

# A Brief Review of an Experience Reuses Approach

Navjot Kaur<sup>#</sup>, Ramandeep Kaur<sup>\*</sup>

<sup>#</sup>CSE Department, Aryabhata Group of institutes,  
Barnala, India.

<sup>\*</sup>CSE Department, CT IEMT  
Jalandhar, India.

**Abstract**— Today, no corporate can do business without software. So, the demand for new software applications at low cost within short time period is currently increasing at an exponential rate which requires a substantial change to software industries. However, conventional methodologies have not achieved such drastic gain of the productivity and quality yet. Reuse of products, processes and experience originating from the system life cycle is seen today as a feasible solution to the problem of developing higher quality systems at a lower cost and in shorten time. This review paper highlights such a technical and social experience management infrastructure, called the experience factory, aimed at reuse of software development life cycle experiences, processes and products.

**Keywords**— *experience management, experience factory, methodologies, products, process, system life cycle.*

## I. INTRODUCTION

Almost any business today involves the development or use of software. It is either the main aspect of the business, the value added to the product, or on the critical path to project success. It permeates every aspect of life. If an organization is not investing heavily in the basic aspects of the software business then it will not be competitive and may not be in the business in the future [1].

Software development 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. For obvious reasons, learning from experience needs to be a permanent endeavor. Thus, an organization has to handle a “continuous stream of experience”[3]. To develop software for computers is a complex set of tasks. It involves several scientific disciplines, like understanding the needs of other people, and technical issues like transferring requirements into a computer program. This is supported by reuse and experience repository systems that assist in capturing and reusing all kinds of software artifacts (e. g., code, patterns, frameworks) and processes as well as experiences related to these artifacts and processes [4]. To provide physical storage and retrieval mechanism for reusable assets, and to provide simple mechanisms to support the reuse culture in the software company a technical and social infrastructure, called the experience factory was introduced.

## II. EXPERIENCE FACTORY

Reusing and handling experiences in companies are often referred to as knowledge management. One way to organize this is by giving the responsibility for capturing and reusing

experiences to a separate part of the development organization [6].

The Experience Factory is a logical and/or physical organization that supports project developments by analyzing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand (Figure 1).

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. This principal implies the logical separation of project development (performed by project organization) from the systematic learning and packaging of reusable experiences (performed by experience factory). It gives support for an organization to implement continuous improvement using experience and goal oriented measurements.

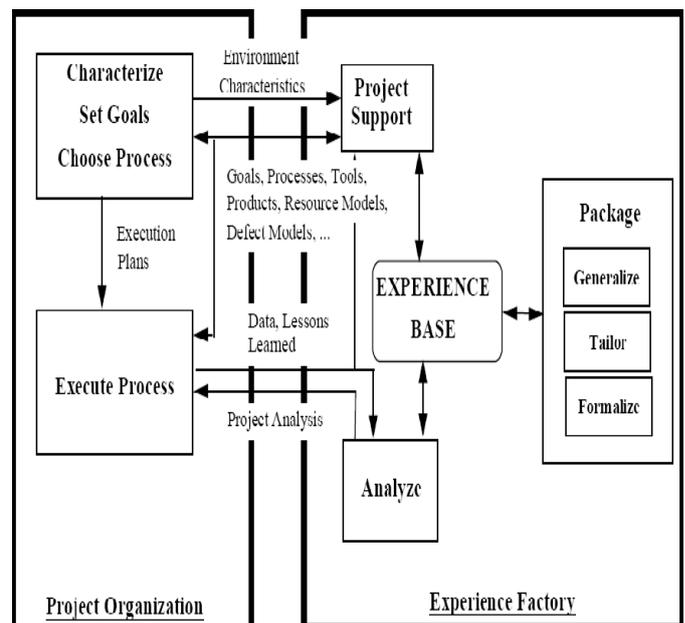


Fig.1. Experience Factory [2]

EF is based on the Quality Improvement Paradigm [2] 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 experience factory is a logical and physical organization, and its activities are independent from the ones of the development organization. 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 activities of the development organization and of the experience factory can be outlined in the following way:

- The development organization, whose mission is to develop and deliver systems, provides the experience factory with product development and environment characteristics, data, and a diversity of models (resources, quality, product, process) currently used by the projects in order to deliver their capabilities.
- The experience factory, through processing this information and other state-of-the-practice notions, will return direct feedback to each project activity, together with goals and models tailored from previous project increments. It will also produce, store and provide upon request baselines, tools, lessons learned, data, all presented from a more generalized perspective [1].

Experience Factories recognize that improving software processes and products requires:

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

#### A. Principal behind Experience Factory approach

It is very important to discuss the principal behind Experience Factory in the term of three-layer architecture which is essential for achieving following requirements:-

1. The system shall support geographical distributed organization allowing them to share and manage experience packages remotely.
2. The repository shall be robust, reliable and portable to standard computer platform.
3. The user interface layer/ Presentation layer shall be as platform independent as possible.
4. The data model shall be simple but powerful enough to model diverse class of "experience packages".
5. The software should be self easy to learn and self explanatory.
6. The stored information shall be easy to search and retrieve from repository.

#### B. How to build and run an Experience Factory

To start an EF there are two possible approaches: a top-down or a bottom-up approach. That is proceeding from defining processes, structures, products, and responsibilities to collecting concrete experience data, or else collecting data and

preceding back up a similar hierarchy. The authors of [7] propagate a top-down approach, which aims to define and establish the required elements before the improvement activities and the data collection takes place. This provides a guiding, and more or less stable structure and the time to focus on analysis of results and products rather than on integrating changes in the structure while working with them. Five key steps characterize the described top-down approach: (1) Obtain commitment, (2) Establish structure (3) Establish processes (4) Produce baseline (5) Identify potential changes.

### III. EXPERIENCE BASE

Experience Base (EB) is core of the experience factory which acts as an organizational memory. 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 analysed, synthesized, and packaged experience models that capture past experiences [1]. In Experience factory concept, Knowledge gained is continuously analysed 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. In experience base the experience is stored in the form of experience packages. It can be used to recognize similar situation, to avoid old mistakes and/or to know what has worked before.

#### A. Experience packed for reuse

1. *Process Models:* It is 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.
2. *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.

3. **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 [1].

**B. Experience package**

The main product of the experience factory is the Experience Package. The experience factory packages experiences with techniques, methods, and life cycle models by defining and refining models of their definitions and goals, understanding where they are appropriate and how they need to be tailored to a particular set of environmental characteristics. The content and the structure of an experience package vary based upon the kind of experience clustered in the package. There is a central element that determines what the package is: software development life cycle product or process, a mathematical relationship, an empirical or theoretical model, a data base, etc. The experience packages are defined by the life cycle product. Examples of experience packages are:

1. Product Packages have as their central element a life-cycle product, clustered with the information needed to reuse it and the lessons learned in reusing it. Examples: Programs, Architectures, Designs.
2. Process Packages have as their central element a life-cycle process, clustered with the information needed to execute it and the lessons learned in executing it. Examples: Process models, Methods.
3. Relationship Packages have as their central element a relationship or a system of relationships among observable characteristics of a software project. There are time based relationships and time independent relationships. In any case, these packages are used for analysis and/or forecast of relevant phenomena. Examples: Cost and Defect Models, Resource Models.
4. Tool Packages have as their central element a specific tool, either constructive (Examples: Code Generator, Configuration Management Tool) or analytic (Examples: Static Analyzer, Regression Tester)
5. Management Packages have as their central element any container of reference information for project management. Examples: Management Handbooks, Decision Support Models.
6. Data Packages have as their central element a collection of defined and validated data relevant for a software

project or for activities within it. Examples: Project databases, Quality records [1].

**IV. QUALITY IMPROVEMENT PARADIGM**

It is based upon 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.

QIP consists of the following six steps:

1. Characterize the current project and its environment with respect to existing models and metrics.
2. Set quantifiable goals for successful project performance and improvement.
3. Choose the appropriate measurement models and supporting methods and tools for this project.
4. Execute the processes, construct the products, collect and validate the prescribed data, and analyze it to provide real-time feedback for corrective action.
5. Analyze the data to evaluate the current practices, determine problems, record findings, and make recommendations for future projects.
6. 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”[7].

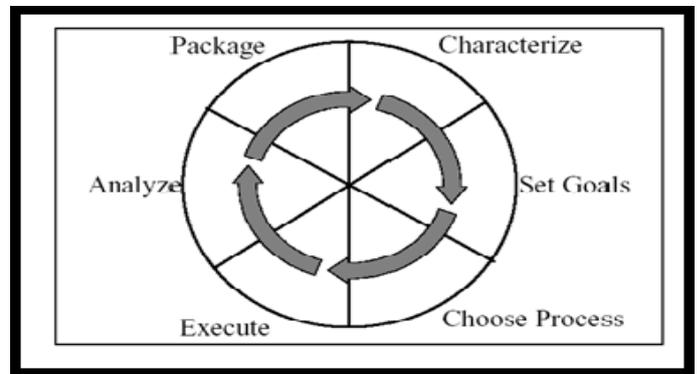


Fig.2. Quality Improvement Paradigm [7]

**ACKNOWLEDGMENT**

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.

**REFERENCES**

- [1].PIA Experience Factory-The PEF Model, The PERFECT handbook [A booklet from ESPRIT Project 9090] Fraunhofer-IESE, Kaiserslautern, 1997.
- [2] Basili, V.R., G. Caldiera, and H.D. Rombach, *The Experience Factory*, in *Encyclopedia of Software Engineering*, J.J. Marciniak, Editor. 1994, John Wiley. p. 469-476.
- [3] Klaus-Dieter Altoff, 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.

- [4] Markus Nick, Klaus-Dieter Althoff, Carsten Tautz “Systematic Maintenance of Corporate Experience Repositories” Computational Intelligence – Special Issue on Maintaining Case-Based Reasoning Systems.
- [5] Markus M. Nick, Raimund L. Feldmann “ Guidelines for Evaluation and improvement of Reuse and Experience repository systems through measurement programs” In Proceedings of the FESMA-AEMES Software Measurement Conference 2000.
- [6] Dingsøyr, Torgeir “An evaluation of Research on Experience Factory” Workshop on Learning Software Organisations at the international conference on Product-Focused Software Process Improvement, Oulo, Finland, 2000 Pages 55 – 66.
- [7] Basili V. and McGarry F. “The Experience Factory: How to Build and Run One,” 17<sup>th</sup> International Conference on Software Engineering, Seattle USA, 1995.