Software Inspections in Practice: Six Case Studies

Sami Kollanus[1] and Jussi Koskinen[2]

Department of Computer Science and Information Systems
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
[1]sami.kollanus@jyu.fi, [2]koskinen@cs.jyu.fi

Abstract. Software inspections have been acknowledged as an important method in software engineering, but they are not well applied in practice. This paper discusses the current practices and the related problems based on six case studies in industrial settings. The analysis of inspection practices was organized according to ICMM, which is a model for systematically assessing and improving software inspection process maturity. The sample case organizations used inspections relatively regularly. The involved units are compared and the revealed practices, their characteristics, inspection problems and implications of the study discussed. The main problem areas were non-existent inspection training, limited formality of inspections and immaturity of inspection metrics.

1 Introduction

Since Michael Fagan published his original software inspection method [2] in 1976, inspections have been acknowledged as an important method in software development. Several researchers have reported great savings or improved effectiveness when using inspections [7]. Unfortunately, regardless of the fact, that inspection is known as a useful method within the software engineering research community, it is not widely applied in practice. There also is very little systematic research conducted, attempting to study the real state of the industrial practices. There are, however, some relevant experience reports. Johnson [3] refers in his paper to an informal survey, where 80% of the 90 respondents practiced inspections irregularly or not at all in their organizations. Ciolkowski *et al.* [1] conducted a survey, which aimed to study the practice of any kinds of software reviews. Based on 226 responses, they also concluded that reviews are irregularly used in software industry.

This paper reports six case studies, which were conducted in organizations producing commercial software. We will later sometimes refer to this set of studied units simply as *case units*. The focus of the paper is to identify strengths and weaknesses in inspection subpractices in the units by using *Inspection Capability Maturity Model* (ICMM) [5]. Since it was necessary to understand the whole organizational context of applying software inspections, data was gathered via interviews. There is an earlier study [6], which provided preliminary problem analysis based on data from two case studies. This paper extends that study by a larger set of involved organizations, by providing information regarding the state of the covered inspection processes, and by gathering discussion concerning the implications.

J. Münch and M. Vierimaa (Eds.): PROFES 2006, LNCS 4034, pp. 377 – 382, 2006.

2 Inspection Capability Maturity Model

This section briefly introduces Inspection Capability Maturity Model (ICMM) [5], which is used in this paper as a framework in analyzing existing inspection practices in the involved organizations. ICMM supports: 1) inspection process maturity assessment, and 2) inspection process improvements. It resembles the internationally well known and well established CMM [9] model, but focuses on the assessment of the maturity of inspection practices instead of the whole software development process. Only the ICMM-levels 2-3 were used in this study, because the upper levels were not currently relevant in the case units.

The second level is called 'Practicing level' in ICMM. It requires an organization to practice inspections regularly. It includes the following specific *process areas*:

P1. *Requirement Inspections*. Requirements have to be inspected in organization's every project. Inspections have to include preparation and reporting.
P2. *Design Inspections*. At least the system architecture description and some other central design documents should be inspected.
P3. *Training for Leaders*. Training should be provided at least for inspection leaders.

The third, 'Defined level' requires an organization to have a well defined inspection process and it has focus on inspection effectiveness. The required process areas are the following:

P4. *Test Case Inspections*. This process area requires inspection of test cases.
P5. *Code Inspections*. The most important parts of the code must be inspected in every project. In addition, project plan should define, which documents are required to be inspected within the project.
P6. *Defined Process*. Inspection process must be defined and documented.
P7. *Training for all*. This includes training for all relevant stakeholders.
P8. *Customized Material*. Inspection support materials must be created and customized for the organization. The material may include for example standards, rules, checklists and scenarios.
P9. *Data Collection and Use*. Inspection data should be collected and used to monitor, control and improve the inspection process.
P10. *Organizational Policy*. There must be clear organizational policy and management's commitment to the inspection practices.
P11. *Assigned Responsibilities*. This refers to the formal responsibilities which concern the inspections.
P12. *Allocated Resources*. This refers to the formal allocation of resources for the inspections.

3 Data Gathering

These case studies had two main goals related to the involved organizations: 1) to find out *how inspections* (or less formal reviews [8]) *are practiced*, and 2) to find out *what are the faced inspection related problems*. The case studies were conducted in six software supplier units within five Finnish companies. These companies produce and