### Goals & Objectives
- Enabling large-scale data analysis on large-scale software technical debt.
- Understanding how software and its technical debt evolves.
- Analyzing the impact of each developer and event on software quality.
- Reducing total ownership costs and schedules.

### Methodology
An approach to analyze software quality before and after each change.
An automated infrastructure to:
- Retrieve a subject system’s information from various sources (e.g., commit history and issue repository).
- Distribute hundreds of relevant revisions on multiple cloud instances, efficiently compile each revision, and run static/dynamic programming analysis techniques on it.
- Collect and interpret the artifacts generated by programming analysis techniques to extract quality attributes or calculate change.

A set of statistical analysis techniques tailored for understanding software quality evolution.
- Simple statistics, such as frequency of code smell introduction or correlation between two quality attributes.
- Machine learning techniques, such as clustering developers based on their impact.

### Future Research
- Implementing a private cloud solution for close-source applications.
- Extending SQUAAD to include more Static Application Security Testing (SAST) COTS.

### Contacts/References
Pooyan Behnamghader, pbehnamg@usc.edu  
Barry Boehm, boehm@usc.edu  
USC Center for Systems and Software Engineer