Selecting software documentation standards

Whenever a group of people starts working together, its members establish a set of communication customs or conventions. When the work is short in duration and/or the group is small, these customs and conventions usually remain informal and no one bothers to write them down. But, if the work extends over more than three months and/or the group exceeds five people, there usually will be a push by at least one individual in the group to begin writing down or formalizing their customs and conventions.

We call written customs and conventions standards. Most standards are about software information or documents and are known as software documentation standards.

At first, writing one's own standards seems like a good idea, but soon the writing becomes tedious and sometimes overwhelming. Inevitably, someone will suggest that they look outside their group for existing standards. After all, it is easier to rip apart someone else's writing than to do it yourself.

Software documentation standards are usually the first standards the group will seek. A quick survey will reveal that standards do indeed already exist. In fact they abound. The group may even find that some standards are in place on a project down the hall. Probably company headquarters can tell them of more documentation standards within their own company. It will also come to light that the US Army, Navy, Air Force, Department of Defense, Federal Aviation Administration, and the National Bureau of Standards can provide sets of documentation standards. Of course, the IEEE documentation standards will be prominent on such a list.

Now the group's problem shifts from creating to selecting software documentation standards. How do we select a set of software documentation standards that will speed communications in our group and that will serve us for a long time?

In selecting documentation standards, many factors should be considered.

- Are standards required by customers?
- Are there company management policies that the standards must support?
- Are group members employing techniques that should be supported by standards?
- Are metrics available to support the standards?
- Are tools available to support the standards?
- Is training available to support the standards?
- How will advancing technology affect use of the standards?
- How much will it cost to obtain and tailor the standards?

IEEE Unix Standards Committee sets dates

At the January meeting of the IEEE P1003 Working Group, the dates for developing and balloting the first document were established. The meeting followed the /usr/group Standards Committee meeting, where it was agreed that the two groups would merge under the auspices of the IEEE. The IEEE effort will be based on the adopted 1984 /usr/group Standard (available for $15.00 from /usr/group Standards Committee, 4655 Old Ironsides Dr., Suite 200, Santa Clara, CA 95050).

The milestones for the IEEE effort are:
- Final draft review, June 1985
- Balloting cycle, July-October 1985
- Final approval, December 1985
- Published, Quarter 1 1986

The working group meetings are scheduled as follows:
- April 17-19, Boston, Massachusetts
- June 9-11, Portland, Oregon
- September 9-13, Denver, Colorado

To join the working group or balloting group contact James Isaak, Chair IEEE P1003, Charles River Data Systems, Inc., 983 Concord St., Framingham, MA 01701.
• How much will it cost to follow the standards (that is, create and maintain software documentation in accordance with the standards)?

All of the factors on this list are important considerations for the successful implementation of a set of software documentation standards, but the last question on the list regarding the cost to follow the standards will be the focus of our discussion here.

The costs of creating and maintaining software documentation are easily understood when they are partitioned into categories. Before we deal with the cost categories, though, we should note that the cost of creating information (for example, making design decisions) often is mistakenly considered a documentation cost. We have deliberately excluded it from our list because information must be created whether or not it is documented, if a product is to be produced.

**Recording information.** Recording information occurs when a developer sets down information on paper or magnetic tape or on any other medium in such a way that the information is retrievable. If the recording process is different from the creating process, recording becomes a software development cost. This is the cost developers and project managers fear most. But, if the recording and creating processes occur simultaneously (as in the case of using an on-line tool), the cost of recording can be avoided altogether.

**Storing information.** Storing information amounts to keeping information safe and organized when nobody is using it. Information may be stored on-line or in off-line libraries. The cost of storing information should account for the media on which the information is stored, the facilities where the media is kept, and the staff time necessary to maintain libraries.

**Retrieving information.** Retrieving involves pulling information from an on-line library to a terminal screen or checking a document out of an off-line library. This cost usually is insignificant and most people ignore it.

**Producing documents.** Producing a document may involve merely printing it from a computer at minimal cost or having it reproduced in a more complex operation normally involving a print or graphics shop. If print or graphics shop work is required, the cost should be quoted and included in the official project budget.

**Using information.** The cost of using information is often overlooked. Information is read and analyzed by the technical staff for the purpose of building enhancing or repairing a product.

Three conditions drive the cost of using information. The most expensive way of using information occurs when some documentation is available but is very flawed or out of date. The staff will work with the documents until frustration gets the best of them. Then they will resort to the product (source code) as the only reliable source of information. A less expensive but still quite costly way of using information occurs when no documentation is present. In this case the staff must start work with the source code. The least costly situation is established by having accurate and timely software documentation at hand. Under these conditions the staff may quickly determine what needs to be changed in a product.

Software documentation standards are frequently selected or rejected solely on the basis of cost. A lot of people controlling budgets don't know what all the costs associated with software documentation standards really are or aren't. One way to help groups select cost effective documentation standards is to partition carefully the costs of adhering to the standards, so everyone can see all related expenses clearly. (Remember the cost of not documenting.)

This time we've concentrated on cost as a factor in selecting software documentation standards. It is by no means the only important factor. In later issues we'll hit other criteria.

---

**P1016:** A Guide for Software Design Descriptions: H. Jack Barnard, Mail Stop 1D30, AT&T Information Systems Labs, 11900 N. Pecos St., Denver, CO 80234; (303) 538-3976.

**P1028:** A Standard for Software Reviews and Audits: Charles P. Hollocker, Western Electric, 901 Rolling Dr., Lisle, IL 60532; (312) 979-4137.

**P1042:** A Guide for Software Configuration Management: Richard Van Tillburg, Hughes Aircraft Corp., Bldg. 618, MS B209, PO Box 3310, Fullerton, CA 92634; (714) 732-2307.

**P1044:** A Standard Classification for Software Errors, Faults, and Failures: Dick Evans, TRW, Mail Stop 144-1065, One Space Park, Redondo Beach, CA 90728; (213) 535-5448.

**P1045:** A Standard for Software Productivity Metrics: Eleanor Antreassian, Computer Sciences Corp., 99 Westville Ave., Caldwell, NJ 07006; (201) 226-1574.


**A Standard for Software Quality Metrics:** Norman F. Schneidewind, Professor, Dept. RSA CS, Code 54SS, Naval Postgraduate School, Monterey, CA 93940; (408) 646-2719/3211.

**A Standard for User Documentation:** Christopher Cooke, Martin Marietta Aerospace, Mall Station 98, 103 Chesapeake Park Plaza, Baltimore, MD 21220; (301) 338-5644.

**A Guide for Software Verification and Validation:** Jerry Mersky, Logicon, Inc., 255 West Fifth St., San Pedro, CA 90731; (213) 831-0631.

**A Standard for Software Development Plans:** Richard Thayer, CS Dept., California State University, 6000 J St., Sacramento, CA 95819; (916) 454-6834.

**A Guide for Software Maintenance:** David Cargo, Honeywell: MN 26-3331, Honeywell Plaza South, Minneapolis, MN 55408; (612) 870-2277.

**A Standard for the Software Life Cycle Process:** David Schultz, Computer Sciences Corp., 4600 Powder Mill Rd., Beltsville, MD 20705; (301) 937-0760.