

An Empirical Measurement Project for Software Quality

Koji Torii and Mike Barker, NAIST (EASE)

Abstract

This paper addresses the need for software quality in the Japanese software industry, and what empirical software engineering can do in that area. It discusses the Japan Software Engineering Center, the Empirical Approach to Software Engineering Project and their collaboration. It also describes the Empirical Project Monitor, an open source application for data collection and analysis, which the EASE project has used in cooperation with industry partners to help introduce empirical software engineering practices in Japan to improve software quality.

1 Introduction

The software industry of today faces a crisis in software development. Software defects cause large social and economic losses. For example, defects in the software in mobile phones, trains, airplanes and banking systems threaten us all. In March 2005, a programming glitch occurred in the Shinkansen (Japanese bullet train) software by ATC. This glitch stopped 100 trains. In February, 2005, programme defects in engine control in eight models of cars led to recalls of the cars.

At the same time, the amount of development expected from developers continues to increase. The third generation cellular phone has about five million lines of code. This is the same scale as the third online system introduced into banking systems in 1985. However, even as sales of the information service industry increased by 45 per cent, the number of employees only increased by 6 per cent during the five years from 1998 through 2003.

2 Background

A Japan Information Service Industry Association (JISA) in 2004 indicated that 51 per cent of cost estimation is based on measurement data, but 43 per cent is still based on experience and analogy. This indicates the reality of measurement of customised software development.

Basili said that empirical software engineering focuses on building models which are experimentally validated in the real world through measurements, analysis, and feedback so that the understanding evolves over time. This empirical paradigm has been applied in other disciplines, and its scientific use in software engineering to collect, analyse and apply quantitative and qualitative data supports understanding and improving the software product, software development process, and software management. It does require real world laboratories, where researchers and developers work together to observe and manipulate the process of building software systems.[1]

Rombach points out that empirical observations allow us to characterise phenomena, laws based on repeated observations allow us to predict phenomena, and theories based on observations allow us to explain phenomena.[2]

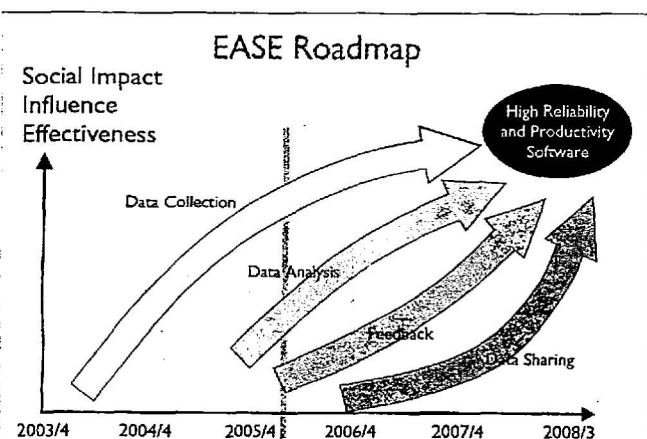
3 The Response

The Japan Software Engineering Center (SEC) was opened in October 2004 with a budget of 1.48 billion yen under the Ministry of Economy, Trade and Industry (METI). They are charged with conducting in-depth practical studies to solve the issues of today's software industry, especially in the areas of software process improvements, software measurement standards, and demonstrations of best practices in advanced software development projects. This mission requires establishment of a common understanding of the quantitative approach to project improvement and best practices for software development. Their goal of gathering data, analysing it, and promoting measurement standards connects with the Empirical Approach to Software Engineering project.[3]

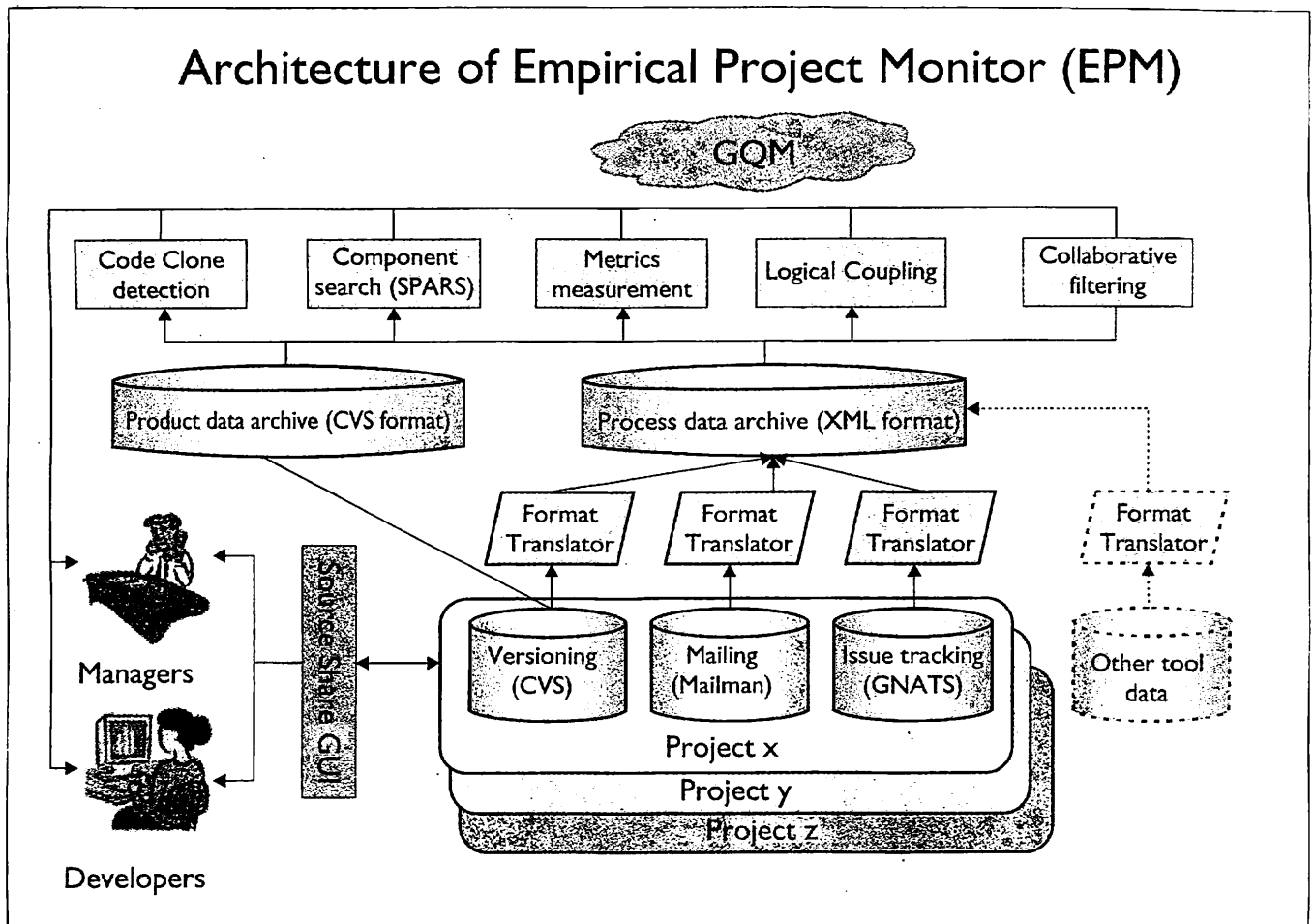
The Empirical Approach to Software Engineering (EASE) project is one of the leading projects of the Ministry of Education, Culture, Sports, Science and Technology (MEXT). Founded in 2003 with a budget of 200 million yen per year for five years, EASE aims to help the software industry in Japan adopt the empirical approach, including measurement, analysis, and feedback for improvement of software quality and productivity. To support this, the project has initially focussed on development of the Empirical Project Monitor and working with Japanese companies to use this in their work. This demonstrates what a quantitative data collection system looks like, provides examples of how data collection, analysis, and feedback work with industry in Japan, and allows for cooperation with SEC in their mission. [4]

The EASE Roadmap illustrates how the four technologies contribute to improving software quality and productivity in Japan.

The EASE project expects to produce three major levels of impact on software development in Japan. First, it provides a framework demonstrating how industry and academia can collaborate in



Architecture of Empirical Project Monitor (EPM)



empirical software engineering research. Since historically Japanese academic research has not been an active participant in industrial explorations, this constitutes a major step forward. This level of impact is relatively abstract and long-term. [5] Second, it provides guidance on using empirical methods in improving quality and productivity in the Japanese software industry. This educational mission is more concrete and can be expected to produce results in the mid-term as Japanese software companies apply the empirical paradigm. Third, EASE has developed an application called the Empirical Project Monitor (EPM) to help demonstrate data collection in industrial environments. This tool already has been used in several industry projects, and provides the most concrete and near-term impacts.

The Empirical Project Monitor (EPM) automatically collects development data from several development activities, including CVS for code development, GNATS or Bugzilla for defect tracking, and mailman or other mailing list managers for project email. The pluggable architecture allows companies to add other sources of project data for collection, and for researchers and analysts to expand the analysis tools.

For example, EASE is currently working on adding collaborative filtering, logical coupling, component search, and code clone

detection to the analysis tools. Collaborative filtering uses similarities in data sets to dynamically calculate values and reduce the effects of missing data. Logical coupling looks at how modules within programmes are related to predict the need for related modifications due to defect changes. Component search provides programming access to a library of components to encourage reuse. Code clone detection helps identify copied and similar code segments which can identify related systems and modules which may share defects. EPM provides a consolidated standard approach to collection, analysis, and feedback of empirical data for software development projects, allowing quantitative comparisons as modifications and improvements to the software process are made.

EASE project members are also working with industry to further extend EPM with tools and methods of empirical measurement to handle other parts of the software development lifecycle. In particular, the upstream areas of requirements collection and analysis and design pose issues for the industry and for empirical measurement. While research suggests that the largest impact on quality occurs through improvements in requirements and design, these are also extremely difficult areas for quantitative measurement and analysis. EASE is also examining how to relate