Information and Software Technology 63 (2015) 44–57

Contents lists available at ScienceDirect

Information and Software Technology

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

A fuzzy logic based approach for phase-wise software defects prediction using software metrics

Harikesh Bahadur Yadav, Dilip Kumar Yadav *

Department of Computer Applications, National Institute of Technology, Jamshedpur 831014, India

ARTICLE INFO

Article history:
Received 29 August 2014
Received in revised form 24 January 2015
Accepted 2 March 2015
Available online 19 March 2015

Keywords:
Software defect
Software defect density indicator
Software metrics
Fuzzy logic
Software reliability

ABSTRACT

Context: The software defect prediction during software development has recently attracted the attention of many researchers. The software defect density indicator prediction in each phase of software development life cycle (SDLC) is desirable for developing a reliable software product. Software defect prediction at the end of testing phase may not be more beneficial because the changes need to be performed in the previous phases of SDLC may require huge amount of money and effort to be spent in order to achieve target software quality. Therefore, phase-wise software defect density indicator prediction model is of great importance.
Objective: In this paper, a fuzzy logic based phase-wise software defect prediction model is proposed using the top most reliability relevant metrics of the each phase of the SDLC.
Method: In the proposed model, defect density indicator in requirement analysis, design, coding and testing phase is predicted using nine software metrics of these four phases. The defect density indicator metric predicted at the end of the each phase is also taken as an input to the next phase. Software metrics are assessed in linguistic terms and fuzzy inference system has been employed to develop the model.
Results: The predictive accuracy of the proposed model is validated using twenty real software project data. Validation results are satisfactory. Measures based on the mean magnitude of relative error and balanced mean magnitude of relative error decrease significantly as the software project size increases.
Conclusion: In this paper, a fuzzy logic based model is proposed for predicting software defect density indicator at each phase of the SDLC. The predicted defects of twenty different software projects are found very near to the actual defects detected during testing. The predicted defect density indicators are very helpful to analyze the defect severity in different artifacts of SDLC of a software project.

1. Introduction

Nowadays, people are working under direct or indirect influence of software. Human dependence on software has been increased since last three decades. Software reliability and quality modeling is essential because the software is used in diverse area of various applications. Various historical events illustrate the effect of software failures encountered in and around the world [1]. The consequences of software failures may result in monetary and human losses. Therefore, software quality and reliability prediction is unavoidable and it has become a major research area.

A general method to measure the quality of software is to reveal the presence of defects in it, and usually the metric used for it is software defect density. The software defect density is defined as the total number of defects divided by the size of the software. A defect can also be defined as a product anomaly [2]. Software reliability is the probability that software will not cause the failure of a system for a specified period of time under the specified conditions. The probability is a function of the inputs to, and use of, the system as well as a function of the existence of faults in the software. The inputs to the system determine whether existing faults, if any, are encountered. Software reliability model were designed to quantify the likelihood of software failure. A failure is defined as the termination of the ability of a functional unit to perform its required function. In the other way, failure can defined as an event in which a system or system component does not perform a required function within specified limits [3].

Software defect density prediction plays an important role in producing reliable software. In order to achieve target defect estimate, it is required to predict the defect density indicator at the end of each phase of the SDLC. Numerous models have been proposed for estimation and prediction of software reliability in the

* Corresponding author. Tel.: +91 9931897599.
E-mail addresses: yadavaharikesh@gmail.com (H.B. Yadav), dkyadav1@gmail.com (D.K. Yadav).

http://dx.doi.org/10.1016/j.infsof.2015.03.001
0950-5849/

past three decades. It is observed that traditional models for software reliability prediction are neither universally successful in predicting the reliability of the software nor generally tractable to users [4]. The majority of models are based on probabilistic approach. Failure data are not available in the early phases of the SDLC. However, the failure information during the early phases of the SDLC is available in the form of expert knowledge which may be reflected in terms of software metrics [5]. In fact, most of the software metrics are associated with uncertainty. The smaller size of software testing data, unrealistic assumptions, and the fact that some measures cannot be defined precisely, are the key reasons that a fuzzy logic based approach should be consider for predicting the software defects. The term defect density indicator (DDI) is used in this proposed model to stand for defect density at the end of the each phase of the SDLC. Therefore, in this paper, a fuzzy logic based phase-wise software defect prediction model is proposed using the reliability relevant metrics of the each phase of the SDLC.

The rest of the paper is organized as follows: In Section 2, related work is discussed. In Section 3, the proposed methodology is presented. Section 4 and 5 describe the twenty case studies and predicted result, respectively. In Section 6, model validation is described. Sensitivity analysis is discussed in Section 7. Conclusion is presented in Section 8.

2. Related work

The software development is performed by a human being that is measured in terms of man-hour. The man-hour is of fuzzy nature. Therefore, failure-free software development is a challenging task. It is very essential to ensure that the underlying software will perform its intended functions correctly. Therefore, there is a growing need to ensure reliability of software systems as early as possible.

Numerous models have been proposed for estimation and prediction software reliability [6,7]. The air force's Rome laboratory [8] developed a model for early software reliability prediction. In this model, they selected some factors that are related to the fault density in the requirement analysis, design, coding, and testing phases. The model is mainly based on the software requirement specification document. Agresti and Evanco [9] proposed a model to predict the defect density on the basis of process and product characteristics. This model uses multivariate regression analysis for defect density prediction. In a similar study, Wohlin and Runeson [10] proposed two novel methods to estimate the number of remaining defects based on the review information. Conclusions about the remaining number of defects are then drawn after reviews. Smidts et al. [11] proposed software reliability prediction model based on the requirements of software and failure modes. The input of this model is failure data however these data would not be available during early phases unless there would have been similar executed projects. Early reliability assessment of UML based models starts with analysis of the unified modeling language model of software architecture followed by the Bayesian framework for reliability prediction. However, this model assumes that failures of components are independent of each other. This means that software coding languages allowing shared state information are not eligible for this method [12]. A phase based model for predicting reliability was proposed by Gaffney and Davis [13,14]. The model is mainly based on the fault statistics found during the review of various software development phases. In short, these traditional models for defect prediction are organization specific and not flexible.

The main factor influencing the number of residual defects in software is the size of the software [15,16]. Residual defect of software can be predicted by using only software complexity or software size metric [17]. According to Fenton and Neil [18], most of the software defects prediction models use size and complexity metrics to predict the residual defects. Software size metrics measures the intrinsic complexity of the software [19]. Therefore, it is better to use size and other software metrics for prediction of the software residual defects. A model for defect prediction with Bayesian net is developed by Fenton et al. [20]. The main feature is that it does not require detailed domain knowledge and it combines both qualitative and quantitative data. Mohanta et al. [21,22] proposed a model to predict the reliability of object-oriented systems during the early stages of the product development based on bottom-up approach. In this approach, the reliability of the system is estimated based on operation profile and reliabilities of classes. Okutan and Yildiz [23] proposed a novel method using Bayesian networks to explore the relationships among software metrics and defect proneness. Dejaeger et al. [24] compare 15 different Bayesian network classifiers with famous defect estimation methods on 11 data sets. They concluded that simple and comprehensible networks with fewer nodes can be constructed using the Bayesian network.

Zhang and Pham [25] suggested thirty-two factors which have impact on the software reliability in all stages of the software development process. A similar study conducted by Li et al. [26,27] where they have done phase-wise ranking of software metrics which influence the software reliability. These software metrics were ranked with respect to their ability in predicting software reliability through an expert opinion elicitation process.

Catal and Diri [28,29] provided a systematic review of various software fault prediction studies with focus on metrics, methods, and datasets. Radjenovic et al. [30] reported that process metrics are more successful in finding the faults compared to traditional size and complexity metrics. Hall et al. [31] find out that out of 208 studies only 36 studies are useful and majority is less useful than they could be. This makes it difficult for software developer team to select a model to match their context.

Pandey and Goyal [32] have proposed an early fault prediction model using process maturity and software metrics. They have considered the fuzzy profiles of various metrics in different scale and have not explained the criteria used for developing these fuzzy profiles. Yadav et al. [33] proposed a software defect prediction model in which they had considered only the uncertainty associated over the assessment of software size metric and three metrics of requirement analysis phase. Recently, Pandey and Goyal [34] developed a multistage model for residual fault prediction. They have considered 10 software metrics as input to the model. Maa et al. [35] analyze the ability of requirement metrics for software defect prediction during design phase.

On the basis of above literature survey and review, it is found out that the software reliability is a function of number of residual defect in the software. Reliability relevant software metrics play a vital role in defect prediction and these metrics are of fuzzy nature. Therefore, in this research paper, a fuzzy logic based model for phase-wise software defects density prediction is developed using the reliability relevant software metrics.

3. Proposed methodology

The architecture of the proposed model is shown in Fig. 1. In the proposed model, defect density indicator in the first four phases of SDLC is predicted based on the measures present in first four phases of SDLC. Therefore, proposed model leverages the reliability relevant top metrics [27] of the requirements analysis, design, coding and testing phases of SDLC. Metrics are denoted using elliptical structures in the proposed model. Requirements stability,