Chapter 6

A Comparative Study of Bayesian and Fuzzy Inference Approach to Assess Quality of the Software Using Activity-Based Quality Model

Kawal Jeet
D A V College, India

Nitin Bhatia
D A V College, India

Renu Dhir
Dr. B R Ambedkar National Institute of Technology, India

ABSTRACT

Maintainability is one of the important characteristics of quality of software. It is the measure of efforts needed to modify the software. Large number of subjective techniques has been developed in industry to deal with assessment or prediction of this characteristic. But these techniques generally fail due to their inability to break down maintainability to a level of actual evaluation. They also lack homogeneity in the models thus developed and so fail to take into account the cost factor associated with maintainability. Activity based quality model is found to decompose maintainability to an actual analyzable level. It manages maintainability in terms of software maintenance efforts but it lacks quantitative evaluation of this characteristic. Bayesian approach to deal with this model added quantitative feature but also added crispness to the system developed. In this chapter, the authors propose the use of fuzzy approach to correct the existing Bayesian approach to deal with activity based quality model. A comprehensive comparative study is presented to show the effectiveness of proposed technique.

DOI: 10.4018/978-1-4666-2958-5.ch006
A Comparative Study of Bayesian and Fuzzy Inference Approach to Assess Quality of the Software

INTRODUCTION

Overview and Motivation

According to ISO 9126 model, quality is the totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs (Losavio et al., 2003 and Jung et al., 2004). Assessing or predicting the quality of software is a very challenging task in practice as well as research. To deal with software quality a list of attributes is required to be defined that are appropriate for software. So quality is composed of following six attributes:

1. **Functionality**: It indicates the capability of the software product to provide intended functions.

2. **Reliability**: It indicates the capability of the software product to maintain its level of performance under stated conditions for a stated period of time.

3. **Efficiency**: The ability of the software product to provide appropriate performance under stated conditions, relative to the amount of resources used.

4. **Usability**: The capability of the software product to be understood, learned and used by the user.

5. **Portability**: The capability of the software product to be transferred from one environment to another. The environment may include organizational, hardware or software environment.

6. **Maintainability**: The capability of the software product to be modified. Modifications may include corrections, improvements, or adaptations of the software to changes in the environment and in the requirements and functional specifications.

Comparison between Various Quality Models

Various quality characteristics present in different models Boehm, MacCall, FURPS, ISO9126, Dromey are shown in Table 1.

From Table 1 it can be seen that maintainability is an important characteristic that is being paid attention by almost all the models popular in research and practice (Moses, 2009). Most of the software life cycle cost is consumed by continuous adaptation, extension and bug fixing of existing software and hence the maintainability of software (Pigoski, 1996).

With the maturation of software development practices, software maintainability has become one of the most important concerns of the software industry. The total cost of maintaining a widely used program is typically 40 per cent or more of the cost of developing it (Brooks, 1995).

So it is obvious that any software dependent organization wants to reduce spending for software maintenance activities. This term is most frequently associated with more flexible software and significantly reduced long-term costs. This means the desire for high maintainability is really a desire for low maintenance efforts. Current

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Reliability</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Usability</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Efficiency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Portability</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Maintainability</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Related Content

Formative User-Centered Evaluation of Security Modeling: Results from a Case Study
[www.igi-global.com/article/formative-user-centered-evaluation-security/64192?camid=4v1a](www.igi-global.com/article/formative-user-centered-evaluation-security/64192?camid=4v1a)

Innovative Strategies for Secure Software Development
[www.igi-global.com/chapter/innovative-strategies-secure-software-development/77792?camid=4v1a](www.igi-global.com/chapter/innovative-strategies-secure-software-development/77792?camid=4v1a)

Analysis and Evaluation of Software Artifact Reuse Environments
[www.igi-global.com/article/analysis-and-evaluation-of-software-artifact-reuse-environments/119990?camid=4v1a](www.igi-global.com/article/analysis-and-evaluation-of-software-artifact-reuse-environments/119990?camid=4v1a)

Tools for the Study of the Usual Data Sources found in Libre Software Projects
[www.igi-global.com/chapter/tools-study-usual-data-sources/29442?camid=4v1a](www.igi-global.com/chapter/tools-study-usual-data-sources/29442?camid=4v1a)