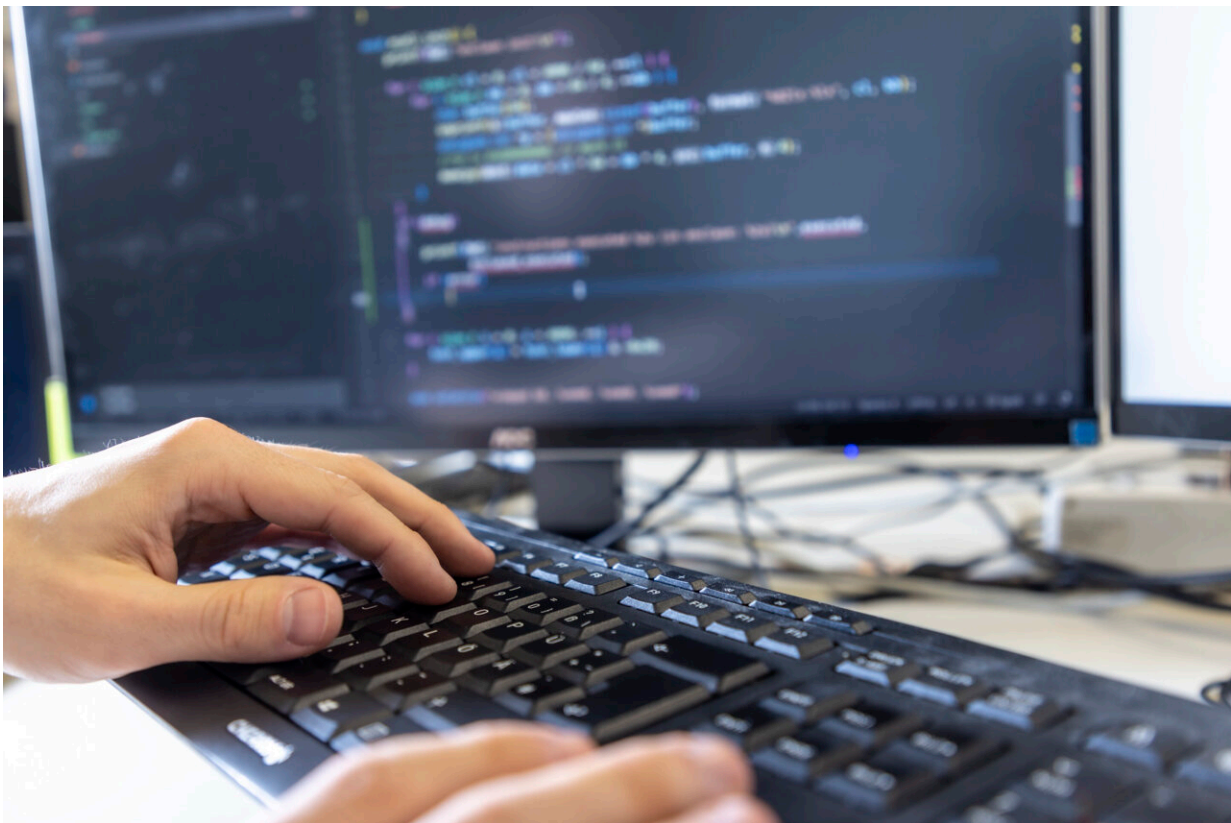


Researchers speed up fault localization during software development

July 31 2024



Debugging is currently still a big time waster. Credit: Lunghammer - TU Graz

Modern software applications usually consist of numerous files and several million lines of code. Due to the sheer quantity, finding and correcting faults, known as debugging, is difficult.

In many [software companies](#), developers still search for faults manually—something which takes up a large proportion of their working time. Studies indicate that this accounts for between 30 and 90% of the total development time.

Birgit Hofer and Thomas Hirsch from the Institute of Software Technology at Graz University of Technology (TU Graz) have developed a solution based on existing natural language processing methods and metrics that can greatly speed up the process of finding faulty code and thus debugging.

Fault localization uses up the most time

"As a first step, we conducted surveys among developers to find out what the biggest time wasters are when debugging. It turned out that the actual bug fixing is not the big problem at all, but that programmers mainly get bogged down with locating faults, i.e. narrowing down the search to the right area in the program code," explains Birgit Hofer.

Based on this realization, the researchers set about finding a solution to this problem which is also scalable to applications with a lot of code.

Although there are efficient model-based approaches in which a program is converted into a logical representation (referred to as a model), this only works for small programs. This is because the computing effort increases exponentially with the size of the code.

The approach taken up by Birgit Hofer and Thomas Hirsch represents certain software properties in numbers—for example, the readability or complexity of code—and can also be used for large amounts of code, as the computational effort only increases linearly.

Comparison of bug description and code

The starting point for fault localization is the bug report, for which testers or users fill out a form in which they describe the observed failure and enter information about the software version, their [operating system](#), the steps they took before the failure occurred and other relevant information.

Based on this bug report, the combination of natural language processing and metrics analyzes the entire code with regard to classes and the names of variables, files, methods or functions and the calls to methods and functions.

The application identifies [code](#) sections that best correspond to the bug report. As a result, the developers receive a list of five to 10 files ranked according to the probability of their being responsible for the observed failure.

The developers also receive information on the type of [fault](#) that is most likely to be involved. This data can be used to locate and fix the bug more quickly.

"The working time of software developers is expensive, yet they often spend more of this expensive time locating and fixing bugs than developing new features," says Birgit Hofer.

"As there are already a number of approaches to eradicating this problem, we have investigated how we can combine and improve them so that there is a basis for commercial application. We have now laid the foundations and the system works. However, in order to integrate it into a company, it would still have to be adapted to the company's respective needs."

The debugging system is available via the "[GitHub](#)" platform. On the [project website](#) the papers and repositories associated with this research can be found.

More information: Birgit Gertraud Hofer, Automated Debugging in Use. (2024) [DOI: 10.55776/P32653](#)

Provided by Graz University of Technology

Citation: Researchers speed up fault localization during software development (2024, July 31) retrieved 31 July 2024 from <https://techxplore.com/news/2024-07-fault-localization-software.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.