International Journal of Embedded and Real-Time Communication Systems
Volume 9 • Issue 2 • July-December 2018

A Survey on Quality Attributes and Quality Models for Embedded Software

Zouheyr Tamrabet, University of Oum El Bouaghi, RELA(CS)2 Laboratory, Oum El Bouaghi, Algeria
Toufik Marir, University of Oum El Bouaghi, RELA(CS)2 Laboratory, Oum El Bouaghi, Algeria
Farid MOKHATI, University of Oum El Bouaghi, RELA(CS)2 Laboratory, Oum El Bouaghi, Algeria

ABSTRACT

This article describes how software quality engineering is an inevitable activity, which must be accomplished during software development process in order to avoid software failures and ensuring its quality. Embedded systems are computer platforms, which require high quality software. Many researchers interested in embedded systems have demonstrated that the quality of the embedded software has a significant effect on the performances of the entire system. In the literature, several works have been emerged from this line of research. The aim of this article is to present a survey of the most important works, which deal with embedded software quality engineering. A comparative study is also given in order to show strengths and weaknesses of each work.

KEYWORDS

Embedded Software Quality Attributes, Embedded Software Quality Models, Embedded Software Quality, Embedded Software, Embedded Systems, Quality Attributes, Quality Models, Software Quality

1. INTRODUCTION

In our daily life, several computer systems are included in the surrounding digital products such as cell phones, digital cameras, home appliances and medical devices. These products are called embedded systems. Generally, the embedded systems are founded to perform specific tasks either in telecommunication, computing, controlling, etc. The main characteristic of these systems is the combination of hardware and software in order to meet the requirement of users. This characteristic makes the application of the traditional development process inadequate. Especially, more attention must be given to the partitioning/integrating phase between hardware components and the intangible components of the software (Jamont, 2005). Furthermore, the quality represents a fundamental query in the development of such systems. In fact, embedded systems consist not only on functional requirements (the functional properties like taking pictures for camera and calling for phone, etc.) but on the non-functional requirements which represent the quality attributes of the developed systems. The quality attributes refer to systems quality aspects like performance, usability, security, etc. (Muhammad et al., 2010). Nevertheless, there is no common consensus on how to identify quality attributes of a system in general and of an embedded system in particular (Sherman, 2008; Wijnstra, 2001; Choi et al., 2008; Jeong & Kim, 2012; Oliveira et al., 2013).

Most embedded systems are critical. Consequently, the development of them with poor quality can produce financial, health, and even life losses. We can refer to the bug discovered in the embedded software of Volkswagen cars which affect the fuel consummation (Nikolaus, 2015). However, mastering embedded system's quality is a hard task. This difficulty is due to the nature of the software

DOI: 10.4018/IJERTCS.2018070101

(abstract, intangible, etc.) on the one hand. On the other hand, it is due to the nature of the embedded systems, especially the integration of the software part in the whole system (mechanic, electronic, etc.) (Wagner, 2013). As a result, software quality engineering for embedded systems, as for any other software system, is an indispensable activity in order to produce secure, safe, functional and reliable systems.

In this paper, we present a survey of the most important works, which deal with embedded software quality engineering. We also give a description of each one and present their strengths and weaknesses through a comparative study.

The remainder of this paper is organized as follows. In section 2, we give a brief overview of embedded systems. Section 3 presents a background on software quality. Section 4 is dedicated to present a state of the art of embedded systems quality engineering. Section 5 presents the most important works dealing with software quality attributes and quality models of embedded systems. Section 6 provides a comparative study between the most important works of the previous section. Some conclusions and future work directions are given in section 7.

2. EMBEDDED SYSTEMS

In spite of their increasingly use, embedded systems have no universally accepted definition. For some, an embedded system is part of a larger system and performs some of the requirements of that system (Carvalho & Meira, 2009). For others, they are considered as computers embedded inside electronic devices, or even as electronic programmable devices integrated in a larger heterogeneous system (Muhammad et al., 2010). The embedded systems are also commonly known as "hidden computer systems." They are dedicated systems for particular tasks, with no standard inputs/outputs and with limited resources (Koopman, 1999). Despite the lack of a standard definition of embedded systems, there is a general consensus about the common characteristics of such systems. Hence, an embedded system is consisting of software integrated in a hardware system. It is developed for specific tasks and it has limited resources.

Obviously, we must consider certain constraints imposed generally during the development of the embedded systems. Especially, constraints like size, weight, cost and energy. Furthermore, embedded systems could have real-time constraints which make them categorized into real-time and non-real-time embedded systems depending on the type of the imposed constraints. In the case of real-time embedded systems, another classification is found (Muhammad et al., 2010): hard real-time, firm real-time, and soft real-time systems, based on their functioning according to certain deadlines. Hard real-time system means that any missed deadline provokes system failure. Firm real-time system tolerates with some missed deadlines. Soft real-time system tolerates with every missed deadline; however, this tolerance decreases the system performance.

Taking into account the strong constraints imposed by the application domains of embedded systems led to the development of specific methods and methodologies to develop such systems. Especially, the design phase which is a critical one in the development cycle of embedded systems. Therefore, the co-design is increasingly used. The co-design allows to design at the same time both the hardware and the software to get the appropriate features. It also permits to delay as max as possible the hardware choices in opposition to the classical approach were the hardware choices are made in first place. Due to the hardware description languages (such as VHDL, VeriLog, etc.) we can synthesize and test digital circuits according to the co-design concept. The co-design includes as phases: Specification, Modeling, Partitioning, Synthesis and Optimization, Validation, Integration, Integration Tests (Jamont, 2005).

As it is previously mentioned, software quality is an important field in software engineering. In fact, the development of software with poor quality can produce catastrophic consequences. Although, the study of software quality has been initiated forty years ago (McCall et al., 1977), this subject still an active research area. Actually, the most promising research trends focus on customizing the

Related Content