Available online at www.sciencedirect.com

ScienceDirect

Information and Software Technology 50 (2008) 782–796

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

Predicting weekly defect inflow in large software projects based on project planning and test status

Miroslaw Staron [a,*], Wilhelm Meding [b]

[a] *IT University of Göteborg, Förskningsgången 6, 412 96 Göteborg, Sweden*
[b] *Ericsson SW Research, Ericsson AB*

Received 8 May 2007; received in revised form 25 September 2007; accepted 1 October 2007
Available online 12 October 2007

Abstract

Defects discovered during the testing phase in software projects need to be removed before the software is shipped to the customers. The removal of defects can constitute a significant amount of effort in a project and project managers are faced with a decision whether to continue development or shift some resources to cope with defect removal. The goal of this research is to improve the practice of project management by providing a method for predicting the number of defects reported into the defect database in the project. In this paper we present a method for predicting the number of defects reported into the defect database in a large software project on a weekly basis. The method is based on using project progress data, in particular the information about the test progress, to predict defect inflow in the next three coming weeks. The results show that the prediction accuracy of our models is up to 72% (mean magnitude of relative error for predictions of 1 week in advance is 28%) when used in ongoing large software projects. The method is intended to support project managers in more accurate adjusting resources in the project, since they are notified in advance about the potentially large effort needed to correct defects.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Software metrics; Predictions; Defects

1. Introduction

Discovering defects during the development process in large software projects has usually been seen as an activity related to quality management. The question of the number of defect is investigated in the context of defect density [4,20], predicting the final quality of software products [3], fault slip-through [8], software inspections [23], or predicting the number of defects remaining in the software after the inspection process [13,22]. These studies are focused on the defects as a measure of software quality, while from the project and quality management perspectives, defects discovered during the project indicate also work effort required to complete the project with a given quality level. When planning projects, software organizations account for a certain effort needed to correct defects discovered during testing – i.e. defect removal before the maintenance phase. An important aspect, nevertheless, is whether the effort is distributed correctly on a weekly basis. If a higher than expected number of defects appear, the project might not have capacity (at a given time) to remove these defects while progressing, which in turn might lead to project delays. If a lower than expected number of defects arises, the costs of the project are unnecessary high and the resources are unnecessarily blocked instead of being used in other projects at the same organization. In this paper, we address the issue of predicting an immediate defect inflow in large software projects in order to support project managers in managing the project and quality managers in controlling quality and project performance.

We define the *defect inflow* as the number of non-redundant defects being reported into the defect database. The non-redundant defects in the database are usually non-trivial defects found in the system, which need to be

* Corresponding author. Tel.: +46 31 772 1081.
E-mail addresses: miroslaw.staron@ituniv.se (M. Staron), wilhelm.meding@ericsson.com (W. Meding).

0950-5849/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2007.10.001

investigated and fixed. The effort required to remove these defects is considerable if the number of defects reported is high. Therefore the project and quality managers appreciate advance notification of potential problems ahead. A simple estimation equation used at the company is

$$\text{Effort} = C_{\text{DefectEffort}} * \text{Number of Defects}$$

where $C_{\text{DefectEffort}}$ is the constant denoting the average amount of effort required to fix a defect. This constant is based on experience at the company, obtained by examining the historical data. Although this average value of cost per defect is not an accurate value, in practice an average value is a good estimator because of two reasons:

- There is always a lower limit (more than 0) for handling the defects – reporting, analyzing, fixing (or rejecting), and in practice most defects are not rejected, but fixed.
- There is an upper limit for handling the defects: even if the defect is very difficult to fix, if it is important (which is very often the case for difficult defects) then the company puts in extra resources (e.g. experts) to solve it in order to continue with development/testing.

Hence, in practice, the average value is often used by project managers during estimations and planning.

The independent variable in this equation (*Defects*) is the focus of our research as in this paper we address the following research question:

How can we predict defect inflow for 3 weeks in advance?

We address this research question by constructing and evaluating models for predicting defect inflow in two projects (a concluded baseline project where we create the models and an ongoing project where we evaluate them). We use multivariate linear regression to create prediction models for the defect inflow for up to 3 weeks in advance. We constructed one model per each week which we predict – i.e. 1, 2, and 3 weeks in advance. The prediction models presented in this paper are based on the plan of testing progress and their execution status. These models are complemented with the models developed previously and presented in [21] and the current models can be used when the previously presented models are not applicable due to violation of their assumptions.

The models are developed in close cooperation with our industrial partner – Ericsson. The models presented in this paper are constructed based on data from a baseline project at this organization and they are evaluated in another ongoing large software project. The results of the evaluation show that the accuracy of the predictions is as high as 72% and that the models provide useful predictions.

This paper is structured as follows. Section 2 presents the related work important for our research. Section 3 describes the context of our study – the defect inflow in large projects in the organization. Section 4 presents our research method and Section 5 presents the candidate prediction models, the final prediction models and their evaluation. Finally, Section 6 presents our conclusions.

2. Related work

In our work we consider the defect inflow to be the function of characteristics of work packages (e.g. the accumulated number of work packages reaching a particular milestone) and not directly the characteristics of the affected components (e.g. size or complexity). Using the characteristics of components as the sole predictors would only provide us with a possibility to predict the defect density of the component and extrapolate this data on a weekly/monthly basis (based on when the component will be put under testing). Such an approach would be an extension of the current work on defect density, e.g. [1,3,4,17,18,20]. In our case, nevertheless, this approach seems not feasible, because the information about how the components are to be affected by the project is not available at the time of developing predictions; in particular the change of size and complexity is not available. Furthermore, for large software projects, the predictions of defect densities have been found to be insufficient [10]. For short-term predictions, the data on size and complexity of components was not available on a weekly basis simply because measuring the size and complexity change is not meaningful for particular weeks; the measurements of component characteristics are done according to project plans – e.g. builds – and not on a weekly basis (i.e. not according to calendar time). In our future work we intend to evaluate whether it is feasible to reconfigure this data and use it as an auxiliary prediction method.

Another alternative approach is to use data on test cases to predict the number of defects (e.g. using such methods as Capture–Recapture [13,22]). For the short-term predictions we intend to use this in the next step of our research, however, this data still needs reconfiguration since it suffers from the same problems as mentioned above – i.e. being done on a subsystem/component basis and not on the work package basis.

Although the prediction of defect inflow seems to be closely related to the area of reliability modeling, it differs significantly from it. In particular, the reliability modeling is concerned about the software reliability after release, e.g. [2,7,16], or [11], while our research is focused on the rate of defect inflow (i.e. discovery and reporting) during the project development. In early stages of our research we applied the Rayleigh model [15], and discovered that the defect inflow profile in the studied organization is significantly different from the profile described by the Rayleigh model.

Finally, our research is related to the research on fault slip-through [8]. In our research we intend to use the methods for predicting fault slip-through as the next refinement of this method. However, at this point of time the data for fault slip-through was not available at weekly basis due to the definition of fault slip-through. We intend to work on using this measurement later on to improve the results of our work.

Download English Version:

https://daneshyari.com/en/article/549942

Download Persian Version:

https://daneshyari.com/article/549942

Daneshyari.com