SHF: Medium: Collaborative Research: Finding and Fixing Architectural Hotspots: An Economics-Based Decision Support Approach
PIs: Yuanfang Cai (Drexel), Rick Kazman (U. Hawaii)

Recent research has revealed the strong correlation between error-proneness/change-proneness and software architecture: even though a software system may have hundreds of buggy files, these files always form just a few architecturally connected groups—the architecture hotspots. Hotspots exhibit architecture problems that propagate errors among large numbers of source files. This phenomenon has been observed over numerous projects, both open source and industrial, regardless of their domain, age, or programming language. The implication is that it is impossible to reduce error or change rates in complex software systems without fixing the architecture problems that cause errors to propagate.

The objective of the research proposed here is to guide the identification of high-maintenance architecture problems, quantitatively characterize their consequences in terms of software quality and productivity, create business cases to justify their refactoring, with the end goal of reducing maintenance costs through strategic architecture improvement.

The key to this research is to automatically extracting architecture hotspots, and quantifying their economic consequences in terms of increased bug fixing effort or reduced ability to deliver features. This quantification involves building models that allow an architect or project manager to reason about the economic implications of refactoring hotspots. Collecting information that is broadly available in software project—on bugs, changes, and commits—an architect can measure base rates of defects and changes in architectural hotspots and compare these to the rates of changes and defects in the project in general. Furthermore, by analyzing significant numbers of large-scale projects, it will be possible to categorize hotspots into recurring architecture anti-patterns. An architect, armed with a catalog of anti-patterns and an identification of hotspots in his/her own project can then plan refactorings to the hotspots and confidently estimate the costs of such refactoring. Finally, this approach will be validated via longitudinal studies with industrial partners where both refactoring recommendations and economic predictions can be made and tracked over time. A comparison of the before and after states will provide the necessary data to validate and tune the prediction models.

Intellectual merit. This is the first work to identify software errors by locating architecture problems that propagate errors, making it possible to significantly reduce error-proneness through targeted architectural refactoring. This work is also the first that enables quantitative analysis of the economic consequences of architecture hotspots, economic implications of refactoring, and the creation of refactoring business cases. Modeling software architecture as multiple, overlapping design spaces, formed by both structure and evolutionary relations, creates new possibilities for how software architecture should be viewed, analyzed, and quantified.

Broader impact. If successful, this research will fundamentally change how software defects are discovered, examined, and handled: instead of examining hundreds of individual defective files in a large system, an architect only needs to examine a few hotspots—architecturally connected groups of files. Thus many defects can be fixed simultaneously, by removing their architectural underpinnings, and this will provide substantial savings to large-scale industrial software projects. This research will produce significant and direct impact through extensive industrial collaboration. And it will also have significant educational impact by providing tool-support for the teaching of software architecture and design analysis.