Publication: Asian Scientist
Date: 16 September 2019
Headline: The (Software) Bug Doctor

The (Software) Bug Doctor

SMU Associate Professor David Lo has developed automated solutions to help software developers efficiently detect and repair defects in their code.

By Juliana Chan – The movies may portray software bugs as monumental, complex mathematical errors that only a savant can detect and resolve. In reality, some of the most serious cases of software failure were due to minor mistakes—easy errors, or simply a bad line of code—that the developers failed to spot in their code before product release.

While it is hard to ascribe a price to a software bug, software failures cost the global economy an estimated US$1.7 trillion in 2018, according to the Software Fail Watch, a report on software failures issued annually by Austrian software testing firm Tricentis. It identified 606 recorded software failures, impacting half of the world’s population and 314 companies.

Here’s where software analytics researchers like Associate Professor David Lo of the Singapore Management University (SMU) School of Information Systems can help. To improve software quality and reliability, Professor Lo analyses the large amount of data produced during the software lifecycle, including source code, bug reports and user feedback, and produces automated solutions that help to either prevent or reduce the impact of software failures.

“I work at the intersection of software engineering and data science. I analyze different kinds of software artifacts, such as code, execution traces, bug reports, Q&A posts and developer networks, and the interplay among them,” said Professor Lo, who revealed that he became fascinated by programming after a foray into QuickBASIC as a secondary school student.

Publication: Asian Scientist
Date: 16 September 2019
Headline: The (Software) Bug Doctor

Making debugging painless

Manual detection and repair software programs of their defects or bugs, known colloquially as "debugging", usually requires heavy investments in time and resources. Automated debugging solutions can dramatically improve developer productivity and system quality.

"I have designed an array of automated solutions to help developers manage a large number of bug reports. The solutions include techniques to identify duplicate bug reports, prioritise bug reports, assign bug reports to suitable developers, and locate buggy files given a bug report," Professor Lo said.

His solutions also include technologies that identify risky or buggy software changes (also known as commits), automatically repair programs, and uncover issues in deployed software through large-scale user feedback monitoring.

In a recent study, titled "Emerging app issue identification from user feedback: experience on WeChat" and published in the Proceedings of the 41st ACM/IEEE International Conference on Software Engineering, Professor Lo and colleagues at the Chinese University of Hong Kong and Tencent Inc. built a new tool to accurately identify emerging app issues early from large-scale user feedback. Called DIVER, short for iDentifying emerging app Issues Via usER feedback, the tool analyses real-time feedback from WeChat users. Once a problem is detected, the emerging issue is sent to developers immediately.

After the deployment of DIVER on WeChat, DIVER helped developers identify 18 emerging issues of its Android app and iOS app in January 2018. In evaluation experiments, DIVER significantly outperformed an existing user feedback analysis tool, IDEA, by 29.4% in precision and 32.5% in recall on average.

"Detecting emerging issues early is important to minimise the adverse impact of bugs in apps to their users. DIVER was able to identify many emerging issues of WeChat early, such as an emerging issue about failure to send 'red packets' in the WeChat app for iOS," he said.

Easy, breezy software development

Like finding a needle in a haystack, it can be very challenging to find specific lines of code from a large code base. Additionally, due to the fast pace of software development, documentation is often unavailable or outdated. All of these issues make automated tools for effective source code and documentation management crucial in software development.

In an ongoing project, titled "Automatic inference of software transformation rules for automatically back and forward porting legacy infrastructure software", or ITRANS, Professor Lo and colleagues aim to fully or partially automate the process of porting source code—such as bug fixes, security patches and drivers—from one version of an infrastructure software to another. The project is funded by the National Research Foundation Singapore and the Agence Nationale de la Recherche in France.

One solution the researchers developed, Coccinelle4J, was presented at the 33rd European Conference on Object-Oriented Programming (ECOOP 2019) in July 2019. Coccinelle4J allows developers who code in Java to specify patterns for program matching and source-to-source transformation.

“This process [of porting source code] is often done manually in industry, which costs time and resources. Automation can reduce IT costs and improve system reliability and security at the same time, as bugs and security loopholes in many systems can be addressed more quickly,” Professor Lo said.

Reliable AI system development

Code can be found aplenty in the burgeoning field of artificial intelligence (AI). Here, Professor Lo is keen to solve an emerging problem—how best to adapt software engineering processes and tools that are currently used to design conventional software for AI system development.

“I’m currently starting to work in the area of AI system engineering. AI is advancing rapidly and has been, or will be, incorporated into many systems that we interact daily, such as self-driving cars,” Professor Lo said. “My immediate future goals are to investigate and characterise the limits of the current best practices and tools in AI system development, and design novel solutions that address those limitations.”

In a project with colleagues from China, Australia and Canada, Professor Lo is characterising and quantifying the impact of technical debts and bugs in popular deep learning frameworks, such as TensorFlow and PyTorch.

“Deep learning systems today are often highly dependent on popular deep learning frameworks. Hence, quality issues on these frameworks have widespread ramifications and impact. A good understanding of such issues is necessary for us to subsequently develop testing, debugging, verification and even self-healing solutions,” Professor Lo said.

Asian Scientist Magazine is a media partner of the Singapore Management University Office of Research & Tech Transfer.