speed up its use, and
 - training material to speed up the introduction the process users.
- Process Control Models:** It is a mechanism for controlling the performance of a process and thereby the result of the process. It is the quantitative way for enhancing the understanding of the process performance in a project. It must be analyzed and interpreted by the persons with the best understanding of the actual situation. These models are an important input to the development of reusable experience, indicating what, when and how to be reused.

- Project Experience:** Project Experience is all knowledge, both quantitative and qualitative, gained by measurements, interviews, feed-back sessions and analyses. Process experience consists of process control models with data, measurement database, feed-back reports, improvement recommendations, dependency graphs and project characteristics. It is a base for systematic learning. Learning is achieved when understanding a problem situation and then changing the behavior to avoid repeating problems that arises in similar situations, i.e. gain experience.
- Experience Packages:** It is a collected set of consistent process models, process control models and process experience that form an consistent unit. It is the main work product within the experience factory and it models the knowledge of the organization [7].

C. Experience Base:

The gained experiential knowledge is integrated into an information system that serves as a repository which allows comprehensive reuse and continuous improvement in Software Engineering practices. Such reuse repositories are referred to as Experience Base (EB). It is core of the experience factory which acts as an organizational memory [8]. The objective of experience base is to collect and enable access to a diversity of empirical data, process information, models, process knowledge and experience arising from software development process in highly accessible format. An effective experience base contains an accessible and integrated set of analyzed, synthesized, and packaged experience models that capture past experiences. In Experience factory concept, Knowledge gained is continuously analyzed and re-stored into the Experience Base. Once the Knowledge is stored it is called experience and covers models (such as process models, product models, quality models), instances (such as process traces, products, measurement data, techniques, tools), and qualitative experience (such as lessons learned). The Experience Base therefore has to be maintained regularly. In order to achieve this, Experience Base must also improve over time and continuously add value.

Experience Base acts as a Reuse Repository, which stores the artifacts and inter-relates them with further context information that is needed throughout improvement cycle. The structure of a repository is a key in obtaining good retrieval results. No matter how "intelligent" the matching algorithm, if components are indexed or otherwise structured poorly it will be difficult to achieve good retrieval performance. The intuitive and widely held assumption is that up-front investments in structuring a repository result in a proportional increase in the ease with which components can be reused [9].

D. Experience Base Architecture:

The Experience Base Architecture represents blue print of Experience Base. DISER method demands to set up a vision for the resulting knowledge management infrastructure. Given Fig. 2 presents the Experience Management Content Framework (EMCF). EMCF acts as a vision for a comprehensive management of experience within an organization.

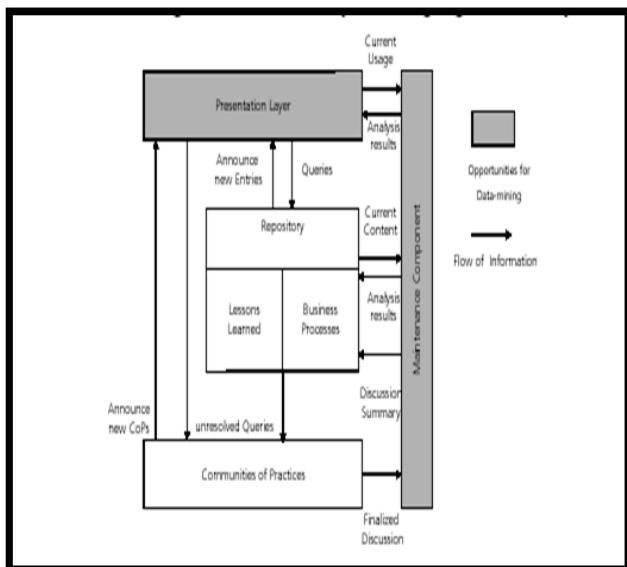


Figure 2. Architecture of Experience Base [10]

The EMCF consists of four basic components:

- a. Presentation Layer
 - b. Repository
 - c. Communities of Practices
 - d. Maintenance Component.
- a) The Presentation Layer is the interface of the EB to the regular user. It provides
 - (a) Uniform access to the information residing within the EB
 - (b) Stores the user preferences and settings
 - (c) Adapts and aggregates information within the EB based on those preferences.
 - b) The Repository contains the explicitly captured and consolidated experience of an organization. A combination of business process descriptions and lessons learned was chosen as a starting point. Further experience management activities can be set up on that base.
 - c) The Communities of Practice component is a forum for the member of an organization for discussion of current problems, questions and open issues.
 - d) The Maintenance Component supports the EF team in maintaining and developing the content of the EB (i.e., the data within the repository) and the services offered to the organization (via the Presentation Layer). This component offers the second place of a sensible application of data-mining methods: The content as well as the usage of the services can be analyzed to trigger, guide or otherwise support maintenance activities [10].

III. QUALITY IMPROVEMENT PARADIGM

It is based upon the notion that improving the software process and product requires the continual accumulation of evaluated experience in a form that can be effectively understood and modified into a repository of integrated experience models that can be accessed and modified to meet the needs of the current project. The paradigm implies the separation of project development from the systematic learning and packaging of reusable experiences.

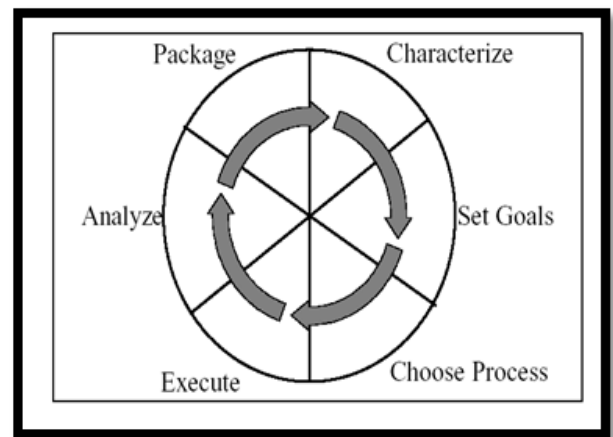


Figure 3. Quality Improvement Paradigm [11]

A. QIP consists of the following six steps:

- a. Characterize the current project and its environment with respect to existing models and metrics.
- b. Set quantifiable goals for successful project performance and improvement.
- c. Choose the appropriate measurement models and supporting methods and tools for this project.
- d. Execute the processes, construct the products, collect and validate the prescribed data, and analyze it to provide real-time feedback for corrective action.
- e. Analyze the data to evaluate the current practices, determine problems, record findings, and make recommendations for future projects.
- f. Package the experience as updated and refined models and other forms of structured knowledge gained from these and subsequent projects and save it in an experience base to be reused in future projects”[11].

IV. CONCLUSION

In this paper an approach for Experience Management has been presented. This focuses on how experience related to software development can be packed and shared across different organization or in a organization using Experience Factory infrastructure. Experience Factory supports learning and reuse and generates a tangible corporate asset in the form of packaged experiences. It help us to understand how software is built from scratch and where the problems are, define and formalize effective models of process and product, evaluate the process and product in the right context, predict and control process and product, qualities, package and reuse successful experiences and feedback experience to current and future projects. A real strength of this approach is it can start from small and expand with increase of knowledge and experience.

V. ACKNOWLEDGEMENT

Navjot Kaur would like to thank Mr. Neeraj Sharma, Associate Professor at department of computer science, Punjabi university, Patiala for helpful discussion and comments on the topic of this paper.

VI. REFERNCES

- [1]. Basilli V. and Caldiera G., "Improve Software Quality by reusing Knowledge and Experience", Sloan Management Review, 37(1), 1995, pp. 55-64.
- [2]. B.jalender, Dr. A Goverdhan, Dr.P Premchand "A pragmatic approach to software reuse" Journal of Theoretical and Applied Information Technology, 2010.
- [3]. R.L.Feldmann, "Developing a Repository Structure That Allows Comprehensive Reuse And Continuous Improvement", In Proceedings of the workshop on "Learning Software Organization," Kaiserslautern, June 16, 1999.
- [4]. V. R. Basili, H. D. Rombach, "Support for comprehensive reuse," IEE Software Engineering Journal, September 1991, pp. 303-316.
- [5]. Victor Basili, Roseanne Tesoriero, Patricia Costa, Mikael Lindvall, Ioana Rus, Forrest Shull, and Marvin Zelkowitz "Building an Experience Base for Software Engineering: A report on the first CeBASE eWorkshop" Published at PROFES 2001, pp 110 – 125.
- [6]. Basili, V.R., Caldiera, G., and Rombach, H., "The Experience Factory," In Marciniak, J. (ed.) Encyclopedia of Software Engineering, vol. 1, Chapter X, pp. 468–476, John Wiley & Sons, NJ, USA, 1994.
- [7]. PIA Experience Factory-The PEF Model, The PERFECT handbook [A booklet from ESPRIT Project 9090] Fraunhofer-IESE, Kaiserslautern, 1997.
- [8]. Klaus-Dieter Althoff, Markus Nick, Carsten Tautz "Concepts for Reuse in the Experience Factory and their implementation for CBR –System Development" Published in the Proceedings of the 11th German Workshop on Machine Learning, 1998.
- [9]. Raimund L. Feldmann, Jurgen Munch and Stefen Vorwieger "Towards Goal-Oriented Organizational Learning: Representing and Maintaining Knowledge in an Experience Base," pages 236–245, San Francisco Bay, CA, USA, June 1998 [10th International Conference on software Engineering and knowledge management].
- [10]. Klaus-Dieter Althoff, Bjorn Decker, Susanne Hartkopf, Andreas Jedlitschka, Markus Nick, Jorg Rech "Experience Management: The Fraunhofer IESE Experience Factory," Proceedings of Industrial Conference on Data Mining, Leipzig, July 24-25, 2001.
- [11]. Basili V. and McGarry F. "The Experience Factory: How to Build and Run One," 17th International Conference on Software Engineering, Seattle USA, 1995.