Computers will soon be able to fix themselves – are IT departments for the chop?

October 13 2017, by Saemundur Haraldsson, Alexander Brownlee And John R. Woodward

They call me the digital lizard. Credit: Jeffrey B. Banke

Robots and AI are replacing workers at an alarming rate, from simple manual tasks to making complex legal decisions and medical diagnoses. But the AI itself, and indeed most software, is still largely programmed by humans.

Yet there are signs that this might be changing. Several programming tools are emerging which help to automate software testing, one of which we have been developing ourselves. The prospects look exciting; but it raises questions about how far this will encroach on the profession. Could we be looking at a world of Terminator-like software writers who consign their human counterparts to the dole queue?

We computer programmers devote an unholy amount of time to testing software and fixing bugs. It's costly, time consuming and fiddly – yet it's vital if you want to bring high quality software to market.

Testing, testing …

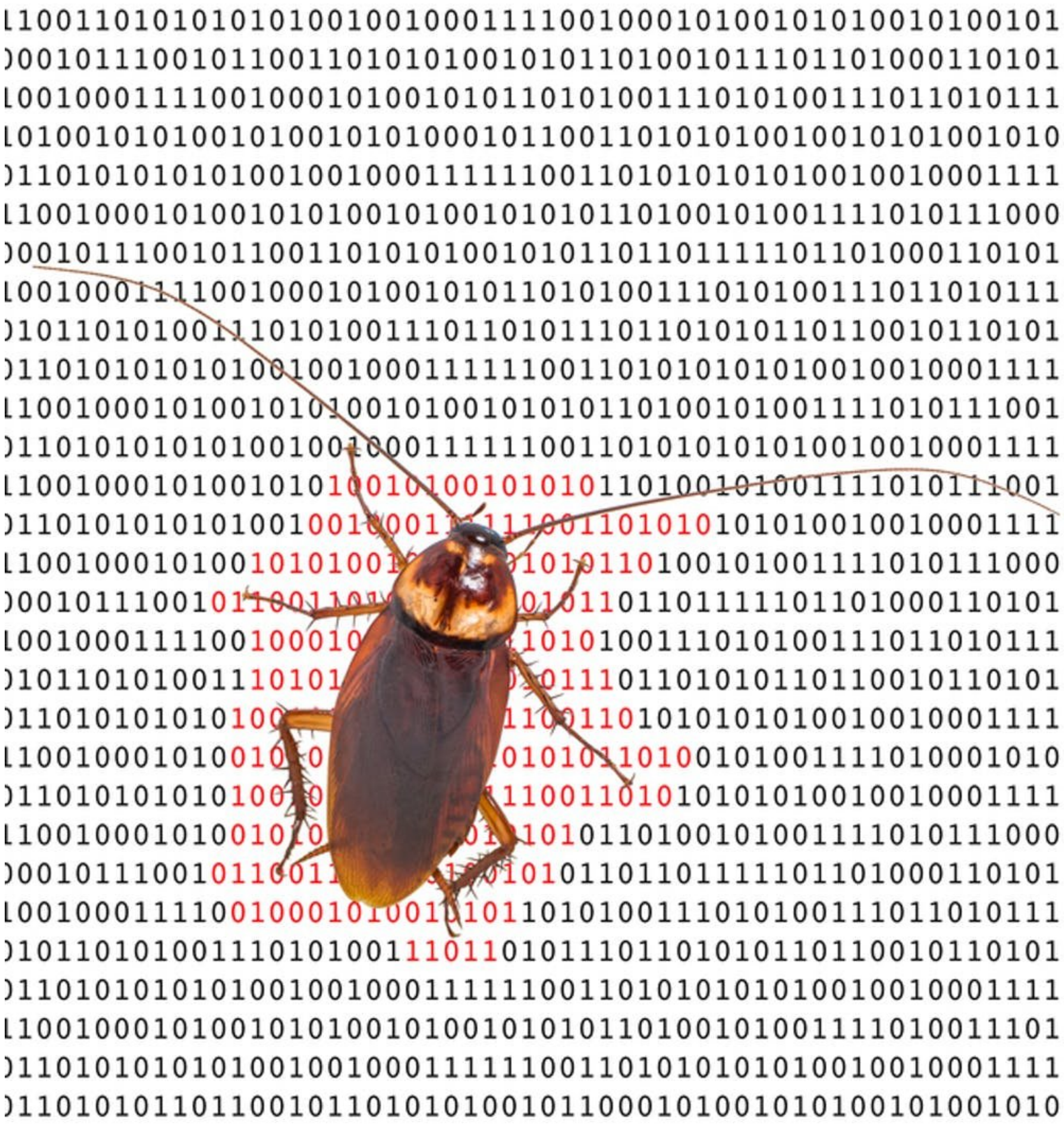
A common method of testing software involves running a program, asking it to do certain things and seeing how it copes. Known as dynamic analysis, many tools exist to help with this process, usually throwing thousands of random choices at a program and checking all the responses.

Facebook recently unveiled a tool called Sapienz that is a big leap forward in this area. Originally developed by University College London, Sapienz is able to identify bugs in Android software via automated tests that are far more efficient than the competition – requiring between 100 and 150 choices by the user compared to a norm of nearer 15,000.

The difference is that Sapienz contains an evolutionary algorithm that learns from the software's responses to previous choices. It then makes new choices that aim to find the maximum number of glitches and test the maximum number of kinds of choices, doing everything as efficiently as possible.

It may soon have competition from DiffBlue, a spin-out from the University of Oxford. Based on an AI engine designed to analyse and understand what a program is doing, the company is developing several automated tools to help programmers. One will find bugs and write software tests; another will find weaknesses that could be exploited by hackers; a third will make improvements to code that could be better expressed or is out of date. DiffBlue recently raised US$22m in investment funding, and claims to be delivering these tools to numerous blue chip companies.

The tool that we have developed is dedicated to bug hunting. Software bugs are often just an innocent slip of the finger, like writing a "+" instead of a "-"; not so different to typos in a Word document. Or they can be because computer scientists like to count differently, starting at zero instead of the number one. This can lead to so-called "off by one" errors.



Bug on out. Credit: Phichak

You find these annoying little glitches by making one small change after another – repeatedly testing and tweaking until you make the right one.

The answer is often staring you in the face – a bit like the game "Where's Wally?" (or Waldo if you're in North America). After hours of trying, you finally get that a-ha moment and wonder why you didn't spot it sooner.

Our tool works as follows: office workers go about their normal administrative duties in the daytime and report any bugs in software as they find them. Overnight, when everyone is logged off, the system enters a "dream-like" state. It makes small changes to the computer code, checking each time to see if the adjustment has fixed the reported problem. Feedback from each run of the code is used to inform which changes would be best to try next time.

We tested it for four months in a Reykjavik organisation with about 200 users. In that time, it reported 22 bugs and all were fixed automatically. Each solution was found on these "night shifts", meaning that when the programmer arrived at the office in the morning, a list of suggested bug fixes were waiting for them.

The idea is to put the programmer in control and change their job: less routine checking and more time for creativity. It's roughly comparable to how spell checkers have taken much of the plod out of proof-reading a document. Both tools support the writer, and reduce the amount of time you probably spend swearing at the screen.

We have been able to show that the same system can be applied to other tasks, including making programs run faster and improving the accuracy of software designed to predict things (full disclosure: Saemundur recently co-founded a company to exploit the IP in the system).

Future shock?

It is easy enough to see why programs like these might be useful to software developers, but what about the downside? Will companies be able to downsize their IT requirement? Should programmers start fearing that Theresa May moment, when the automators show up with their P45s?

We think not. While automations likes these raise the possibility of companies cutting back on certain junior programming roles, we believe that introducing automation into software development will allow programmers to become more innovative. They will be able to spend more time developing rather than maintaining, with the potential for endlessly exciting results.

Careers in computing will not vanish, but some boring tasks probably will. Programmers, software engineers and coders will have more automatic tools to make their job easier and more efficient. But probably jobs won't be lost so much as changed. We have little choice but to embrace technology as a society. If we don't, we'll simply be left behind by the countries that do.

This article was originally published on The Conversation. Read the original article.

Provided by The Conversation

Citation: Computers will soon be able to fix themselves – are IT departments for the chop? (2017, October 13) retrieved 2 May 2024 from https://techxplore.com/news/2017-10-departments.html

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.