The Software Archive

Proposal Outline

- Motivation
    - Our civilization is being transformed to one that runs on software. It is of critical importance to all of us.
    - Software is a form of literature and has intellectual value to readers.
    - Preserving such artifacts provides raw materials for future generations of software archeologists, historians and software developers. They can learn from the past regarding
        - What worked and what didn't
        - What was brilliant and what was a failure
        - How the technology of software evolved
        - The history and development of fundamental software concepts and architectures, including:
            - Data structure and algorithms
            - Performance and other tradeoffs
            - Use and evolution of programming languages
            - Coding styles, idioms, and their evolution
            - Influences from other disciplines and genres
    - Code and designs that this software manifests tells us much about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped these products in their time
    - We must act now because:
        - Many authors of seminal systems are still alive but elderly
        - Many participants may have the source code or design documents for these systems collecting dust in their offices or garages
        - Time is our enemy; over time these artifacts will become lost forever
    - No one else is doing this. No comprehensive and intentional activity has yet been undertaken to preserve the artifacts of software history
- Goals
    - We wish to create a permanent repository of the world's important software and the stories behind it. Objects to be collected include software, digital and print documents, photographs, video, and ephemeral items.
    - The primary goal of this project is preservation.
        - Interpretations and analyses will use this material, but they will be done as part of other projects.
        - Exhibits will use this material, but they will be developed as part of other projects.
    - The emphasis is on the engineering and scientific contributions, but we also include documentation of commercial and societal consequences.

- We give priority to fragile objects and elderly people because they are at risk of loss.
- We collect collateral information and objects that supply context.

- Audience:
  - Historians who want to understand the development of the technology
  - Software analysts who want to understand the architecture of programs and how it, and the design process, has changed
  - Code warriors who want to read programs that others -- pioneers, heroes, competitors, friends -- have written
  - Software engineering researchers who want to derive and analyze statistics from a broad base of diverse source code
  - IP researchers who are searching for prior-art evidence
  - Patent and copyright owners who are searching for IP violations
  - Posterity: people not yet born who will do things we can't imagine with this archive

- Scope
  - We call here all software creations "programs", but we mean software broadly defined. We include: programs, systems, applications, tools, etc.
  - We include noteworthy languages, compilers, operating systems, key applications in communications, graphics, design, simulations, spreadsheets, word processing, email, notable AI systems, embedded systems, games, systems for transactions, finances and accounting, software engineering, manufacturing, education and other socially interesting applications.

- Mechanism
  - The appendix contains a list of objects which could be collected for the programs. For most programs we will collect only a small number of the objects.
  - This is primarily a web-based collection for which paper and magnetic/optical data media are considered an intermediate form that must be scanned or converted to be useful to the project. Whether the originals are kept is a decision of the curatorial staff.
  - Related physical ephemera (T-shirts, coffee cups, hats, etc.) will be added to the physical artifact collection of the Computer History Museum.
  - Conceptually we are filling in a matrix where the columns are the programs and the rows are the objects to collect for those programs
    - In general the matrix will be sparsely filled
    - Some columns -- for particularly important programs -- will be well-populated.
  - This is a distributed effort that can only succeed by involving hundreds or thousands of contributors.
  - The long-term project targets are:
    - 100 "critically important" programs that are proactively targeted for complete archiving with as many of the objects as possible, including newly-done interviews of participants
    - 1000 "important" programs that are proactively solicited (by announcements and by email sent to known participants) and reactively collected with as many objects as are submitted.

        - 10,000 programs that are reactively collected with probably little more than source code, a couple of documents, and some commentary from the contributor

- Filter

    - We want to be as inclusive as possible, subject to:
        - resource limitations such as people time for processing, and disk space
        - the desire to keep the signal-to-noise ratio reasonably high
    - Because of the above, we cannot accept everything
    - But want the filter to allow – in fact, encourage – more than just the conventionally-defined "important" software. We need to be unbiased with respect to:
        - Type of software
        - Programming language
        - Approach or basic philosophy
        - Identity or nature of the developer(s)
        - Success or failure of the software
        - Country or region of origin
        - Purpose of the software
        - Architecture, design approach, or development methodology
        - Time period in which the software was developed
    - One suggested filter mechanism is:
        1. Establish a "software archive acquisition committee" of 10-20 judges who participate by email.
        2. Establish criteria for acceptance which permits a liberal number of "excuses" to collect a program, some of which are quite subjective:
            - It is the first of its kind
            - It is an important example of a genre
            - It was successful (had >100 users over its lifetime)
            - It is the first or important example of a technique, algorithm, or style
            - It is interestingly unique
            - It is beautiful
            - It had an important consequence
            - It was written by someone important
            - It was used by someone important
        3. Establish a web form for submission which includes
            - A description of the program
            - Which criteria from the list it meets, in the submitter's opinion
            - Which objects are available, now and possibly in the future
            - Evidence of authenticity (provenance) of the objects being submitted
                - *We need to be concerned about forgeries. The need for prior-art patent evidence provides ample motivation for nefarious acts.*
        4. Establish an ongoing process:
            - The committee reviews the submissions in emailed batches
            - Committee members have 7 days to vote yes/no on each submission
            - Any submission that gets 3 or more "yes' votes is accepted

- The registrar is notified of the decision and enables transfer of the objects from the contributor to the archive

- Accessibility
  - We provide no special index structure other than a top-level table of contents.
  - We depend on a Google engine or equivalent to facilitate searches of the entire archive.
  - We provide no personal assistance for finding objects or using the archive.
  - There is no charge for using the archive for non-commercial use.
  - Limitations on the use of material from the archive may be required by law. We may need an "I Agree" gateway agreement to communicate and get acceptance of the general rules.
  - All aspects of the collection process should be open to public scrutiny
    - The identities of the participants
    - The principles and practices of collection
    - The procedures and limitation on fair use
    - Descriptions of all collected items

- IP issues
  - We know that copyright ownership for many of the programs will be questionable
  - We will defend making archival copies as a library under the "Fair Use" doctrine
  - We will make submissions as publicly available as we are permitted by law.
  - There will be a mechanism to deny public access to objects where necessary. We will turn on the "deny" bit in a variety of circumstances:
    - Request of the contributor
    - Request of the copyright owner
    - Likelihood that the copyright owner, if known, would ask to deny public access
    - Likelihood of future commercial value which accessibility would inhibit

- Implementation
  - A full-time software archivist/registrar will monitor submissions and will communicate with contributors and users.
  - A part-time IT person will manage the repository and the web entry to it.
  - Equipment needed is: <TBD>
  - Overhead and management will be provided by the Computer History Museum.

- Marketing and awareness
  - We need to communicate information about this project to the various potential stakeholders.
  - We need to encourage submissions and participation.

Appendix

Objects to collect to preserve software

This is an inclusive list of objects to collect in order to preserve the history of a particular piece of software. ("Software" or "program" as used here generically includes a program, an application, a system, etc.)

This is a taxonomy. No priority or importance should be inferred from position on this list.

For no piece of software will all of these objects be collected. For some important software we will try to collect as many as possible. For most software only a few of the objects will be collected.

Each object must be collected with information about its provenance so that historians and researchers can gauge its accuracy and authenticity. We must be aware that in some cases there could be motivation for the contribution of falsified or counterfeit objects.

- The program
    - Copies of the source code intended to be read by people
        - Machine-readable when possible, otherwise on paper which will be scanned
    - Compilable copies of source code.
        - Presumes that the original software development environment is available or can be simulated
        - Depending on the complexity, that also may require collecting
            - Macro libraries, include files, precompiled libraries, etc.
            - Software development tools and files ("make", "lib", "link", etc.)
            - Subroutines, granularity of codes
    - Ready-to-run object code
        - Presumes that the original execution environment is available or can be simulated
    - Packaged or distributed versions of the program
- Collateral materials about the program
    - Documentation
        - User manuals
        - Installation manuals
        - Logic manuals and flowcharts
        - Development notes that elucidate the design process
        - Books and research papers
        - Email Exchanges
        - Contract Materials

- Marketing material
  - Trade show and sales collateral
  - Advertisements
  - Press releases
- Exhibitable objects
  - Card decks, listings
  - Buttons, T-shirts, coffee cups, etc.

- The People
  - The developer's experience
    - Includes architects, designers, implementers, managers and marketers
    - Interviews: audio and/or video, with searchable transcripts
    - Photos and videos of the people and the environment
    - Personal papers and email
    - Written reminiscences, either contemporaneous or retrospective
    - Identities and roles of all participants
    - Biographies of key people
    - Timeline history of milestones and releases; the product's lifecycle and life history
    - Critical assessments of the software
  - The user's experience
    - Videotapes and photos of the software in operation
    - Reviews and analyses by users and competitors
    - Interviews with early or important users: audio and/or video, with searchable transcripts
    - Profiles of typical users
  - The business experience
    - Published reviews
    - Information about the companies involved in development, marketing, sales and support
    - Sales history
    - Competitive environment

Open Issues:

- How do we identify the fundamental intellectual ideas in a particular program? What do we collect to document the evolution of those ideas?
- How do we preserve the history of important concepts like "stack" or "B-tree" or LR(k) parsing?

Change log

10/24/2003	L. Shustek	Original, inspired by the 10/17/2003 Grady Booch workshop
11/30/2003	L. Shustek	Minor changes and additions