

# Chapter 68

## Software Engineering, Process Improvement, and Experience Management: Is the Nexus Productive? Clues from the Indian Giants

**Neeraj Sharma**  
*Punjabi University, India*

**Kawaljeet Singh**  
*Punjabi University, India*

**D.P. Goyal**  
*Management Development Institute, India*

### **ABSTRACT**

*Software engineers have always been experimenting with various models in an endeavor to improve upon software processes. Recently, a new area of software process improvement through experience management has got attention of the software engineering community. Software developers are experimenting with experience management to improve software process. This chapter explores the role of experience management in mitigating the effects of software crisis and investigates the current state of EM in Indian software engineering environment. The chapter discusses the need and integration of knowledge and experience management practices in software engineering environments. The chapter begins with the discussion of the ills of software engineering, classically referred to as “software crisis,” like late delivery of the systems, ill-functionality, and cost overruns, and justifies how managing the knowledge and experience of software engineers and practitioners can ward off these evils. The chapter then discusses the models and methodologies for knowledge and experience management in software engineering. Finally, the chapter sheds light on the major works carried out in this area and suggests possible future research avenues in this domain.*

DOI: 10.4018/978-1-4666-4301-7.ch068

## INTRODUCTION

Software has become ‘woven into the threads of our daily lives’ (Glass, 1996). It has become an inherent constituent of survival for almost all organisations as well as for individuals in today’s world of competition. May it be telecommunications, transportation, medical services or the defence of a nation, software is a critical component. With Internet and mobile technologies becoming omnipresent, and growing use of embedded software in consumer products, individuals are increasingly becoming dependent on software. For this ever increasing dependence over software, problems in developing software can have ravaging effects not only at individual and organisational levels but also at national and international levels. As a result, the improvement of software development processes has become the pressing area of concern for software engineering (SE) professionals and researchers.

Software engineering is a highly intricate, multi-stage process, usually involving teams of multi-domain experts and professionals working through different phases and activities, sometimes even beyond the boundaries of geographical space and time. Complexity in the field is further aggravated by continuous technology revisions and replacements. With the successful completion of every new project, new proportions in the SE knowledge are added. Even failures in the project execution lead to the generation of new experience that was not already known. Thus knowledge in SE is constantly growing, leading to another kind of problem – keeping track of what this knowledge is, where it is, or who owns it. Managing this knowledge in a systemic way and treating those who possess knowledge as valuable assets in an organisation, is the most potent solution to leverage the knowledge of individuals within the organisation.

Software engineering organisations qualify to be the perfect candidates for experience management initiative because of the knowledge-intensive

nature of SE tasks (Birk et al., 1999; Rus and Lindvall, 2002). Understanding and balancing user requirements, applying the most appropriate development methodology, choosing suitable tools, and storing project related experience are some of the examples which justify the knowledge-intensive nature of the software engineering discipline. Software development, like any other design work, is not rigid mechanical manufacturing process. Rather, it has a strong creative component involving human and social interaction that cannot be totally pre-planned (Conradi and Fuggetta, 2002). That is why we *develop* software and not *produce* it. Every person involved in software development has to make a variety of decisions, each of them with several possible options, in contrast to a production process which is mostly structured and programmable. Choice of what software to develop, selection of the team-members to carry out the project, selection of technique of programming, finalization of design from among many available alternatives, selection of kinds of tests and test cases for testing the final product, ways of providing user-manuals or help are some of the examples of the cross-roads which every software engineer faces. What helps software engineers make the ‘right choice’ when in situation of doubt? Well, most of the time it is their personal experience, gut feeling and intuitions that guide the software engineers. But for bigger projects, involving large group of people, knowledge sharing among individuals in the team is indispensable. And this is where experience management comes into picture.

The major objective of the chapter is to study the use of experience management (EM) systems in software engineering environments. It is especially interesting to study this in a setting with people who are very skilled in using computers, like people developing software. The chapter also investigates the current EM practices followed by Indian software engineering organisations. For the purpose of this study, an exhaustive literature search has been carried out. For the empirical evidence of EM in

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/software-engineering-process-improvement-experience/77763](http://www.igi-global.com/chapter/software-engineering-process-improvement-experience/77763)

## Related Content

---

### OPAC Usability Problems of Archives: A Case Study of the Hong Kong Film Archive

Ada Chi Wai Chung and Dickson K. W. Chiu (2016). *International Journal of Systems and Service-Oriented Engineering* (pp. 54-70).

[www.irma-international.org/article/opac-usability-problems-of-archives/153171](http://www.irma-international.org/article/opac-usability-problems-of-archives/153171)

### Towards A Semi-Automatic Transformation Process in MDA: Architecture, Methodology and First Experiments

Slimane Hammoudi, Wajih Alouini, Denivaldo Lopes and Marianne Huchard (2010). *International Journal of Information System Modeling and Design* (pp. 48-76).

[www.irma-international.org/article/towards-semi-automatic-transformation-process/47385](http://www.irma-international.org/article/towards-semi-automatic-transformation-process/47385)

### A Survey on Brain Tumor Segmentation and Classification

T.A. Jemimma and Y. Jacob Vetharaj (2022). *International Journal of Software Innovation* (pp. 1-21).

[www.irma-international.org/article/a-survey-on-brain-tumor-segmentation-and-classification/309721](http://www.irma-international.org/article/a-survey-on-brain-tumor-segmentation-and-classification/309721)

### Online Behavior Modeling: An Effective and Affordable Software Training Method

Charlie Chen, Terry Ryan and Lorne Olfman (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2230-2246).

[www.irma-international.org/chapter/online-behavior-modeling/29503](http://www.irma-international.org/chapter/online-behavior-modeling/29503)

### Modulation Recognition for Software Defined Radio Signal

Sudhar Sophia, M. Madheswaran and S. Sasikumar (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications* (pp. 398-412).

[www.irma-international.org/chapter/modulation-recognition-software-defined-radio/66479](http://www.irma-international.org/chapter/modulation-recognition-software-defined-radio/66479)