Ain Shams University – Faculty of Engineering
Electrical Engineering - Computer and Systems Department

Intelligent Software Quality Management

A Thesis

Submitted in partial fulfillment of the requirements for the

M.Sc. Degree in Electrical Engineering
(Computer and Systems)

Submitted By

Ahmed AbdelMonem Ahmed Hamada

B.Sc. Electrical Engineering
(Computer & Systems Engineering)
Ain Shams University, 2000

Supervised By

Prof. Dr. Hussein I. Shahein
Dr. Mohamed N. Moustafa
Computer and Systems Engineering Department
Faculty of Engineering, Ain Shams University

Cairo 2009

Ain Shams University – Faculty of Engineering
Electrical Engineering - Computer and Systems Department

APPROVAL SHEET

Name Ahmed AbdelMonem Ahmed Hamada

Thesis Intelligent Software Quality Management

Degree Master of Science in Electrical Engineering (Computer and Systems)

Examiners Committee

Name, Title and Affiliation **Signature**

1. **Prof. Dr. Ahmed Mahmoud Darwish** …………………………..
   Minister of State for Administrative Development
   & Professor in the Computer Engineering Dept.
   Faculty of Engineering
   Cairo University

2. **Prof. Dr. Hussein Ismail Shahein** ………………………………
   Professor in the Computer and Systems Engineering Dept.
   Faculty of Engineering
   Ain Shams University

3. **Prof. Dr. Yasser Hisham Dakroury** ………………………………
   Professor in the Computer and Systems Engineering Dept.
   Faculty of Engineering
   Ain Shams University

Date: / / 2009

STATEMENT

This dissertation is submitted to Ain Shams University for the degree of Master of Science in Computer Engineering.

The work included in this thesis was carried out by the author at the Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University.

No part of this thesis has been submitted for a degree or qualification at other university or institution.

Name Ahmed AbdelMonem Ahmed Hamada

Signature

Date / / 2009

ABSTRACT

INTELLIGENT SOFTWARE QUALITY MANAGEMENT

Ahmed AbdelMonem Ahmed Hamada

Master of Science in Electrical Engineering (Computer & Systems Engineering)

Ain Shams University, 2009

It is vital to obtain data so that actions can be taken to improve any computer software performance. Such improvement can be measured in terms of higher quality, increased customer satisfaction and decreased cost of quality.

Different researchers have proposed software quality models to help measure the quality of software products. These models often include metrics for this purpose. Some of the classical and recent models are discussed and analyzed in this work showing the points of strength and weakness of each model type.

To alleviate most of the problems of the previous models, a new comprehensive software quality model – FUPRIMDSO – is proposed and analyzed. The FUPRIMDSO model is not the only contribution, but, a complete solution is discussed throughout the thesis to enable an effective and efficient use of the proposed model. The goal is to help the development team in prioritizing the important metrics while developing the software products according to some inputs from the user and the objectives of the software being developed.

The solution developed is called the Quality model Analysis Program (QAP). Basically, it is a fuzzy system that weights the proposed model attributes according to certain rules.

Our solution enables software project managers to better utilize their resources and take specific actions to better improve the quality of the software produced.

Keywords: QAP, Fuzzy, Quality Model, Software Quality, Measures, FUPRIMDSO.

ACKNOWLEDGMENT

First of all I want to thank ALLAH for everything, and one of these things is supporting me to complete this research that I hope it can be a small contribution in the Software Engineering science.

I wish to thank and to give my deep appreciation to my teacher and advisor Prof. Dr. Hussein Shahein who backed me up and supported the research from the early beginning based on his deep knowledge in the computer engineering field and his deep familiarity with the needs and importance of research.

Also, I want to thank Dr. Mohamed Moustafa for the great effort he did, he supported me with his knowledge and ideas, he added value to my research through technical guidance and support in every time. His continuous directions led to completing this research.

And I want to thank both my supervisors Prof. Dr. Hussein and Dr. Mohamed for lightening to me the importance of having the driver of contributing to the benefit of people.

My appreciation extends to all those who supported and encouraged me to complete this effort, my mother for her support and for encouraging me, to my wife for her patience, effort and support and to my brothers. I would like to send my acknowledgment too to the soul of my father who kept explaining to me the importance of the post graduate studies.

Table Of Contents

List Of Tables

List Of Figures

CHAPTER 1: INTRODUCTION

1.1 Introduction

It is very important to collect data to be able to take managerial and technical actions that can improve the effectiveness and performance. The increase in customer satisfaction, quality improvement and cost of quality reduction can measure such improvement.

Different software models were proposed to help measure the software product quality. The models are built on different metrics to measure the quality of the software. All the current quality models lack a complete set of software quality metrics. The project managers collect huge number of software metrics data to plan for the software development or to measure the quality of the software. The literature review shows a need for a complete solution that supports the project managers in best utilizing their resources and enables the tradeoffs between conflicting quality requirements. And to be able to build this solution and prioritizes the quality metrics, then a quality model that shares all of the software quality attributes need to be implemented. Our proposed software quality model is called the FUPRIMDSO and our proposed quality solution is called the Quality model Analysis Program (QAP).

1.2 Problem Statement

Based on the literature review and from the analysis that was performed on the classical and generic software quality models, there is no comprehensive guideline that can present a common view for the software quality characteristics. In addition to that, new software types have emerged and software development has advanced significantly while the relative importance of various factors has changed and with the spread of the internet, for example, new aspects of security have become more important. Also, there is no approach that can give the project mangers the required insight to prioritize the quality metrics according to their importance and based on the nature of the project.

1.3 Approach

In this thesis, the focus is on proposing a new comprehensive software quality model - FUPRIMDSO - and defining a complete approach which we called the Quality model Analysis Program to enable the effective use of this model in prioritizing the metrics according to importance. The projects will need to collect some input parameters that will be used to feed a fuzzy inference engine to provide the project team with the relevant quality metrics prioritized by importance. These metrics can be used to plan for the software development or measure the quality of the software. The Quality model Analysis Program adopts an intelligent weighting system that uses the Fuzzy logic to prioritize the metrics according to their importance based on the inputs collected from the project managers. This data mining approach gets value out of the large amounts of data and identifies patterns and relationships [1].

1.4 Thesis Organization

In chapter one the basic approach for the intelligent software quality management is discussed and the organization of the thesis is explained.

In Chapter Two: The Classical Quality Assurance Models are discussed; the metrics classifications and Various Quality Techniques are explained.

In Chapter Three: More recent models are discussed and the relationship between these models is analyzed. Also, the new Comprehensive model (FUPRIMDSO) is discussed.

In Chapter Four an overview of the Statistical Surveys and Data Collection methodologies is discussed along with the need for establishing an intelligent system.

In Chapter Five the Software classifications and requirements are explained. The Details of the Survey and building the knowledgebase and the Statistical Survey Optimization are discussed.

In Chapter Six we discuss The Fuzzy Logic and the different techniques for calculating the Defuzzification and the degree of membership.

In Chapter Seven Designing the inference rules, building the fuzzy engine are discussed. Also, we explain the inputs, outputs and the processing of the Quality Analyzer Program.

In Chapter Eight we discuss the results followed by drawing some conclusions and pointing to some Future work initiatives.

CHAPTER 2: QUALITY ASSURANCE MODELS

2.1 Overview

When software is developed, the desired level of quality must be determined. Frequently, during software development, quality is often considered acceptable when the development time has run out or someone in management or marketing decides that the product is ready for the market. This may only mean that all required functions are developed and that the software basically works and doesn't crash.

Quality is subjective, so to have a good quality product that meets various requirements and criteria, various aspects of software quality must be measured or quantified in an as objective a way as possible. To make quality measurable, it must be split into parts which can be measured. These parts can't possibly contain all aspects of quality; they only simplify the quality. This simplification is called a "quality model". Without a quality model systematic quality measurement cannot be made.

Over the years, researchers proposed and successfully brought into practice a considerable number of various quality assurance techniques applicable to software product lines. Yet, such methods as component certification and regression testing are applicable only in late phases of the software lifecycle, where the cost of modification is very high. In order to lower the cost of quality, methods for early evaluation and control for quality have to be employed. Approaches based on quality models cope with this requirement. According to ISO/IEC International Standards [2], the term quality model is defined as "the set of characteristics and relationships between them, which provides the basis for specifying quality requirements and evaluating quality".

Various researchers have produced models, usually taxonomic, of software quality characteristics or attributes that can be useful for discussing, planning, and rating the quality of software products. The models often include metrics to "measure" the degree of each quality attribute the product attains.

Usually these metrics may be applied at any of the product levels. They are not always direct measures of the quality characteristics of the finished product, but may be relevant to the achievement of overall quality.

2.2 Classical Approach- Early Models:

Some of the classical thinking in this area goes back to McCall [3] (1977) and Boehm [4] (1978). Each model may have a different set of attributes at the highest level of the taxonomy, and selection of and definitions for the attributes at all levels may differ. The important point is that the system software requirements define the quality requirements and the definitions of their attributes.

In this section, we will shed light on some of these classical models whereby we will discuss their quality factors, quality attributes, and their characteristics.

2.2.1 McCall's Model

This model was produced by Jim McCall for the US Air Force and it aimed to lessen the gap between developers and users. He mapped the views of the user with the priority of the developer.

Three main perspectives were identified by McCall for characterizing the quality attributes of a software product.

These perspectives are product revision, product transition, and product operations.

In his model McCall identified the below 11 quality factors broken down by the three perspectives afore-mentioned [5].

McCall's Software Quality Factors:

0. Correctness

1. reliability

2. efficiency

3. integrity