To **overcome cost** and **data barriers**, we prototyped a modular architecture that enables the rapid adoption of automated classifiers for static analysis alerts.

**Problem**
Static analysis alerts for security-related code flaws require too much manual effort to triage efficiently. **Organizations** are reluctant to fully adopt automated alert classifier technology because of barriers, including high cost, lack of expertise, and shortage of labeled data.

**Solution**
Develop an extensible architecture that supports classification and advanced prioritization, and builds on a novel test-suite-data method we developed.

- We developed a model and code intended to enable organizations to quickly start using classifiers and advanced prioritization by making API calls from their alert auditing tools.
- We implemented a prototype of the model.
- We developed adaptive heuristics for classifiers to adapt as they learn from test suite and natural program data.

**Approach**
- Design the architecture.
- Develop an API definition.
- Implement a prototype system.
- Develop adaptive heuristics and test them with datasets that combine test-suite and real-world (e.g., DoD) data.
- Test the architecture and prototype with collaborators.

**Source Code Analysis Integrated Framework Environment (SCAIFE)**

- **Registration Module**
  - Stores local projects
  - Displays project and alert data
- **Registration Module**
  - Generates registration tokens
  - Provides authentication
- **Prioritization Module**
  - Stores and evaluates prioritization formulas
- **Statistics Module**
  - Creates, runs, and stores classifiers
  - Stores adaptive heuristic algorithms
  - Stores automated hyperparameter optimization algorithms
- **DataHub Module**
  - Stores tool and alert information
  - Stores test suite metadata and alert determinations
  - Generates speculative mappings
- **Any static analysis tool can instantiate APIs to become a UI Module. For example**
  - SCAFE
  - DHS SWAMP
  - CCDC CSISR SWAT
- **Other aggregator tools**
  - Single static analysis tools

FY19 Artifacts

**Code and Test Results**
- Beta SCAIFE prototype VM (v1, v2) released to collaborators (August & September 2019)
- API definitions (0.0.2-0.0.5) YAML publication
- SCAFe v3 and v4: tool released with new features for collaborators to generate data
- SCAFe DevOps improvements for research transitionability
- SCAFe v.4.4: released to collaborators with features for SCAIFE integration (August & September 2019)
- Code developed for prototype
- Adaptive heuristics

**Publications**
- SCAFe API Definition and Prototype
- Manual: How to Review & Test the Beta SCAIFE (v1, v2) VM (August & September 2019)
- SEI blog post: An Application Programming Interface for Classifying and Prioritizing Static Analysis Alerts (July 2019)
- SEI whitepaper: SCAFe API Definition Beta Version 0.0.2 for Developers (June 2019)
- SEI technical report: Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools (May 2019)
- SEI blog post: SCAFe v3: Automated Classification and Advanced Prioritization of Static Analysis Alerts (December 2018)
- SWACon paper: Introduction to Source Code Analysis Laboratory (SCAFe) (November 2018)
- SEI webinar: How can I use new features in the CERT SCAFe tool to improve how my team audits static analysis alerts? (November 2018)
- Classifier Development Research
- Four in-progress papers addressing precise mapping, architecture for rapid alert classification, test suites for classifier training data, and API development

Project members developed (1) an architecture, (2) an API definition, and (3) a prototype system for static analysis alert classification and advanced alert prioritization.