Getting the most out of new tools

Robert M. Poston, Software Standards Editor

Pop-up menus, rainbow colors in multiple, overlaid windows, graphic designs playing on the screen — such are the attractions of new software development tools. No wonder so many of us are enticed by new products and technologies.

Salespeople thrill customers with new-product excitement. But we've got to get past our fascinations and down to the serious business of producing software.

Is this tool just a toy, or will it really help my company for a long time? Is this tool the best one for my organization, or should we shop more? How do we get the paybacks the marketing representative promises?

Tool or toy? People in software development are constantly pushed to produce more. That means they need something to relieve them of work or, at least, make their work easier and faster.

A superior tool provides a substitute for human labor. It will do some work for its user. An average tool will make work easier, but the user still has to do the work himself. And, always, a good tool makes doing the right thing the easiest thing to do.

A toy makes doing the job look appealing, but it doesn't reduce the user's work. In fact, using it may actually add time to the job.

One toy I recently tried let me draw dataflow diagrams and structured design charts on a graphics screen. The drawing was fun. The colors were bright. The pictures on the screen were pretty. Too bad that it took more time to position and size the graphics symbols than it did to illustrate the same information the old way on my text editor.

Selection criteria. Specifying selection criteria for a tool that is going to be used for more than a year requires a lot of thought — thought not only about how the tool works but about the environment where the tool must operate.

What hardware and software tools are already in place? When will other tools be added? How do people in the shop develop software now?

That leads us to another important question. Will the new tool improve the way our shop works, or will we have to change how our shop works to use the new tool?

I believe a tool should do work the way you want it to. You should not have to change the way you do things to use the tool. Who's the master, anyway?

Standards must be considered whenever we buy new tools. Standards may be very simple and informal, like conventions and customs that have evolved in an organization to help people work together. They may be professional standards such as those developed by the IEEE and ACM, or they may be military standards.

In all cases, when a new software tool is brought into an environment where it does not conform to prevailing standards, someone must force the tool to fit the standards or else someone must change the standards.

A tool that makes it easy to produce dataflow diagrams has little value if structured systems analysis is not a standard technique in the shop.

The message, then, is to keep standards on your list of selection criteria or pay the price of making adjustments later.

Now that we have a pretty good idea how a tool would fit into our shop, let's nail down the operational features of a top-quality tool.

Potential users should ask:

1. Does this product do precisely what we need?
2. Does it perform fast enough?
3. How long will it take to learn to use this tool effectively?
4. Is this tool easy to use?
5. How often does the tool fail?
6. How fast can we get a workaround or fix for a problem?
7. Can input to this tool be output from another tool, and, conversely, can output from this tool be input to another?

Payback. After we have selected the tool that seems to suit our organization best, how do we realize the promised return on our investment?

Most marketing people from reputable companies make reasonable claims about the benefits that can be derived from using their tools. But, they usually tell us that benefits are maximized when the tools are fully used. We rarely understand the effect of that idea.

Generally, tools are not fully used for two reasons: users seldom become completely proficient with a tool, and many people aren't qualified to perform a job manually, let alone direct a tool to do it.

We all know power users who can wring the last drop of performance out of a tool. The majority of us don't aspire to be power users, though — we just want to learn enough to get our jobs done. Some people will learn only the bare minimum.

Since there are only a few power users in most groups, we probably are not going to get the maximum usage or savings the salesperson talked about.

The problem of unqualified users is a serious one. "Garbage in, garbage out" still applies.

A tool to speed up complex, high-level work such as systems analysis requires a sophisticated, experienced user — one who understands the concepts underlying systems analysis. A good tool can make a competent analyst very fast, but no tool alone is going to make an inexperienced person into a productive systems analyst.

Maximizing payback, then, means getting more users more proficient as quickly as possible.

People should be brought up to speed in necessary methodological techniques and trained to use the tool just before the technique/tool is applied to a real project. During the first projects, tool and technique experts should be available to the development team to answer questions as they come up.

You know that technical guru in your organization who everybody goes to with questions? He's a key communicator. He's an ideal person to have fielding questions about the new tool (especially if he was in on the selection and now promotes the use of the tool).

Have the development team follow through with evaluations. How much money is this tool really saving? No evaluations probably mean the tool...
doesn't appear to be saving money, and managers are laying low for a while.

Rethinking standards. Tools and standards go hand in hand. Right now the IEEE has two kinds of standards: standards for products and standards for processes.

Standards for products remain relatively static, and nearly any tool can have its output formatted to fit product standards. The output, text or graphic, can be recorded according to standards (or guides) such as ANSI/IEEE 830, A Guide for Software Requirements Specifications, and ANSI/IEEE 829, A Standard for Software Test Documentation.

Such is not the case when it comes to process standards. Tools may be used to support processes. They also change how we perform processes. In fact, some tools such as program generators automate a process and take it out of human hands entirely.

New tools are announced daily. We'll have to speed up the IEEE standards development cycle (two years plus on the average) tremendously to create a process standard that won't be made obsolete by some new tool before we finish developing the standard. Is continued development of process standards worth the attempt in view of such a dynamic industry? What alternatives do we have?

Keep in touch. I wish all of you a bright and prosperous new year. I am sorry, though, that many of you expect problems in getting funding for travel to conferences, meetings, and seminars in 1987.

A good way to stay in touch with your profession at home is to continue to read IEEE publications. Perhaps you have a certain topic you'd like to read about in the Software Standards department. Maybe you have a suggestion or comment for me. Use the reader service card in the back of this magazine. It's an easy way to get word to me.

---

**Bulletin board**

**Early information**

For overseas readers and others requiring advance information, please note that the next IEEE Standards conference is planned for fall 1987 in the Washington, D.C., area. Roger Martin is chairing the event. He will be providing us details as they develop.

**Standard 1084 published**

ANSI/IEEE Std 1084-1986, Glossary of Mathematics of Computing Technology, can be ordered now by mail from the IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854.

Please request Standard 1084-1986, stock number SH10595. The single copy price is $8.00 (10 percent discount for IEEE members) plus a $2.00 shipping/handling fee. Major credit cards are accepted and quantity discounts are available. Please call (201) 981-0060 for more information.

**SCM guide balloting**

The Guide for Software Configuration Management, IEEE Project 1042, was distributed by the Software Engineering Standards Subcommittee for ballot last November. The balloting period ended Dec. 14, and results of that vote will be reported soon.

A significant feature of this guide is the inclusion of four detailed appendixes that illustrate the effective practice of SCM in a wide variety of projects. They address critical software development, prototyping, support software maintenance, and product-line life cycle.

**Unix interface definition**

X/Open, a group of 10 major computer companies, has adopted the first commercial internationalization interface definition for Unix. The interfaces are derived from Hewlett-Packard's interfaces for its natural-language support system.

The new definition will let Unix systems handle eight-bit ASCII character codes. Unix systems have been based on the seven-bit ASCII character code set. The eight-bit codes are the minimum to support the major European languages' special symbols and accented characters. The design is open-ended and may be extended to cope with 16-bit codes, which would support Asian languages such as Chinese and Japanese.

**Posix efforts boosted**

The IEEE 1003 Posix working groups reported significant progress in their meetings in late September in Palo Alto, California.

The efforts were boosted after the National Bureau of Standards recommended Posix as the basis for a federal information processing standard. Also, the ISO's US Technical Advisory Group for Technical Committee 97 tentatively recommended that the Posix effort be the basis for an ISO work item to develop an international standard for an operating system environment based on the Unix model. (Formal recommendation is expected early this year.)

Details on what the three Posix subgroups accomplished are available from Jim Isaak, Charles River Data Systems, 983 Concord St., Framingham, MA 01701.

**Comment period for C**

X3, the Accreditation Standards Committee on Information Processing Systems, announced a four-month public review and comment period on the draft proposal for American National Standard X3.159-198x. The standard specifies the form and establishes the interpretation of programs written in C. The comment period ends March 7.

Copies of the draft can be obtained for $65 from Global Engineering Documents, Inc., by calling (800) 854-7179.

**Data structure standard OKed**

The American National Standards Institute has approved a standard for a mechanism that lets data structures be easily moved among different computer systems, regardless of make or medium. ANSI/ISO 8211-1985 specifies medium- and system-independent file and data-record formats for information interchange.

It defines a data-descriptive file that allows interchange with minimal external description, so a sender can preserve structure information and convey it to the receiver, who can then remap the information without losing structure or content.

Copies of the standard are available for $9 from ANSI Sales Dept., 1430 Broadway, New York, NY 10018.