

Subject: Preserving classic software products

From: "Booch, Grady" [REDACTED]

Date: 4/12/2002, 10:01 PM

To: "paul.mcjones@acm.org" <paul.mcjones@acm.org>

I'm seeking your help in preserving classic software products before they are forever lost to future generations.

As Bjarne Stroustrup, the inventor of C++, has observed, "our civilization runs on computers." There are a number of organizations - most notably the [Computer Museum History Center](#) - dedicated to preserving classic hardware, but it occurs to me (and others) that we should also preserve classic software as well.

There are a number of reasons to act now. Although many of the authors of such systems are still alive and many others may have the source code or design documents for these systems collecting dust in their offices or garages, time is our enemy and over time, these artifacts will become irretrievable. By the way, notice that I said "source code" and "design documents." Executables are interesting only insofar as we have the machines upon which to run them (and preserving running hardware is an entirely different issue), but the code and designs that these executables manifest tell us much about the state of the software practice, the minds of their inventors, and the technical, social, and economic forces that shaped these products in their time. Preserving such artifacts for future generations is certainly a valuable historical curiosity, but they also offer a statement of prior art (relevant to the issues of proving/disproving software patents), the evolution of software architecture (and how they were products of their time), and the creation of architectural patterns (patterns are common solutions to common problems, and such patterns have emerged for particular domains and development cultures). Preserving such artifacts also provides the raw materials for future generations of software archeologists, historians, and software developers (the latter who can learn from the past regarding what worked and what didn't as well as what was brilliant and what was an utter failure).

What constitutes a classic software product? I can only offer the subjective criteria that such a product is one that made a seminal technical, economic, or social difference. Lisp 1.5, the Mark I Time Sharing System, Smalltalk-72, Dijkstra's THE operating system, Unix/Linux, and EMACS are all examples of technically significant products. DOS, Windows 3.1, VisiCalc, and the original Mac OS all had significant economic impact. Space War, Pong, Zork, sendmail, IM, and the original Netscape browser all wove themselves into the fabric of society.

I'm not affiliated with the Computer Museum, although, from my discussions with them, my intent is to initiate this effort but to deliver up all such artifacts to them for their preservation and care. I expect to be creative - some of you have or may know of people who have dusty card decks, listings, or disks that we'd want to scan, capture and otherwise preserve, while certain companies may have source code locked up in their vaults as precious intellectual property yet who might be open to the suggestion of putting that source code into some secure escrow so that 50 or even 100 years from now, at which time there would be zero value in that IP, we could open a time capsule of software.

You are receiving this email because you are an important contributor to the theory and/or practice of software. I've emailed about 500 such folks, ranging from Turing Award winners to CTOs/CEOs, and researchers. I apologize if I've spammed you, but I know of no better way to begin this effort than to connect with those people who have helped invent the present and will continue to invent the future.

To draw this message to a close, I'm only asking two things of you. At the very least, could you respond to the first? If you are so able, some help on the second would be of enormous value.

First, tell me what's in your top ten list of classic software products. What software artifacts would you want to preserve for future generations, products that made a difference in shaping the software industry?

Second, please let me know if you have access to any such artifacts or if you know of colleagues who do. I'm aware of a few related efforts to keep old software running, and if you know of other such activities, please help me get in touch with

their organizers. I'll happily absorb card decks, listings, documents, napkins full of sketches, and the like. If your company owns the rights to some of these artifacts, let's enter into a serious dialog about how we might put those artifacts into secure escrow to save them before your company discards them by intention, neglect, or merger/economic demise. If you are or know of the original architects of these classic software products, let's also talk about capturing an oral history of their creation.

Thank you for your time in listening to my ranting. Personally, I'm doing this because the software industry has been good to me, and I simply want to give back to those who will drive our industry in the future.

Grady Booch

Chief Scientist

[Rational Software Corporation](#)

