MARCH 2019 VOLUME 33, NUMBER 3

PM Network®

The Power of the Right REFERENCES

PAGE 40

THE **TECH TALENT** GAP
PAGE 14

FAILING **UP**
PAGE 22

BUST-RESISTANT PORTFOLIOS
PAGE 34

MAKING PROJECT MANAGEMENT INDISPENSABLE FOR BUSINESS RESULTS.®

PMI® Project Management Institute

GETTY IMAGES

Artificial Exterminators

Billions of lines of code are written each year, with many applications built on open-source code shared widely among developers. Therein lies a problem: A single error can ripple into thousands of vulnerabilities, giving hackers a gateway to access secure information or disrupt major infrastructure systems. Even on a smaller scale, bugs can delay product releases, crash systems and hamper companies financially. The mistakes are costly: Software failures in 2017 resulted in US$1.7 trillion in lost revenue due to issues such as stock market price declines, lost revenue during system downtime and delayed future product releases because of talent being diverted toward remediation efforts, according to IT firm Tricentis.

Project teams are now testing whether artificial intelligence (AI) could be a solution. Whereas human detection of software bugs is time-consuming and imperfect, AI can identify common bugs quickly and efficiently. Government agencies in the U.S. and China have launched research projects to fuel the use of AI to spot errors in code. The private sector is also jumping in, with Facebook and French video game company Ubisoft launching projects last year to develop their own bug-spotting AI tools.

"The statistical nature of machine learning involves us changing the way we work."

—Yves Jacquier, Ubisoft Montreal, Montreal, Quebec, Canada

Game Changer

Ubisoft's project team fed its AI tool 10 years' worth of code from its software library to teach it what mistakes had previously been found and fixed. Rather than pointing out specific bugs, the tool tells programmers about the statistical likelihood of a bug appearing in a certain part of code.

One of the challenges Ubisoft had to address throughout the project was getting programmers on board, says Yves Jacquier, executive director, production studio services, Ubisoft Montreal, Montreal, Quebec, Canada. "The statistical nature of machine learning involves us changing the way we work," he says. Unlike traditional software, in which developers write out rules for the application to follow, machine-learning algorithms use data to guide how the software should act. "It requires a lot of change management to adapt the solution from a technical standpoint and determine the optimal threshold that maximizes the number of bugs caught while not having too many false positives."

To help ease the transition, the team is rolling out the tool iteratively, beginning with its Canadian video game production projects. The company is also training individual programming teams on how to use it. While labor-intensive, the benefits make it worthwhile: The company estimates that using such techniques can catch 70 percent of the bugs before reaching testing phases, freeing up teams to work on features that add more value.

Bug Spotter

Last year, the U.S. government completed a US$8 million project funded by the U.S. Defense Advanced Research Projects Agency and the U.S. Air Force Research Lab at the nonprofit research and development organization Draper. The development project sought to create a series of algorithms that enabled automated detection and repair of software flaws using its neural network-based machine-learning system, DeepCode. One of the team's challenges during the four-year project was finding the right data to train the tool.

"There weren't a lot of examples in the wild of code that were labeled as good and labeled as bad," says Jeffrey Opper, program manager, national security and space, Draper, Cambridge, Massachusetts, USA.

To address the issue, the team curated large sets of training data using specially developed test suites and open-source libraries. They flagged and labeled problems in the code as "bad," using static analyzers to teach DeepCode what errors look like. They also relied on an internal team of software experts at Draper to test DeepCode's accuracy and reduce the number of false alarms it raised. Training and refining DeepCode took 18 months, with the project wrapping in October. "The tool proved that DeepCode classifiers, with sufficiently robust data, can identify code flaws with significantly greater accuracy than open-source static analyzers," Mr. Opper says. *—Ambreen Ali*