The Journal of Systems and Software 98 (2014) 59–78

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Selecting software reliability growth models and improving their predictive accuracy using historical projects data

Rakesh Rana[a,*], Miroslaw Staron[a], Christian Berger[a], Jörgen Hansson[a], Martin Nilsson[b], Fredrik Törner[b], Wilhelm Meding[c], Christoffer Höglund[d]

[a] Computer Science & Engineering, Chalmers/University of Gothenburg, Göteborg, Sweden
[b] Volvo Car Corporation, Göteborg, Sweden
[c] Ericsson, Göteborg, Sweden
[d] SAAB AB, Göteborg, Sweden

ARTICLE INFO

Article history:
Received 20 March 2014
Received in revised form 11 August 2014
Accepted 19 August 2014
Available online 28 August 2014

Keywords:
Embedded software
Defect inflow
Software reliability growth models

ABSTRACT

During software development two important decisions organizations have to make are: how to allocate testing resources optimally and when the software is ready for release. SRGMs (software reliability growth models) provide empirical basis for evaluating and predicting reliability of software systems. When using SRGMs for the purpose of optimizing testing resource allocation, the model's ability to accurately predict the expected defect inflow profile is useful. For assessing release readiness, the asymptote accuracy is the most important attribute. Although more than hundred models for software reliability have been proposed and evaluated over time, there exists no clear guide on which models should be used for a given software development process or for a given industrial domain.

Using defect inflow profiles from large software projects from Ericsson, Volvo Car Corporation and Saab, we evaluate commonly used SRGMs for their ability to provide empirical basis for making these decisions. We also demonstrate that using defect intensity growth rate from earlier projects increases the accuracy of the predictions. Our results show that Logistic and Gompertz models are the most accurate models; we further observe that classifying a given project based on its expected shape of defect inflow help to select the most appropriate model.

1. Introduction

Embedded software is today an integral part of most products, on which we depend for smooth functioning of our daily life. Embedded software does not only provide functionality, it also drives innovation in mobile phones, satellite systems, home appliances, and aircrafts. Reliability is an important attribute of such systems and one way of evaluating their reliability is to use software reliability growth models (SRGMs). SRGMs are the result of applying reliability engineering theory to the software development domain. The defect inflow data is modeled using mathematical models that quantify the change in reliability of the given software artifact during its development and testing. SRGMs help to answer an important practical question as to when the given software quality is good enough and thus, when can we stop testing (Dalal and Mallows, 1988). The good-enough quality is also referred to as release readiness of a given product (Kapur and Bhalla, 1992). From the reliability standpoint, one of the most important factors for deciding if a software is ready for release is the number of remaining defects (latent defects). By comparing the predicted total number of defects (asymptote of SRGMs) and the number of defects discovered and resolved to date, software managers can decide if the software is ready to be released (Quah, 2009).

Apart from answering the important release readiness question, SRGMs can also be used to make the software testing process more efficient (Malaiya et al., 1992). However, requirements for the successful application of SRGMs for optimal resource allocation and the assessment of release readiness of software differ. Models which can be applied early in the project and have higher ability to accurately forecast the expected shape of the defect inflow profile are useful for optimizing test resource allocations. While SRGMs that are accurate in forecasting total expected defects in a software product (asymptote) late in the development/testing phase are better suited for assessing the release readiness of a given software system.

* Corresponding author at: Department of Computer Science and Engineering, Chalmers/University of Gothenburg, Hörselgången 5, 417 56 Göteborg, Sweden. Tel.: +46 0 720 414 344.
E-mail addresses: rakesh.rana@gu.se, rakeshrana555@yahoo.co.in (R. Rana).

http://dx.doi.org/10.1016/j.jss.2014.08.033
0164-1212/

Although more than hundred SRGMs have been proposed and evaluated in the literature (Lyu, 2007), many of the earlier studies evaluating SRGMs have focused only on how well they could fit to the observed defect inflow data. The evaluation of the predictive power of SRGMs in the literature has generally been limited to only the last few data points (typically last 10% of data) (Rana et al., 2013a; Pham, 2003). The difficulty of applying SRGMs in industry is compounded with the lack of studies focusing on specific industrial domains (Rana et al., 2013c) and scarce guidelines to select the best SRGMs for a given software process/application. We focus on the following research questions that are important for reliability practitioners and project managers in software organizations, denoted RQ1–RQ4 below:

Since software development projects have a planned amount of testing resources, we explore how SRGMs can help to allocate these resources more effectively. We assess which SRGMs are best for this purpose, i.e. we evaluate the SRGMs' ability to correctly predict the shape of the future defect inflow during an on-going project.

RQ1. Which SRGMs are best to assist decisions for optimal allocation of testing resources?

When the software system has been developed and tested, the most important question is: Is the software ready to be deployed (released) or does it need more testing? We evaluate, which SRGMs are best for assessing the release readiness of software systems from the reliability standpoint.

RQ2. Which SRGMs are best for assessing the release readiness of a software system?

Given that software development organizations usually have data on a large number of historical projects, it is also important to evaluate how we can use this experience to make the reliability predictions for current projects more accurate. This is addressed by the following research question:

RQ3. Does using information from earlier projects improve release readiness assessment?

Further, there exists no agreement on which models are the best for a given software development process or industrial domain especially during the early phases of a software project (Ullah et al., 2012), thus we analyze how to select the best SRGM for a given purpose based on available data on an on-going project:

RQ4. How to make the choice of SRGM more effective?

The answers to these questions are the key to successfully applying SRGMs in industrial settings. Evaluation of long-term predictive power of SRGMs in the automotive domain was done in our earlier work (Rana et al., 2013a). In this paper we extend the analysis by using additional data from two more large organizations engaged in embedded software development but in different application areas (telecom and defense). With the unique setting of large-scale software projects we are able to answer the research questions with higher generalizability. We are also able to make distinctions between the applicability of different SRGMs based on different project attributes, defect inflow profiles and development processes. We further use trend analysis for predicting the shape of the defect inflow for on-going projects – which provides practitioners in industry a framework for selecting and applying SRGMs for supporting decisions of practical significance, such as test resource management and evaluating whether the software product is ready for release.

The rest of the paper is structured as follows: Background for the research and a brief discussion around related works is presented in Section 2. In Section 3, we describe in details the design choices of this study, the data, models, and analysis methods we used. Section 4 presents the results and analysis of data with answers to the research questions. Section 5 presents recommendations for industry to apply SRGMs while conclusions are presented in Section 6.

2. Background and related work

Common terms related to software reliability are defined in IEEE 1633: Recommended practice on software reliability (IEEE Reliability Society, 2008), accordingly:

Software reliability (SR) is (A) the probability that software will not cause the failure of a system for a specified time under specified conditions, or (B) the ability of a program to perform a required function under stated conditions for a stated period of time.
Software reliability model (SRM) is a mathematical expression that specifies the general form of the software failure process as a function of factors such as fault introduction, fault removal, and the operational environment.

IEEE standard 1633 also provides metrics used in reliability modeling and specifies the recommended procedure for software reliability assessment and prediction.

SRMs can be classified as white box and black box models (Ullah et al., 2012; Wang et al., 2006). White box models use source code attributes for making the assessment and predicting the defect proneness of a given software artifact, while black box models use defect inflow data for modeling reliability. Based on the nature of the data in use, white box and black box models are also known as static and dynamic models (Yamada, 2014). Dynamic/black box models are usually referred to as SRGMs and use defect data from development and/or testing phases. The failure or reliability process can be modeled using calendar or execution time. Though the execution-time models have been shown to be more accurate, the calendar-time models are easier to apply and more intuitive to interpret.

Different models are based on different assumptions, which make some models better suited than others for a given process. Musa et al. (1987) showed that various families of models have characteristics that are better suited for certain applications. The same conclusion is also achieved in the study by Goel (1985). Thus, one of the important questions in software reliability engineering has been which models to use and how to apply them (Asad et al., 2004). Khoshgoftaar and Woodcock presented a case study (Khoshgoftaar and Woodcock, 1991) to support the claim that Akaike Information Criteria (AIC), based on the log-likelihood function, can be used to select the best model. Sharma et al. (2010) looked at the model selection problem and proposed a quantitative framework based on the distance-based approach that can be used to rank different models and select the optimal one. Stringfellow and Andrews (2002) proposed an empirical method to select a suitable SRGM for making release decisions during the test phase. They iteratively applied different SRGMs and if a given model passed the proposed criteria, it could be used for making release readiness decisions.

In this study we introduce a new approach for selecting the appropriate SRGM, which is based on the observed defect inflow profile. We use the trend of defect intensity to predict the shape of the full defect profile, which is used to select the appropriate SRGM for a given purpose. We also evaluate if using this strategy leads to better model selection.

SRGMs evaluations within specific industrial domains are limited, though some studies have been reported. Wood (1996) evaluated eight SRGMs on data from industry concluding that defect predictions based on cumulative defect inflow data from development and testing was well correlated with after-release