

Software Failures Serious Business

SAN JOSE, Calif., Oct. 5, 2004



(Photo: AP)

QUOTE

"The limit we're hitting is the human limit, not the limit of software. Technology has gotten ahead of our organizational and command capabilities in many cases."

Joshua Greenbaum
Enterprise Applications Consulting

(AP) New software at Hewlett-Packard Co. was supposed to get orders in and out the door faster at the computer giant. Instead, a botched deployment cut into earnings in a big way in August and executives got fired.

Last month, a system that controls communications between commercial jets and air traffic controllers in southern California shut off because some maintenance had not been performed. A backup also failed, triggering potential peril.

Computer code foul-ups also recently held Tacoma, Wash.'s budget hostage, delayed financial aid to university students in Indiana and caused retailer Ross Stores Inc.'s profits to plummet 40 percent after a merchandise-tracking system failed.

Such disasters are often blamed on bad software, but the cause is rarely bad programming. As systems grow more complicated, failures instead have far less technical explanations: bad management, communication or training.

"In 90 percent of the cases, it's because the implementer did a bad job, training was bad, the whole project was poorly done," said Joshua Greenbaum, principal analyst at Enterprise Applications Consulting in Berkeley. "At which point, you have a real garbage in, garbage out problem."

As governments, businesses and other organizations become more reliant on technology, the consequences of software failures are rarely trivial. Entire businesses -- and even lives -- are at stake.

Many experts believe the situation will only worsen as software automates new tasks and more systems interconnect with and rely on other computers. Technical challenges may be surmounted, but managing people never gets easier.

"The limit we're hitting is the human limit, not the limit of software," Greenbaum said. "Technology has gotten ahead of our organizational and command capabilities in many cases. It's amazing when you go into companies and see the kinds of battles that go on."

Big software projects -- whether to manage supply chains, handle payroll, track inventory, prepare finances -- tend to begin with high expectations and the best intentions. They're all about efficiency, reliability, cost-savings, competitiveness.

Companies might develop their own programs internally, outsource the job or buy from a company such as SAP AG, Oracle Corp. or PeopleSoft Inc. Regardless of the route, it's usually a major undertaking to get things right.

Often, however, the first step toward total disaster is taken before the first line of code is drawn up. Organizations must map out exactly how they do business, refining procedures along the way. All this must be clearly explained to a project's technical team.

"The risk associated with these projects is not around software but is around the actual business process redesign that takes place," said Bill Wohl, an SAP spokesman. "These projects require very strong executive leadership, very talented consulting resources and a very focused effort if the project is to be successful and not disruptive."

A 2002 study commissioned by the National Institute of Standards and Technology found software bugs cost the U.S. economy about \$59.5 billion annually. The same study found that more than a third of that cost -- about \$22.2 billion -- could be eliminated by improving testing.



A lack of strong leadership appears to have been a factor in HP's problem, which led to the dismissal of three top executives in its server and storage business hours after the company announced disappointing earnings on Aug. 12.

HP did not return a telephone call seeking comment but has said previously that its problems have been resolved. Wohl said the software, made by SAP, was not at fault.

Big projects also can sour during development, particularly when not enough resources are allocated, the people who will have a stake in the new system don't participate in planning and executives don't care. All can lead to miscommunication with the developers.

"Mistakes hurt, but misunderstandings kill," said John Michelsen, chief executive of iTKO Inc., which makes software that helps companies manage big software projects and test them automatically as they're being developed.

Too often, he said, programmers are handed a lengthy document explaining the business requirements for a software project and left to interpret it.

"Developers are least qualified to validate a business requirement. They're either nerds and don't get it, or they're people in another culture altogether," said Michelsen, referring to cases where development takes place offshore.

The Dallas-based company's LISA software attempts to reduce the complexity of testing, so non-technical executives in charge of major software projects can ensure the actual code adheres to their vision.

The lack of robust testing during and after such a project likely contributed to the Sept. 14 radio system outage over the skies of parts of California, Nevada and Arizona.

Though there were a handful of close calls, all 403 planes in the air during the incident managed to land safely, said FAA spokesman Donn Walker. A handful violated rules that dictate how close they are allowed to fly to each other - but the FAA maintains there were no "near misses."

The genesis of the problem was the transition in 2001 by Harris Corp. of the Federal Aviation Administration's Voice Switching Control System from Unix-based servers to Microsoft Corp.'s off-the-shelf Windows Advanced Server 2000.

By most accounts, the move went well except the new system required regular maintenance to prevent data overload. When that wasn't done, it turned itself off as it was designed to do. But the backup also failed. In all, the southern California system was down for three hours, though other FAA centers restored communications within seconds, Walker said.

The FAA's investigation is continuing, and Harris Corp. did not return a call seeking comment.

Michelson said the failure was in inadequate testing.

"On a regular basis, the FAA should have been downing that primary system and watching that backup system come up," he said. "If it doesn't go up and stay up, they would have known they had a problem to fix long before they needed to rely on it."

Another common theme in failures lies in the ranks of employees who actually must use the systems. Often they're not given proper training. There's also a chance that they don't want the project to succeed, especially if they see it as a threat to employment.

"It becomes a major role of (management) to kind of herd the cats in and make them all line up in a reasonable way," said Barry Wilderman, an analyst at the Meta Group. "That's why this stuff is so hard."

By Matthew Fordahl

©MMIV, The Associated Press. All Rights Reserved. This material may not be published, broadcast, rewritten, or redistributed.