Information and Software Technology 56 (2014) 793–806

Contents lists available at ScienceDirect

Information and Software Technology

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

Analyzing the relationships between inspections and testing to provide a software testing focus

Frank Elberzhager [a,*], Jürgen Münch [b], Danilo Assmann [c]

[a] Fraunhofer Institute for Experimental Software Engineering (IESE), Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
[b] University of Helsinki, P.O. Box 68, Gustaf Hällströmin katu 2b, 00014 Helsinki, Finland
[c] Vector Informatik GmbH, Ingersheimer Straße 24, 70499 Stuttgart, Germany

ARTICLE INFO

Article history:
Received 14 June 2013
Received in revised form 20 January 2014
Accepted 17 February 2014
Available online 5 March 2014

Keywords:
Inspection
Testing
Integration
F-measure
Case study

ABSTRACT

Context: Quality assurance effort, especially testing effort, is frequently a major cost factor during software development. Consequently, one major goal is often to reduce testing effort. One promising way to improve the effectiveness and efficiency of software quality assurance is the use of data from early defect detection activities to provide a software testing focus. Studies indicate that using a combination of early defect data and other product data to focus testing activities outperforms the use of other product data only. One of the key challenges is that the use of data from early defect detection activities (such as inspections) to focus testing requires a thorough understanding of the relationships between these early defect detection activities and testing. An aggravating factor is that these relationships are highly context-specific and need to be evaluated for concrete environments.
Objective: The underlying goal of this paper is to help companies get a better understanding of these relationships for their own environment, and to provide them with a methodology for finding relationships in their own environments.
Method: This article compares three different strategies for evaluating assumed relationships between inspections and testing. We compare a confidence counter, different quality classes, and the *F*-measure including precision and recall.
Results: One result of this case-study-based comparison is that evaluations based on the aggregated *F*-measures are more suitable for industry environments than evaluations based on a confidence counter. Moreover, they provide more detailed insights about the validity of the relationships.
Conclusion: We have confirmed that inspection results are suitable data for controlling testing activities. Evaluated knowledge about relationships between inspections and testing can be used in the integrated inspection and testing approach In^{2}Test to focus testing activities. Product data can be used in addition. However, the assumptions have to be evaluated in each new context.

1. Introduction

Verification and validation are an indispensable part of modern software development. The effort and costs for performing such quality assurance are often rather high; testing, in particular, can consume 50% or more of the overall effort in certain environments [1–3]. Still, the resulting quality of the software is often poor, i.e., many defects are found by customers after the release of the software. This can lead to high rework costs or even entail risks for human beings in safety–critical environments.

Different approaches for optimizing quality assurance exist [4]. One of these approaches is automation. For instance, deriving test cases or performing testing could be automated to save time and execute more test cases than with a manual procedure. Another approach is to focus testing activities on those parts that are expected to be defect-prone. To do this, different metrics are usually considered and relationships between such metrics and defect-proneness are exploited. For example, if a correlation between large modules and defect-prone parts were to be shown in a certain environment, such knowledge could be used in future development cycles to allocate test effort to particularly large modules. Common metrics are product metrics (e.g., size, complexity) or process metrics (e.g., number of developers per module, number of changes per module). Typically, metrics applied to

* Corresponding author. Tel.: +49 631 6800 2248; fax: +49 631 6800 92248.
E-mail addresses: frank.elberzhager@iese.fraunhofer.de (F. Elberzhager), juergen.muench@cs.helsinki.fi (J. Münch), danilo.assmann@vector.com (D. Assmann).

http://dx.doi.org/10.1016/j.infsof.2014.02.007
0950-5849/

defect information that is available early, e.g., from inspections or reviews, are currently not used to focus testing activities.

As static quality assurance techniques such as inspections and reviews, and dynamic quality assurance techniques such as testing typically have the same objective – finding defects, and thus improving the software – there is a basis for a joint application, and it seems promising to also consider data from inspections to provide a software testing focus. We have proposed the integrated inspection and testing approach In²Test [5], which explicitly uses inspection defect data to focus testing activities. In²Test uses so-called selection rules, which define a software testing focus by considering different metrics. This approach has to be calibrated first, which requires knowledge about the relationships between inspections and testing for the context (i.e., the development environment) at hand. Such kind of knowledge is typically not available [6]. Therefore, the In²Test approach starts with assumptions about the relationships between inspections and testing and evaluates them. There are different possibilities for doing such an evaluation. In this article, we compare three different evaluation procedures (i.e., confidence, quality classes, and *F*-measure including precision and recall) in order to analyze their suitability.

Earlier studies have shown that the In²Test approach is applicable and more efficient than a non-integrated approach (within the context and limits of the studies) [5,7]. We use the term non-integrated approach to refer to all approaches that combine inspections and testing such that these activities are performed independently from each other. The results from these studies have shown that using early defect data to focus test activities is slightly more efficient than considering only selected product data to focus test activities. In addition, a combination of early defect data and product data saved significantly more effort than using only product data [8]. However, we considered only an informal evaluation scheme, which offers only low significance when deriving long-term conclusions. Consequently, we extend and detail our analysis in this article, concentrating on the following overall research question:

> Which evaluation procedure leads to the highest validity of selection rules for focusing testing based on inspection defect data?

With the analysis presented in this article, we directly answer the overall research question and substantiate the results from our earlier studies with more detailed analyses. The validity of our analysis results is limited to the scope of the presented case studies and the resulting knowledge for focusing testing cannot be treated as universally applicable rules without new evaluations in other environments. However, this article shows a procedure for analyzing inspection data in different contexts to find valid rules for one's own context and, as a consequence, for optimizing quality assurance in a specific context. When presenting our results and the approach, we motivate that it is worth checking inspection and testing defect data and corresponding rules in industrial environments in order to further optimize testing activities.

The remainder of this article is structured as follows: Section 2 reports on related work regarding approaches that aim at providing a testing focus, while Section 3 presents the In²Test approach at a glance, including the three different evaluation procedures. Section 4 presents results from two case studies from different environments where the calibration of the approach was performed and where the different evaluation procedures were applied. Section 5 points out the main lessons learned. Finally, Section 6 concludes this article with a summary and an outlook on future work.

2. Related work: focusing testing activities

Various approaches for focusing testing activities have been proposed. One very popular and established approach is the use of different kinds of metrics for predicting defect-proneness. The idea is that certain correlations exist between metrics and defects in a system. If such relationships are found, they can be used to predict parts where defects are expected, and testing can be focused on such areas. One typical differentiation of data used to calculate such metrics is to distinguish product data, process data, and historical defect data. A lot of different studies have been performed to find such relationships. The first metrics that were considered were size or complexity. While in many studies, correlations could be found that led to suitable predictions of defect-prone parts, these findings were highly context-dependent. Some studies showed, for instance, that large modules contain more defects than small ones, while others showed the opposite [9,10]. D'Ambros et al. [11] present an overview of several metrics and provide an extensive comparison of those metrics. They found that a combination of different metrics turned out to be best in their contexts, but mentioned that those metrics might be of lower quality in other environments. Arisholm et al. confirmed these conclusions [12]. However, defect data from early quality assurance activities such as reviews or inspections are typically not considered when predicting defect-proneness in order to focus testing activities. Also, knowledge about the relationships between inspections and testing is very limited [6].

Another approach for focusing testing is to consider expert knowledge. People with a lot of experience in a certain environment usually know their software very well, and quality engineers often know intuitively where more problems might occur. For example, Nasser et al. [13] describe a knowledge-based approach for test case generation. Knowledge and experiences from inspection experts are sometimes used for defect number predictions to control inspection activities [14], but are typically not used to focus testing. In addition, expert-based approaches significantly depend on the experience level of the appropriate experts as well as on their availability (which is often an issue in practical situations). Furthermore, risk-based approaches can be considered to focus testing [15].

Hybrid approaches are another way to focus testing by combining expert knowledge and data. The Hydeep approach, for example, is able to predict effectiveness and defect values [16]. Such data can be used to focus testing on those parts where the highest number of defects is expected. Fenton et al. [17] propose using Bayesian nets to predict defect numbers, which can be used to decide when to stop testing, but also contribute to the identification of defect-prone parts that should be in focus during testing activities. However, inspection data are typically not used for focusing testing.

3. Background: the integrated inspection and testing approach In²Test

3.1. The In²Test approach at a glance

The basic idea of the integrated inspection and testing approach is to use inspection defect data to predict defect-prone parts or defect types for testing that are likely to appear during testing.

The process starts with an inspection (step 1). To remain flexible, no specific inspection technique is required; for example, a formal inspection, a team review, or a more informal peer desk-check could be used to find defects. Different kinds of inspection metrics can be applied, such as number of defects per part or defect density per part, or a defect classification can be used to classify each defect found.

As inspection is a manual activity and the result of an inspection depends on several context factors such as the experience of the inspectors, the kind of software product, or the inspection process,