

APPLYING AI TO REDUCE SOFTWARE IMPROVEMENT COSTS

Ipek Ozkaya and James Ivers

September 2019

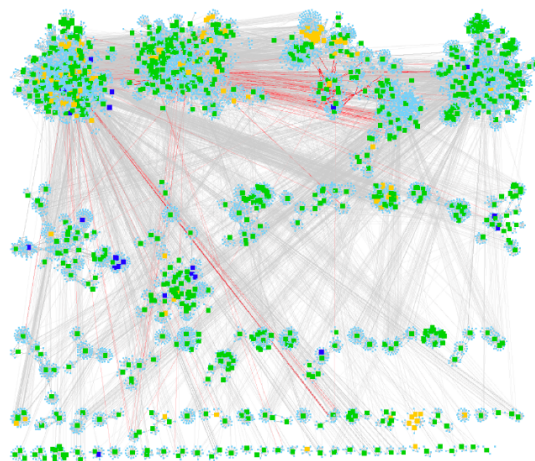
Software-reliant systems need to evolve over time to meet new requirements and take advantage of new technology. However, all too often the structure of legacy software becomes too complicated to allow such improvements to be made quickly and cost effectively. Programs often find that before they can undertake strategic modernization or migration activities like migrating to the cloud or rehosting software on new platforms, they first have to improve the structure of their software. This challenge is particularly common in long lived systems.

While software can be refactored to isolate a collection of functionality, which is an essential step in making modernization and migration activities practical, this is a labor intensive process whose cost can be difficult to justify. In one DoD example, a contractor estimated that 14,000 hours of software development work alone (excluding integration and testing) would be needed to isolate a mission capability from the underlying hardware platform.

The SEI has kicked off several projects that apply AI techniques to automate labor-intensive software engineering activities, starting with automation that recommends and implements refactorings that isolate functionality from its tangle of dependencies with the rest of the system. Our work combines advances in search-based software engineering, static code analysis, machine learning algorithms, and refactoring knowledge. With this combination, we aim to reduce the time required for this kind of architecture refactoring by two-thirds. If successful, our work will reduce the development time required for the above example to less than 5,000 hours.

With the work that we have already completed, we can help programs with C# software to analyze the implications of plans to break legacy applications into service or microservice architectures, migrate services to the cloud, rehost software to new platforms, or replace dated software components with newer options. Our initial work can help determine to the size of proposed changes, which is beneficial for portfolio analysis or increment planning within a program.

The SEI would like to collaborate with the right programs to apply this work to address today's important problems and gain feedback to improve our ongoing research.



Contact Us

Ipek Ozkaya
Technical Director
Engineering Intelligent Software Systems
ozkaya@sei.cmu.edu
412-268-3551

James Ivers
Initiative Lead
Architecture Analysis, Design, and Automation
jivers@sei.cmu.edu
412-268-7793

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612
Web: www.sei.cmu.edu