Chapter 1.21
Different Views of Software Quality

Bernard Wong
University of Technology Sydney, Australia

ABSTRACT

This chapter examines the different definitions of quality and compares the different models and frameworks for software quality evaluation. It will look at both historical and current literature. The chapter will give special attention to recent research on the Software Evaluation Framework, a framework for software evaluation, which gives the rationale for the choice of characteristics used in software quality evaluation, supplies the underpinning explanation for the multiple views of quality, and describes the areas of motivation behind software quality evaluation. The framework has its theoretical foundations on value-chain models, found in the disciplines of cognitive psychology and consumer research, and introduces the use of cognitive structures as a means of describing the many definitions of quality. The author hopes that this chapter will give researchers and practitioners a better understanding of the different views of software quality, why there are differences, and how to represent these differences.

INTRODUCTION

Adopting an appropriate Quality Assurance philosophy has been often viewed as the means of improving productivity and software quality (Hatton, 1993; Myers, 1993). However unless quality is defined, it is very difficult for an organization to know whether it has achieved quality clearly. To date, this has usually involved conformance to a standard such as AS3563 or ISO9001 or following the Capability Maturity Model of the SEI. The challenge often faced is that one finds as many definitions of quality as writers on the subject. Perhaps, the latter have been remarkably few in number considering the obvious importance of the concept and the frequent appearance of the term quality in everyday language.

Though the topic of software quality has been around for decades, software product quality research is still relatively immature, and today it is still difficult for a user to compare software quality across products. Researchers are still not clear as to what is a good measure of software quality because of the variety of interpretations of the meaning of quality, of the meanings of terms to describe its aspects, of criteria for including or excluding aspects in a model of software, and of the degree to which software development procedures should be included in the definition. A particularly important distinction is between what represents quality for the user and what represents quality for the developer of a software product.

Perceptions of software quality are generally formed on the basis of an array of cues. Most notably, these cues include product characteristics (Boehm et al., 1976; Carpenter & Murine, 1984; Cavano & McCall, 1978; McCall et al., 1977; Kitchenham & Pfleeger, 1996; Kitchenham & Walker, 1986; Sunazuka et al., 1985). The cues are often categorized as either extrinsic or intrinsic to the perceived quality. Simply, intrinsic cues refer to product characteristics that cannot be changed or manipulated without also changing the physical characteristics of the product itself; extrinsic cues are characteristics that are not part of the product (Olson & Jacoby, 1972). Price and brand are thus considered to be extrinsic with respect to product quality.

This chapter examines the different definitions of quality and compares the different models and frameworks for software quality evaluation. This chapter will address both the topics of interest for the information systems community and the software engineering community. It will look at both historical and current literature. The chapter will give special attention to recent research on the Software Evaluation Framework, a framework for software evaluation, which gives the rationale for the choice of characteristics used in software quality evaluation, supplies the underpinning explanation for the multiple views of quality, and describes the areas of motivation behind software quality evaluation. The framework has its theoretical foundations on value-chain models, found in the disciplines of cognitive psychology and consumer research, and introduces the use of cognitive structures as a means of describing the many definitions of quality.

BACKGROUND

Software users today are demanding higher quality than ever before, and many of them are willing to pay a higher price for better quality software products. The issue of software quality has come to the forefront in Europe, the United Kingdom, the United States, and more recently Australia. The quality movement in software is not new. A search of the information systems literature has shown that attempts to achieve quality software have been on-going for many years. Software quality models include the product-based view (Boehm et al., 1976; Carpenter & Murine, 1984; Cavano & McCall, 1978; McCall et al., 1977; Kitchenham & Pfleeger, 1996; Kitchenham & Walker, 1986; Sunazuka et al., 1985), process focused models following a manufacturing-based view (Coallier, 1994; Dowson, 1993; Humphrey, 1988; Ould, 1992; Paulk, 1991), and more recently, techniques and tools to cater for the user-based view (Delen & Rijsenbrij, 1992; Erikkson & McFadden, 1993; Juliff, 1994; Kitchenham, 1987; Kitchenham & Pickard, 1987; Thompsett, 1993; Vidgen et al., 1994). However, the many models and approaches seem to contradict each other at times. Garvin (1984) tries to explain these contradictions by introducing different views of quality. He describes the models as transcendental-based view, product-based view, manufacturing-based view, economic-based view, and user-based view, which we will define later.

As the software market matures, users want to be assured of quality. They no longer accept the claims of the IT department at face value, but expect demonstrations of quality. There is a firm belief that an effective quality system leads to increased productivity and permanently reduced costs, because it enables management