Information Sciences 423 (2018) 398–411

Contents lists available at ScienceDirect

Information Sciences

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

Comparing the reliability of software systems: A case study on mobile operating systems

Vladimir Ivanov, Alexey Reznik, Giancarlo Succi*

Innopolis University, Universitetskaya Str. 1, Innopolis, 420500, Russian Federation

ARTICLE INFO

Article history:
Received 13 March 2017
Revised 22 August 2017
Accepted 25 August 2017
Available online 21 September 2017

Keywords:
Software reliability
Software quality
Software reliability growth models (SGRM)
GQM

ABSTRACT

Assessment of software reliability is inevitable in modern software production process. Many works aimed at better models for measurement and prediction of reliability of software products. Tens of approaches have been developed and evaluated so far. However, very few works focus on approaches to compare existing systems with respect to reliability. Despite a tremendous importance to practice (and software management area), a complete and sound comparison methodology does not exist. In this paper, we propose a methodology for software reliability comparison. The methodology extensively applies the GQM approach and software reliability growth models. The methodology has been thoroughly evaluated on a case of assessment and comparison of three open source mobile operating systems: Sailfish, Tizen and CyanogenMod.

1. Introduction

Comparing the reliability of software systems is often of paramount importance. Nowadays, problems of software quality assurance and prediction of behavior of software systems have high importance due to the fact that software is included in the most areas of human activity, especially in safety-critical domains (hydrotechnic, transport, chemical industry, warning systems, etc.).

As defined by Lyu [46], software reliability refers to the probability of failure-free software operation for a specified period of time in a specified environment. The reliability of software is typically measured as the number of defects that exist in the source code of the released software product or of failures that happen during its execution [41,72,73].

Research on specific aspects of software reliability modeling has been already performed. Most of works focus on the evaluation and comparison of the software reliability models and pay less attention to the comparison of software products using the developed models. Therefore, there is no well-founded and validated methodology for the effective comparison of reliability of software systems.

The goal of this paper is to propose and validate a methodology for comparing the reliability of software systems. The approach is based on the analysis of the bug reports present in the typical bug tracking systems, nowadays used in most software projects and the validation has been carried out on three industrially used mobile operating systems:

Cyanogen, an Android-like operating system without Google-provided services and with slight modifications to the core of the OS.;

* Corresponding author.
E-mail address: g.succi@innopolis.ru (G. Succi).

http://dx.doi.org/10.1016/j.ins.2017.08.079
0020-0255/

Sailfish, an open source operating system being developed by a Finnish company named Jolla in collaboration with a Russian company named OMP; it is an extension of the old Nokia's Maemo project; Sailfish project consists of two major parts: Mer project (core stack of middleware) and Nemo (Linux distribution for mobile devices)

Tizen, an open source mobile operating system being primarily developed by Samsung. These three projects are in the stage of active development, and have significant size (>50 MLOC). This presents an opportunity to assess the reliability of software based on the engineering processes and their metrics.

The paper is organized as follows. In Section 2 we survey the existing proposals in comparing reliability of software systems. In Section 3 we overview our proposed methodology for the assessment of the reliability of software systems, which is based on the GQM approach (detailed in Section 4) and on Software Reliability Growth Models (SRGM, detailed in Section 2.2); survey the area of software reliability growth model. Section 6 presents description of the datasets that have been used for evaluation. In Sections 7 and 8 we analyze outcomes of the empirical evaluation and provide recommendations relevant for decision making in practice. Section 9 concludes the paper.

2. Background

Comparing the reliability of software systems has been a long standing problem [28] that has permeated the software industry for the last 70 years almost [8,17,24,27]. Various proposals have been devised [65], spanning different levels of the lifecycle, starting from the initial design [6] and covering the entire lifetime [22] and covering different aspects of software systems, from the low levels [2,26,38] networking and intercommunications aspects [32], system integration [24], up to metrics [64] and user interfaces [77], or also considering the whole lifecycle [13,14,33,39,48].

However, there are limited works comparing the reliability of software systems that have been applied or, at least, experimented, with real projects.

2.1. Existing approaches to compare the reliability of software systems

A common goal of a comparative study of reliability is to set up a common approach or a system of criteria to do measurable and reproducible results. Most SRGMs are parametric models with few parameters that may be interpreted and compared in a meaningful way. Therefore, existing methodologies try to find and analyze patterns in software reliability growth and use those patterns in a comparison.

Zhou and Davis [78] reported results of open source software evaluation. They used the Weibull model fitted to bug tracking data. An interesting result is a comparison of open source and closed source projects. Along a development cycle, open source projects demonstrated reliability growth pattern similar to closed source projects. A slightly similar study has been presented by Syed-Mohamad and McBride [68]. Authors use two SRGMs (a concave model and an S-shaped model) to compare two open source products. The main research question: whether open source software has a different defect arrival rate to software developed in-house. Developers of open source software tend to dramatically change source code between subsequent releases in order to meet new expectations such as feature requests and may frequently switch to new technology. Another factor is the large community which easily fulfills new feature requests but rarely do a rigorous quality assurance. This is opposite to results presented in the work of Zhou and Davis [78]. Interesting that both studies interpreted results in a meaningful way. This is a clear evidence for the lack of comprehensive research in the area of methodologies for software reliability comparison. One of the most recent comparison studies is presented by Rossi et al. [63]. Authors aimed at presenting an approach to investigate reliability of open source software projects using several SRGMs. They carefully downloaded and cleaned data from three large open repositories: OpenSuse, Mozilla Firefox and OpenOffice.org and applied eight SRGMs: Goel Okumoto, Goel Okumoto S-shaped, Gompertz, Hossain Dahiya, Logistic, Weibull S-shaped, Weibull more S-shaped and Yamada's model. The comparative study is intended to compare two types of software development (namely open source and closed source), rather than compare software products to each other. Given the difference of the products under the examination, this result is very promising and it is supported in the related study carried out by Li et al. [43].

Overall, the state of the art approaches to comparison of software products reliability are mostly ad-hoc and do not follow a well-founded and validated methodology. Despite the lack of such methodology, there are many case studies in the area of reliability analysis of open-source and closed-source products. The case studies are important for our study, in order to show the whole spectrum of ideas in this filed. Thus, any new comparison methodology should be broad enough to cover major part of the spectrum.

Rahmani et al. [62] compare the goodness-of-fit and prediction capabilities of three reliability models using the failure data of five popular open source software products. Ullah et al. [70] investigate the goodness of fit and prediction capability of eight SRGMs. The comparison involves fifty different failure data sets. Contemporary studies involve additional criteria for evaluation. For instance, in one of the most recent works, Ke et al. [36] use the Parr-curve model with multiple change-points to analyze the consumption of testing-effort.

2.2. Software reliability growth models

The reliability of software systems have been described using multiple approaches, including fuzzy models [29,50,56], granular models [57], program invariants [16] regression analysis [47], Markov Models [42,44], neural networks