



© 2002 The Charles Babbage Institute for the History of Information Technology  
211 Andersen Library, 222 – 21<sup>st</sup> Avenue South, Minneapolis, MN 55455 USA

## A Response to “Preserving Software: Why and How”

Elisabeth Kaplan  
Charles Babbage Institute

**T**he preservation of software for historical research has been a matter of interest for archivists, librarians, and museum professionals who specialize in the history of computing for quite some time now. Throughout the 1980s and 1990s, a series of reports, conferences, and individual efforts by institutions including the Charles Babbage Institute, the Smithsonian Institution, the Computer Museum, and the National Archives brought stakeholders in software history together to try to understand the potential research value of this material and attempt to identify viable approaches for addressing its preservation.

A brief timeline of just a few of these efforts may help to put the topic in perspective.

- In 1986, the Computer Museum (in Boston, the forerunner of the current Computer History Museum in Moffett Field, California) commissioned an electronic records researcher, David Bearman, to conduct a study on the topic; the product was a report, “The Concept of a Software Archive: Planning for a Software Archive.” Bearman’s report articulated the issues that have by now become familiar (selection, preservation, institutional framework, copyright issues, potential users) and concluded that the museum should conduct two additional studies before proceeding along this path.
- In 1988, the Library of Congress established a “Machine-Readable Collections Reading Room” which maintained obsolete hardware in order to serve the needs of anyone interested in studying the design, history, and documentation of software and data files. Among the goals of this pioneering project were to determine best practices for selecting, cataloging, and making accessible microcomputer programs and data files. The project was abandoned within a few years; its low usage could not be balanced with the expense of running and maintaining the hardware.

## Iterations – Kaplan – Response to “Preserving Software”

- In 1990, Columbia University held a two-day symposium titled “Preservation of Microcomputer Software,” which brought together historians, archivists, librarians, computer scientists and examined strategies for funding a software archives, along with issues of selection, description, and preservation.
- In 1993, a “First National Software Archives Conference” was held in Seattle and jointly organized by Apple Computer, Microsoft Corporation, and the Smithsonian Institution. Invitees included representatives from professional associations (the Association of Records Managers and Administrators, the Society of American Archivists, the IEEE); major government agencies (the National Archives, the Library of Congress); a variety of universities (Yale, Stanford, MIT, the Universities of California, Michigan, and Washington); and the software industry (including Adobe, Cray, IBM, Lotus, Novell, Sun, Xerox).

These efforts were all well-conceived; in retrospect, their work was remarkable. Organizers and participants at these meetings recognized, commendably, that any effort to preserve software for historical research would have to be collaborative and consortial. No one person or institution could do it alone. They recognized that stakeholders from industry, academia, and the information professions, had to be involved, and that a wide range of expertise was needed. They also recognized that major institutional backing from all of these spheres would be critical.

Yet despite all of this effort, expertise, and institutional support, and despite some significant smaller advances, no national software archives has been established, and there does not appear to be one on the horizon.

What conclusions can we draw from these efforts? In retrospect, it seems that two fundamental issues could not be satisfactorily addressed. First, while all participants agreed that software history is important, that awareness of it should be raised, and that it must be documented, participants simply could not identify a solid user base of any justifiable proportion. Second, as participants stated over and over again, “preserving software” is much more than an act of accumulation. It means conserving, organizing, researching, cataloging, and presenting the materials in ways that researchers can use. To do otherwise is simply hoarding. And no individual institution or consortium of institutions has been able to balance these two issues.

Since then, institutions such as the Smithsonian, the Charles Babbage Institute, and Stanford University have focused efforts on collecting not the software itself, but its documentary record—which may include catalogs and directories, personal papers, business records, oral histories and audiovisual materials. All of these documentary manifestations can be stored, described, and made available to scholars according to established standards—not cheaply, but with the normal expenses associated with the archival enterprise.

## Iterations – Kaplan – Response to “Preserving Software”

In a recent issue of *Annals of the History of Computing*, David Grier summarized a discussion that had occurred at “Mapping the History of Computing—Software Issues,” an international conference held in Paderborn, Germany, at the Heinz Nixdorf Museums Forum in 2000. “Exactly how does one go about preserving software?” Grier asked.

To operate software . . . one needs an operational machine. To keep some of the old computers is beyond the scope of most museums’ capabilities and beyond the interest of all but a few. Equally impracticable is the task of writing software simulations—programs that would allow old programs to run on new machines.

Grier, a noted historian computing himself, continued that “software code, though occasionally interesting, will eventually be of use only to the most dedicated scholar who masters the arcana of languages long forgotten and machines unable to function.” For the historian, Grier concluded, preserving the *documentation* of software development “is an imperfect—but thus far the only economically viable—solution.” Given the source of this statement,<sup>†</sup> and after examining the effort that has gone before, it is hard to justify more.

<sup>†</sup>*IEEE Annals of the History of Computing*, vol. 23, no. 2 (April-June 2001), p. 68.

*Editor’s Note: The revised transcript of “Mapping the History of Computing—Software Issues,” an international conference held in Paderborn, Germany, at the Heinz Nixdorf MuseumsForum in 2000 has just been published: Ulf Hashagen, Reinhard Keil-Slawik, and Arthur Norberg, eds., History of Computing: Software Issues (Berlin: Springer-Verlag, 2002).*

Elisabeth Kaplan, “A Response to ‘Preserving Software: Why and How,’” *Iterations: An Interdisciplinary Journal of Software History* 1 (September 13, 2002): 1-3.