CASE: How real is it?

Janet Butler, Applied Computer Research

"The future of computing is a battle with complexity," argued James Martin, a Pulitzer Prize finalist and author of more than 50 books on computers and communications technology, at the ShowCASE conference sponsored by Washington University and held Sept. 1-2 in St. Louis. "Systems like SDI [Strategic Defense Initiative] are enormously complex and can't be built manually. We need very fast and efficient software development. [But] we're still writing software with pencils and plastic templates," he said, "Programmers need to catch up with engineers and use software tools."

Ken Orr, a leading CASE consultant, said, "We haven't automated the software-development process before because we didn't know how. We could do it intellectually, but it wasn't converted into procedures." But it must be done, he said, "because it's a problem that has to be solved. There are too few professionals capable of building the systems we need. Problems of size and complexity are driving us to use CASE."

But, Orr said, CASE threatens middle managers, good technicians who don't see themselves as managers and who don't know how to manage the new technology. Vaughn Merlyn, an application-development automation consultant, agreed. While software-engineering tools may be bought for one project, they are rarely used on others. "There's a failure to institutionalize the technology," he said.

If CASE is to be the system-development environment of the future, what will it be like and what will it cost? Capers Jones, a software-productivity expert, described his vision of a system in 1995. The configuration would include mainframe databases, minicomputer file servers, primary workstations, portable workstations, home computers, and documentation computers. This system would have more than 100 integrated tools and more than 20 expert systems, plus standard data-interchange and data-conversion utilities. The CASE software would be partitioned into management, technical-support, and physical-environments areas. He estimated it would cost $83.5 million.

What benefits would such an expensive system offer? Jones said for major systems it would give a less-than-10-percent cancellation rate (versus today's 25 percent rate), a one- to two-year development cycle (versus today's three- to 10-year cycle), and million-dollar budgets (versus today's multi-million-dollar budgets). Systems produced from this system would be reliable, have high quality, and require little maintenance, he said.

---

**TOOLS FOR SOFTWARE QUALITY**

*From the leaders in modern software engineering technology*

**Eiffel**: The best object-oriented environment
- Multiple inheritance
- Generic classes, static typing, assertions, and much more
- Implementation through C for full portability
- An open environment, allowing reuse of existing software
- Support for the automatic generation of C packages
- Automatic incremental recompilation
- Tools for documentation, debugging
- The ideal system for developers of high-quality software
- Particularly appropriate for software tool developers

**Cépage**: The multi-language structured editor
- Support for any language you want to use
- A highly visual interactive interface
- Input: menu-driven or just by typing (on-the-fly parsing)
- Good for programs, specifications, technical documents...
- Automatic support for your style standards and conventions
- Adapt to any language in a matter of hours

**OUR SEMINAR: Object-oriented Design**

San Francisco (California) Nov. 19-20

Call us today at (805) 685-1006

270 Storke Road, Suite 7, Goleta, CA 93117 (USA)
In Europe, Cépage and Eiffel are trademarks of Interactive Software Engineering, Inc.

**NEWS BRIEFS**

Boston University boosts SE program. After acquiring the defunct Wang Institute earlier this year, Boston University has expanded its software-engineering masters degree program. The program is part of the university's masters of science program in systems engineering. The new program will begin in January, said John Brackett, a former Wang Institute professor now teaching software engineering at Boston University. The program will have two curricula: one for students with a hardware background and one for those with a software background. The two curricula are identical except that the hardware track requires a data-structures course and a operating systems course while the software track requires a switching-theory course and a computer architecture course. Both curricula will emphasize software engineering for embedded systems and networked systems, Brackett said.

Because of industry demand for such a program, Brackett has established a corporate associates program so local companies can help pay for the program they said they wanted to keep, he said. Associate companies are guaranteed annual admission slots, the number of which is based on their membership fees.

Programmer jobs continue to grow. The number of programmers and software developers in the US grew 8.6 percent to 238,500 from July 1986 to July 1987, according to US Labor Dept. statistics. That pace surpasses the 6.2-percent growth seen between November 1985 and November 1986, when the 1985 computer-industry slump apparently bottomed out. The overall computing and data-processing field saw a 7.5-percent rise in employment to 640,900 from July 1986 to July 1987.

In the same period (the latest available), wages increased by 2.8 percent to an average of $454.96 a week for the overall computing field. (The department does not track programmer wages.) The amount of hours worked each week declined 1.5 percent to 37.6 hours a week.