1

THE SOFTWARE MAINTENANCE OF LARGE SOFTWARE
SYSTEMS:
MANAGEMENT, METHODS AND TOOLS

K H Bennett

Centre for Software Maintenance
University of Durham
Durham
England
DH1 3LE

Abstract

Software Maintenance is the general name given to the set of activities undertaken on a software system following its release for operational use. Surveys have shown that for many projects, software maintenance consumes the majority of the overall software lifecycle costs, and there are indications that the proportion is increasing. Our inability to cope with software maintenance can also result in a backlog of application modifications.

Sustaining the viability of a large software system over many years at an acceptable quality level is a major challenge. This paper discusses the management of the software maintenance process addressing both the organisational context and the implementation of the management plan in terms of the methods available and the tool support for those methods.

1 Introduction

The development of a software system is complete when the product is delivered to the customer or client, and the software installed and released for operational use. We shall use the term software maintenance to encompass the activities (technical and managerial) that are undertaken on the software subsequently. The term software maintenance has become well established in the computing profession, but in many ways it is an unfortunate choice of words, suggesting parallels or similarities with hardware maintenance. Hardware maintenance is required because of the progressive degradation or wearing out of physical materials whereas software is not subject to
It is recognised that some organisations use terms such as *enhancement* or *system redevelopment* to express activities that we shall classify as software maintenance. Some professionals use software maintenance to refer only to bug-fixing. We justify the broader definition of the term because the management approach, the methods and the tool support are common, yet differ substantially from initial development.

We shall argue in section two that the major concern of software maintenance is with *system evolution*. There is hence a strong relationship between software maintenance and software reliability, since it is not sufficient simply to produce a reliable system in the first place: the reliability of that system must be sustained over a period of many years, in the face of staff turnover, changes in software technology, and new management methods. Military software in particular may have a lifetime of several decades. It is not sensible to argue that software should not change. It seems inevitable that the more successful a software system is, the more pressing will be the demands for it to evolve to meet new requirements.

Software maintenance has traditionally been regarded as of very low status within the computing community. Schneidewind (Schneidewind 1987) in his survey paper concludes that there is a general acknowledgement that the subject is an important area but at the same time that there is a substantial technical neglect in the study of it. In other engineering disciplines, maintenance is not a professional - track career (e.g. motor-car maintenance, lawnmower maintenance etc.) and this is another reason why the term software maintenance is not ideal. In the USA, there are attempts to change the name to *software management* to avoid the unwelcome connotations of maintenance. However in this paper we shall use conventional terminology.

There are a number of indications that the status of software maintenance is improving, and this welcome change is addressed in Section 5. The remainder of the paper is structured as follows. Section 2 defines four categories of software maintenance, and assesses their economic implications. Section 3 addresses the problem of maintaining existing code, which typically is poorly documented and has been developed without the benefit of modern software engineering technology. Section 4 considers the production of software which is easy to maintain. We shall define the ease with which software can be maintained as its *maintainability*.

Finally, Section 6 suggests a number of areas which need further research and development.