Quality assurance of design support software: review and analysis of the state of the art

E.Z. Opiyo[*], I. Horváth, J.S.M. Vergeest

Design Engineering Department, Delft University of Technology, Landbergstraat 15, 2628 CA, Delft, The Netherlands

Abstract

In this paper, we review the state of the art and practice in the development and quality assurance of design support software (DSS) products. We first study and analyze the processes of development of DSS products, and then review risk-reducing techniques employed in their development. Based on the analysis and the reviews, we propose a comprehensive methodology for pre-implementation testing of DSS, and concisely present its main elements. One of the distinct characteristics of the processes of development of DSS is that they are heavily research oriented. They typically involve formulation of specification requirements, selection or creation of theories, formulation of methods, designing of algorithms, and writing codes for pilot implementations. Often the problems are unclear and the solution concepts are unknown. Usually the conventional software development models and testing techniques are used in the development of DSS. The problem is that they cover only a subset of the DSS development activities and they do not scale to exactly match the needs in these processes. Almost all software development concepts or mental models demand the developers to prepare specifications, then design software and finally write codes. Even tests and reviews are conducted to exclusively ensure consistency and completeness of requirement specifications, designs and codes. A more robust methodology is required to ensure effective development and testing of DSS. What is needed is systematic development and testing of all early implementations and various stakeholders must also be systematically involved in this.
© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Engineering software development; Design support software; Quality assurance; Testing; Prototyping

1. Introduction

Software began to be used as tools for designing products in 1950s [1,2]. Numerous software tools for supporting designers have ever since been built, most of them dedicated for use in the late phases of the design process, namely for drafting, modeling and engineering analysis. Providers are releasing versions after versions of various design software packages in their efforts to improve services they offer. On the other hand, researchers continually develop new theories and computer-oriented methods, and numerous prototype design support software (DSS) tools have been built in various research institutions. Like in the development of other software products, the developers of DSS use systematic development models and testing techniques, and adhere to practices such as usage of good programming styles and safer programming languages to assure quality. Other techniques in the areas such as requirements engineering and software prototyping as well as standards are also out

* Corresponding author. Tel.: +31-15-278-1196;
fax: +31-15-278-1839.
E-mail addresses: e.opiyo@io.tudelft.nl, ezopiyo@hotmail.com
(E.Z. Opiyo).

there, and can be used in the development of DSS. Conceptual design, engineering analysis, detail design, drafting and design documentation are typical design activities supported by DSS [3,4]. We use the phrase 'design support software' or the acronym DSS in this article to refer to any software tools used in industrial or mechanical engineering design. And the process of development of DSS means development of either the elementary functions, modules, or the entire DSS package.

The objectives of this review are three fold: (i) analysis of how the DSS evolve, (ii) reviewing the processes of development and assurance of quality of DSS, and (iii) concluding on a methodology for prototyping and testing of abstract implementations of DSS. The analysis is intended to reveal how DSS emerge, as well as how this differs from the processes of development of conventional software. Having known what it takes to develop DSS, we study various approaches used in search for their quality and eventually propose characteristic features for a new quality improvement methodology.

This paper is organized as follows. In the following Section we analyze the processes of development of DSS systems. We briefly describe how these systems evolve and present various strategies used to assure their quality. Then, in Section 3 we present a review of various concepts used to shape and direct software development activities, and investigate how far they cover aspects of the processes of development of DSS. A review of various verification and validation (V&V) strategies is given in Section 4. In Section 5, we review other measures and activities that help assure quality. These include requirements engineering, involvement of various stakeholders in the development process, handling of knowledge, ensuring compliance to software quality standards, and measurement of software products. Finally, we discuss and give final words about the reviews and present the characteristic features of the projected methodology in Section 6.

2. Analysis of the processes of development of design support software

The DSS products variously known as computer aided design (CAD) systems, computer aided design and manufacturing (CAD/CAM) systems, or computer aided engineering (CAE) systems are typically complex software products and provide a large spectrum of capabilities and applications. To have a good understanding of the process of development and testing of these kinds of software, we study and analyze various development and testing strategies used to assure quality. Based on the reviews, we compile generic features of these processes.

The historical evidences and the overwhelming number of literature shows that design support software generally originate from theories [5]. To develop DSS, the developers need theories, concepts or mental ideas about how the eventual software will work. Functions are usually bounded by principles and theories underlying a given field [1]. Based on theories, various formal methods have been developed and different kinds of algorithms have been designed and implemented. For instance, for geometric modeling, a wide variety of supporting theories such as *Coons patches, Bezier surfaces, Gordon surfaces, Overhauser surfaces, Bicubic patches, and non-uniform rational B-splines (NURBS)* and *solid modeling* [6–8] loomed in different historical timeliness and have been used as foundational theories in the development of various customized or general-purpose applications. Theory-intensive algorithms such as finite-element and finite-difference have been used in engineering analysis. Various concepts have been used in the areas of product data exchange and in geometric modification. Publications show that various development efforts are typically of unique cooperation between industry and academia [7] and often many individuals develop and refine theories and techniques.

Literature on how the DSS are developed is very sparse. Fig. 1 shows a phased general scheme typically used in the development of DSS. What happens is that the developer starts by exploring possible solutions and then experiment with prototypes, select the best alternative and improve it until sign of success shows up, and afterwards implements an initial version of software. Ad hoc strategies are used in accomplishing various activities within the phases. This approach is followed mainly in research-oriented developments as well as by some developers of commercial DSS products.

To give an insight on how development progresses under this scheme, we briefly analyze the approach