Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# How are software defects found? The role of implicit defect detection, individual responsibility, documents, and knowledge

Mika V. Mäntylä [*], Juha Itkonen

*Department of Computer Science and Engineering, Aalto University, Finland*

ARTICLE INFO

*Article history:*
Received 30 March 2013
Received in revised form 11 December 2013
Accepted 11 December 2013
Available online xxxx

*Keywords:*
Software testing
Defect detection
Activities
Documents
Human factors
Industrial questionnaire study

ABSTRACT

*Context:* Prior research has focused heavily on explicit defect detection, such as formal testing and reviews. However, in reality, humans find software defects in various activities. Implicit defect detection activities, such as preparing a product demonstration or updating a user manual, are not designed for defect detection, yet through such activities defects are discovered. In addition, the type of documentation, and knowledge used, in defect detection is diverse.
*Objective:* To understand how defect detection is affected by the perspectives of responsibility, activity, knowledge, and document use. To provide illustrative numbers concerning the multidimensionality of defect detection in an industrial context.
*Method:* The data were collected with a survey on four software development organizations in three different companies. We designed the survey based on our prior extensive work with these companies.
*Results:* We found that among our subjects ($n = 105$), implicit defect detection made a higher contribution than explicit defect detection in terms of found defects, 62% vs. 38%. We show that defect detection was performed by subjects in various roles supporting the earlier reports of testing being a cross-cutting activity in software development organizations. We found a low use of test cases (18%), but a high use of other documents in software defect detection, and furthermore, we found that personal knowledge was applied as an oracle in defect detection much more often than documented oracles. Finally, we recognize that contextual factors largely affect the transferability of our results, and we provide elaborate discussion about the most important contextual factors. Furthermore, we must be cautious as the results were obtained with a survey, and come from a small number of organizations.
*Conclusions:* In this paper, we show the large impact of implicit defect detection activities in four case organizations. Implicit defect detection has a large contribution to defect detection in practice, and can be viewed as an extremely low-cost way of detecting defects. Thus, harnessing and supporting it better may increase quality without increasing costs. For example, if an employee can update the user manual, and simultaneously detect defects from the software, then the defect detection part of this activity can be seen as cost-free. Additionally, further research is needed on how diverse types of useful documentation and knowledge can be utilized in defect detection.

## 1. Introduction

Finding defects before release is an important and costly software engineering activity that is typically achieved through software testing and reviews. Plenty of academic work on software testing exists, but the connection between the academic work and the realities of software industry have repeatedly been found weak [1,2]. Even though some researchers have studied the actual practice of software testing in the industry [3–5], the diversity of the practice of software testing has not been addressed in academic research. Regarding software review, it has been found that in the industry they often suffer from poor reviewer preparation [6–8] and find lower share of functional defects than evolvability problems [6,9], which suggest that they could be secondary to software testing in detecting functional defects. As the diversity of various defect detection activities that might exist in the industry has not been studied in academic research, we study in this paper the variety of roles, activities, documents, and knowledge used by the people who detect defects in software development organizations.

In our previous case and field observation studies, we have identified the diversity of roles and activities, as well as documentation and knowledge, that are involved in defect detection and testing [10,11]. In this paper, we introduce the concept of implicit

[*] Corresponding author. Address: P.O. Box 15400, FI-00076 Aalto, Finland. Tel.: +358 505771684.

*E-mail addresses:* mika.mantyla@aalto.fi (M.V. Mäntylä), juha.itkonen@aalto.fi (J. Itkonen).

0950-5849/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.12.005

defect detection and study the amount and types of both implicit and explicit defect detection activities performed in four software development organizations. *Implicit defect detection* is an activity where one assesses the quality of the product and detects defects while working toward some other primary goal. The idea that humans can find defects while working towards some other goals has been previously investigated [12]. Additionally, it is utilized in industrial beta testing programs [13] and in internal usage of a company's own software products that is called alpha testing, or dogfooding, i.e., eating your own dog food [14,15]. Prior work has also studied the shares of defects detected in different types of testing and reviews [16,17]. However, to our knowledge, prior work has not studied the relative amount between implicit and explicit defect detection in software development organizations. For example, how many defects are found when testing the software vs. other software development activities that are not primarily done for QA purposes?

This study uses a survey instrument to provide a picture of defect detection activities at an organizational level in four case organizations. In this analysis, we study defect detection activity, the responsibility of individual finding defects, the type of documentation used in defect detection and the oracle information in defect detection. This study extends the earlier observation and case studies [10,11] that have identified the importance of personal knowledge in software testing, by investigating the amount and type of document and knowledge used at the organizational level.

This paper is structured as follows. Next, we present the research methodology and the analytical framework that we used in our analysis. In Section 3, we describe the results of the survey. In Section 4, we discuss our findings and present the related work. Finally, in Section 5, we provide the conclusion of this work.

## 2. Methodology

We collected data through a survey questionnaire from four software development organizations that we know well due to long-term research collaboration. We distributed the survey of defect detection in the development organizations and aimed it at wide coverage of professionals working in wide variety of roles. The measured variables are the number of found defects, document use, activities performed, personal knowledge, and organizational responsibilities.

Next, we describe the analytical framework in Section 2.1, followed by the definition of the exact research questions in Section 2.2. We continue with a description of the survey instrument and data collection in Section 2.3. Section 2.4 describes the measures and the data analysis procedures in detail. We introduce the case companies and the subjects of the survey in Section 2.5. Finally, Section 2.6 discusses the limitations of this study.

### 2.1. Analytical framework

The conceptual framework that we use in the analyses of the survey consists of three main dimensions concerning the defect detection phenomenon. The central concepts in our framework are described and motivated by the existing literature: *implicit and explicit defect detection*, *tester and non-tester roles,* and *documentation and knowledge* used in defect detection.

First, we propose dividing defect detection activities into explicit and implicit defect detection (see *Y*-axis in Fig. 1). We define *explicit defect detection* as an activity whose primary goals are to find defects and assess the quality of the product. Both goals of explicit defect detection can be achieved by various testing and review methods. In this paper, the explicit defect detection activities are software testing and software reviews or walkthroughs. We define

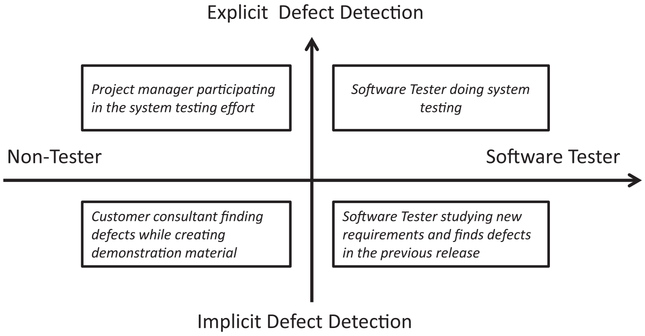
Explicit Defect Detection
Project manager participating in the system testing effort
Software Tester doing system testing
Non-Tester
Software Tester
Customer consultant finding defects while creating demonstration material
Software Tester studying new requirements and finds defects in the previous release
Implicit Defect Detection

**Fig. 1.** Defect detection activity quadrants. *X*-axis represents the role dimension and *Y*-axis represents the activity dimension of software defect detection.

*implicit defect detection* as an activity where one assesses the quality of the product and finds defects while working toward some other primary goal. We argue that almost all people have performed implicit defect detection. Implicit defect detection is very common in our lives as we form opinions and find defects in the things that we use. For example, we have opinions about the quality of our car, our smartphone, or the school we send our children to. Furthermore, we have probably found defects in the things that we use, and we might have even reported these defects back to the responsible organizations. The same kind of implicit defect detection goes on in software product companies, e.g., when a sales person creates a demonstration for the upcoming product release, he/she is performing implicit defect detection as there is a chance that the upcoming release might still have undiscovered defects. Implicit defect detection has also been harnessed by software companies by requiring their employees to use the upcoming alpha versions of the products, called alpha testing or dogfooding [15]. The implicit defect detection performed by external people is called beta testing [13]. The implicit–explicit distinction can be seen as part of experimental designs where subjects have had multiple goals, e.g., perform pension calculation and report data quality defects [12], and create high level test cases and find requirements defects [18]. However, in general, the idea of implicit software defect detection has received limited attention in prior works. We think that large shares of implicit defect detection happen in software development organizations every day, thus, the topic needs to be addressed.

Second, we study the organizational roles of the people performing software defect detection (both implicit and explicit). We divide the defect detection activity based on the roles *tester* and *non-tester* (see *X*-axis in Fig. 1). In our prior case study which was based on defect database data, we found that large shares of defects were found by non-testers [10]. The large contribution of non-testers to defect detection might be more common than previously thought, as further work by us [19], and independent researchers [20], has supported this finding. This paper extends prior works by connecting the roles with different implicit and explicit, defect detection activities as illustrated by the *defect detection activity quadrants* in Fig. 1. In this research, we use a survey instrument to replicate and confirm the results of earlier work that was based on database analysis [10] and interviews [19,20].

Third, we study the documents used and knowledge applied in software defect detection. Earlier work has indicated that documented test cases in manual testing in the software industry are often far from textbook examples and are sporadically used [4,10,21–23]. Furthermore, the benefits of having pre-designed test case documentation in manual testing in terms of defect detection effectiveness are questionable according to experiments comparing test-case-based and exploratory testing [24,25]. Thus, the ques-

Please cite this article in press as: M.V. Mäntylä, J. Itkonen, How are software defects found? The role of implicit defect detection, individual responsibility, documents, and knowledge, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2013.12.005