Software Maintenance in the New Millennium: Issues and Challenges

Ray Kammer, Director of NIST

As we approached the year 2000, concerns about the Y2K “bug” focused attention on the importance of high quality, reliable software in supporting the economic foundation of the US and world economy. Additionally, security attacks against systems have been escalating, as hackers exploit known software vulnerabilities that have not been patched. Public attention on the quality of software is increasing – and demand is increasing for solutions.

The responsibility of a vendor does not end once a product is released. Likewise, user responsibilities do not end one that software has been installed. If software is not properly maintained, a well documented, cleanly designed product can decay into a poorly documented, difficult to maintain headache. The more difficult it is to maintain a software product, the greater the likelihood that software patches will not be correctly installed by users in a timely manner – and that additional vulnerabilities may be introduced during software maintenance. In a network environment, a bug in system software has ramifications beyond just poor operation or lost functionality. A bug is often an avenue of access pursued by a hostile intruder.

According to Business Week: “More than 75% of the [security] incidents…are the direct result of bugs.” “[M]ore real testing -- not ‘beta testing’ in the marketplace -- will yield programs that are less vulnerable to attack.” (Business Week, February 28, 2000, “Locking Out the Hackers”) A significant number of bugs are introduced during maintenance – but this does not have to be. There is a large body of knowledge and relevant standards to provide guidance for the maintenance process. One challenge is to disseminate that information in a easy-to-use, timely manner – accompanied by enhanced development and use of automated software testing.

Over the past seventeen years since the first software maintenance conference, we have learned a great deal about software maintenance. However, each new generation often repeats the same mistakes and has to relearn the same lessons. This is unfortunately true in many other arenas. We must do a better job of learning from our past mistakes.

Technology is not standing still. We talk not of Vacuum tubes and punch cards but the Internet age. Many software problems can be fixed by downloading and installing software patches. Often systems can be configured so that this happens automatically. Interactive television, with its capability to easily deliver software for entertainment and electronic commerce to a mass audience, is around the corner. Devices for pervasive computing are already in widespread use. What comes next might be based on quantum phenomena. It might be something else. We will not know until it arrives. What we do know is that we will continue to need to maintain high quality software. This most difficult challenge can be addressed, but requires diligence and perseverance. We must integrate what we know with new research to preserve and enhance software that we maintain so that we can all benefit from high quality, reliable, secure software.