

TECHNOLOGY & MEDIA

NEW ECONOMY

'Software Gap' Solution: Tools and Technology

By STEVE LOHR

As Internet technology spreads across the economy, the pace may be uneven and the impact uncertain for companies large and small. Yet at least one thing is certain: there will be a huge demand for more software and more reliable software.

Given that writing software remains as much art as science — irritatingly immune to the automation that computing itself has brought to other endeavors — the looming software challenge is stirring concern.

In a report in 1999, an early warning of the coming "software gap" came from a presidential advisory group composed of leading computer scientists from companies and universities. "Software is the new physical infrastructure of the information age," they wrote in framing the issue. "It is fundamental to economic success, scientific and technical research, and national security."

So what is the answer to the software challenge?

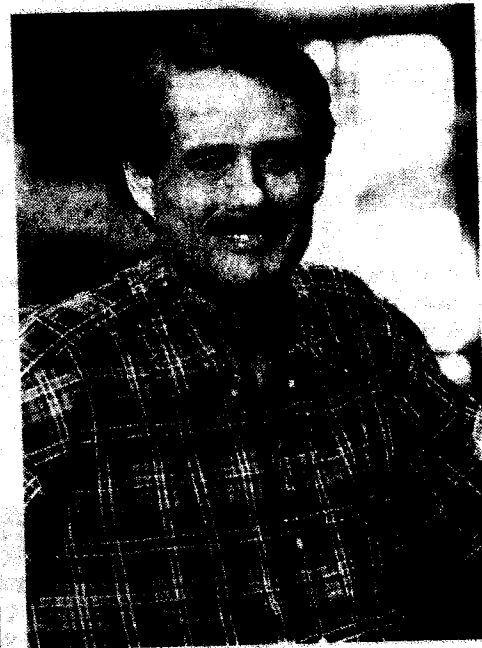
The two main answers, according to computer scientists, are new tools that will make programmers more productive and new technology that will enable more people who use software to program for themselves. And simple economics should speed the process along, they predict, because there could be a handsome payoff for any solutions.

In the long term, the computer scientists agree, the nation's software challenge is not a body-count issue. The continuing public debate surrounding the "software gap" has mainly been focused on the practice of recruiting foreign programmers, who are issued visas for skilled workers, instead of trying to retrain American workers for those jobs. But that is likely to be a short-lived phenomenon, according to the experts.

As the Internet makes it easier to share work around the world, software will be increasingly procured from a global market, just as sophisticated manufactured products are now. A personal computer is made of components from dozens of countries. And there is not a lot of silicon being baked in Silicon Valley anymore, though the Valley remains a hub of innovation for the chip industry. Software, experts say, will probably follow the same pattern with much research, design and early-stage development centered in America but production shifting overseas. The outsourcing of software today from places like India is just the beginning, they predict. The trend will be accelerated by the spread of high-speed Internet connections internationally and the growing share of the world's programmers trained abroad.

Irving Wladawsky-Berger, an I.B.M. software executive who is a co-chairman of the president's information technology advisory committee, estimates that by 2005 the software-dependent technology in use will be 100 times the amount today because the plummeting cost of computing devices, all connected to the Internet, will make them more widespread.

Yet Mr. Wladawsky-Berger also sees several forces that should work to close the gap. One is the shift from software's being a physical product that is shipped to software's becoming a continuous service, delivered over the Internet. The latter approach — combined with the trend toward making software programs from building



Ken Kennedy, a Rice University professor, researches application development.

blocks of code, or objects — will enable companies to order up software services tailored to their requirements, reducing their need to have armies of in-house programmers to make customized software.

Another tool that should reduce the strain on programmers, according to Mr. Wladawsky-Berger, is the rise of more common software standards, a trend led by the Internet and the Web, with its open communications protocols and software interfaces. The trend is being supplemented by the rise of open-source software, like the Linux operating system and the Apache Web server. The underlying code for open-source software is distributed freely, rather than being owned by a single company.



Susan B. Markisz for The New York Times

Irving Wladawsky-Berger is co-chairman of a technology advisory committee.

With traditional proprietary software, programmers spend much time writing software applications to run on a particular vendor's software platform, like Microsoft's Windows, Sun Microsystems' Solaris or I.B.M.'s AIX. "We have this horrible tower of babble in software, and the more we move to standards, the less we are fragmenting the labor of the programming community," Mr. Wladawsky-Berger said.

Internet computing has also cast old software issues in a new light. In the past, software quality was a constant problem, but in most cases flawed software was a nuisance rather than a catastrophe. Internet commerce, however, depends on reliable systems that are up and running around the clock. And while a flaw that may crash a desktop PC is an irritation, a flaw in a program on the Internet could have severe economic or security consequences.

So there is suddenly a surge of research interest in software quality. "Now it's become a very interesting problem," said Amitabh Srivastava, a senior researcher at Microsoft. Indeed, the National Science Foundation and the Defense Department earmarked grants last September totaling \$3.5 million for a three-year research program, the Open Source Quality Project, led by Alex Aiken of the University of California at Berkeley.

The software gap has also brought renewed interest in an old vision, that of making everyone a programmer. The idea is to simplify things enough so that nonprofessionals can program. A computer program, after all, is merely a rule-based set of instructions, not all that different, some say, from following a recipe to bake bread.

There have been partial steps toward the democratization of programming, from

Visicalc, the first electronic spreadsheet, to the hypertext markup language, or HTML, the simplified English-laden computer language used to make Web pages. The reason there are billions of Web pages is because they are easy to build.

"Bringing the computer science down to end users so they can do some simple application development themselves is an incredibly attractive proposition," said Ken Kennedy, a professor at Rice University, who is conducting research on the subject.

Yet Mr. Kennedy, who is also a member of the presidential advisory committee, said all the suggested answers to the "software gap" were likely to be merely positive steps rather than silver-bullet solutions.

"I venture to say that 20 years from now we'll be talking about the software gap," Mr. Kennedy said.