



GSAW 2018

## CAE-SCRUB for Incorporating Static Analysis into Peer Reviews

Lyle Barner  
Jet Propulsion Laboratory,  
California Institute of Technology

© California Institute of Technology  
Government sponsorship acknowledged



**Jet Propulsion Laboratory**  
California Institute of Technology

# What is CAE-SCRUB?

- **Computer Aided Engineering-Source Code Review User Browser**
- Peer review tool for static code analysis
  - Originally developed by Gerard Holzmann of JPL's Laboratory for Reliable Software
  - Currently maintained by JPL's CAE group and Software Quality Assurance (SQA) group
- Used by many past and current JPL projects
  - Baseline version available to JPL projects that can be configured to meet project needs

# The Value of CAE-SCRUB

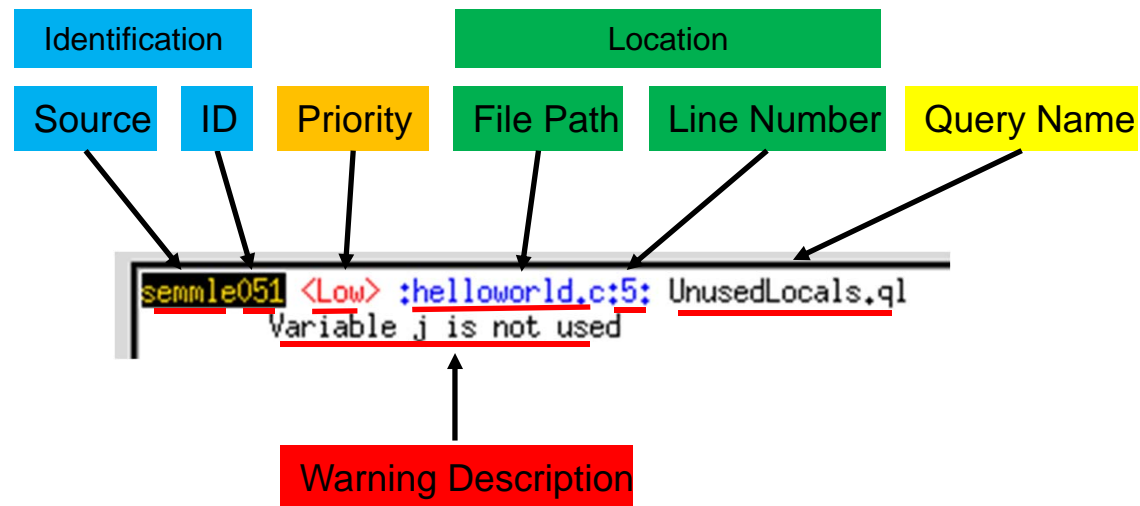
- Helps organize and guide the code review process
  - Aggregates and facilitates review of static code warnings
  - Captures and manages review comments
  - Allows developers and reviewers to concentrate on more contentious issues without neglecting code reviews
    - Combines effectiveness of peer reviews and total coverage of static analysis
- Integrates static code analysis reviews into the software development lifecycle by treating each static analyzer as a “peer” in code reviews

# How it Works

- Use configuration information to invoke different static analyzers to examine source code
- Filter warnings based on the scope of the peer review
- Provide standardized results that can be reviewed using the GUI as part of a regular peer review
- Use GUI's review process to agree with, disagree with, and discuss all issues found by the analyzers and add generic peer review comments
- Review results trigger code changes to resolve issues

# Standardization of Warnings

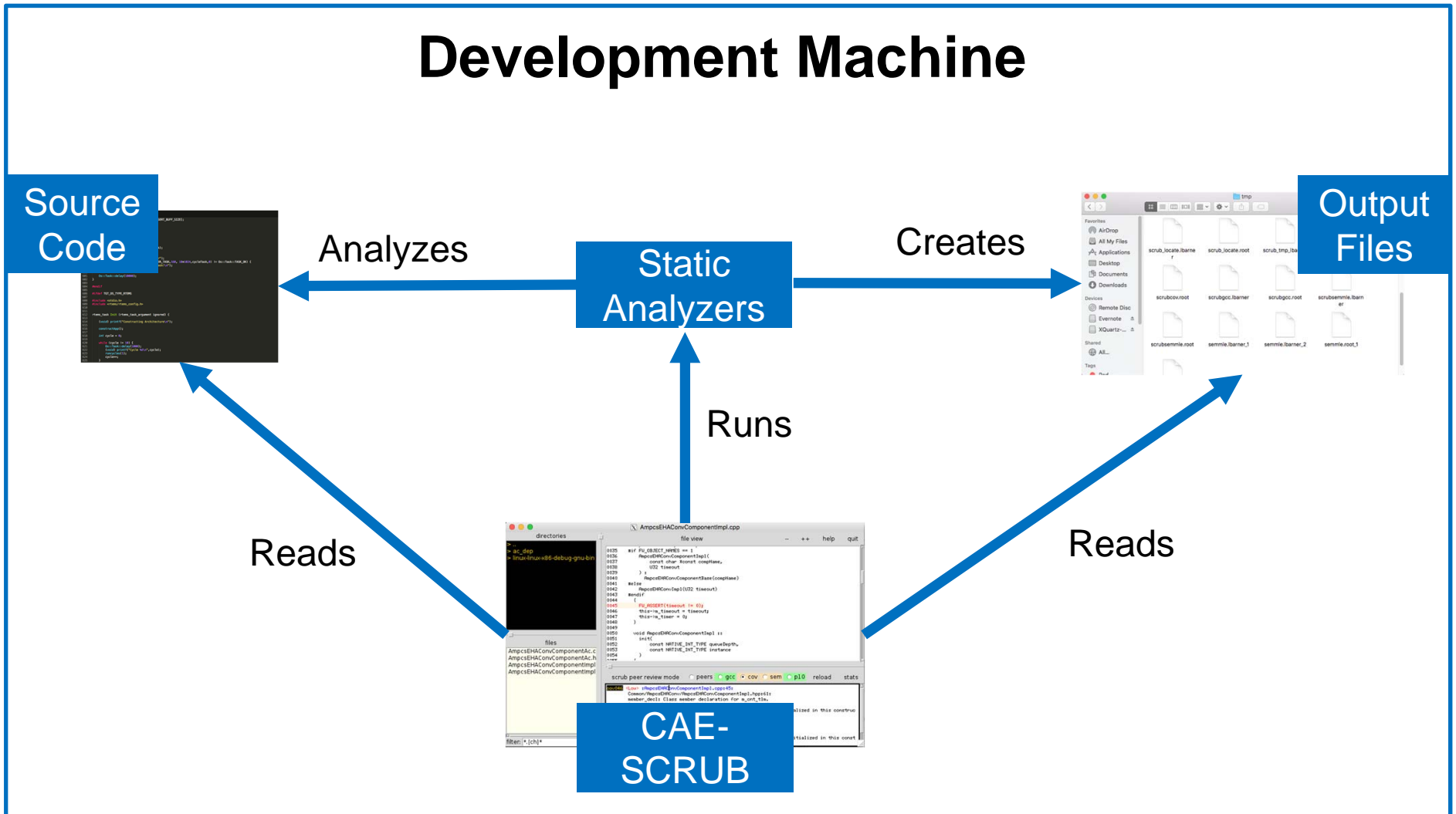
- A common format for displaying warnings
- Post-processing performs mapping from static analyzer format to CAE-SCRUB format



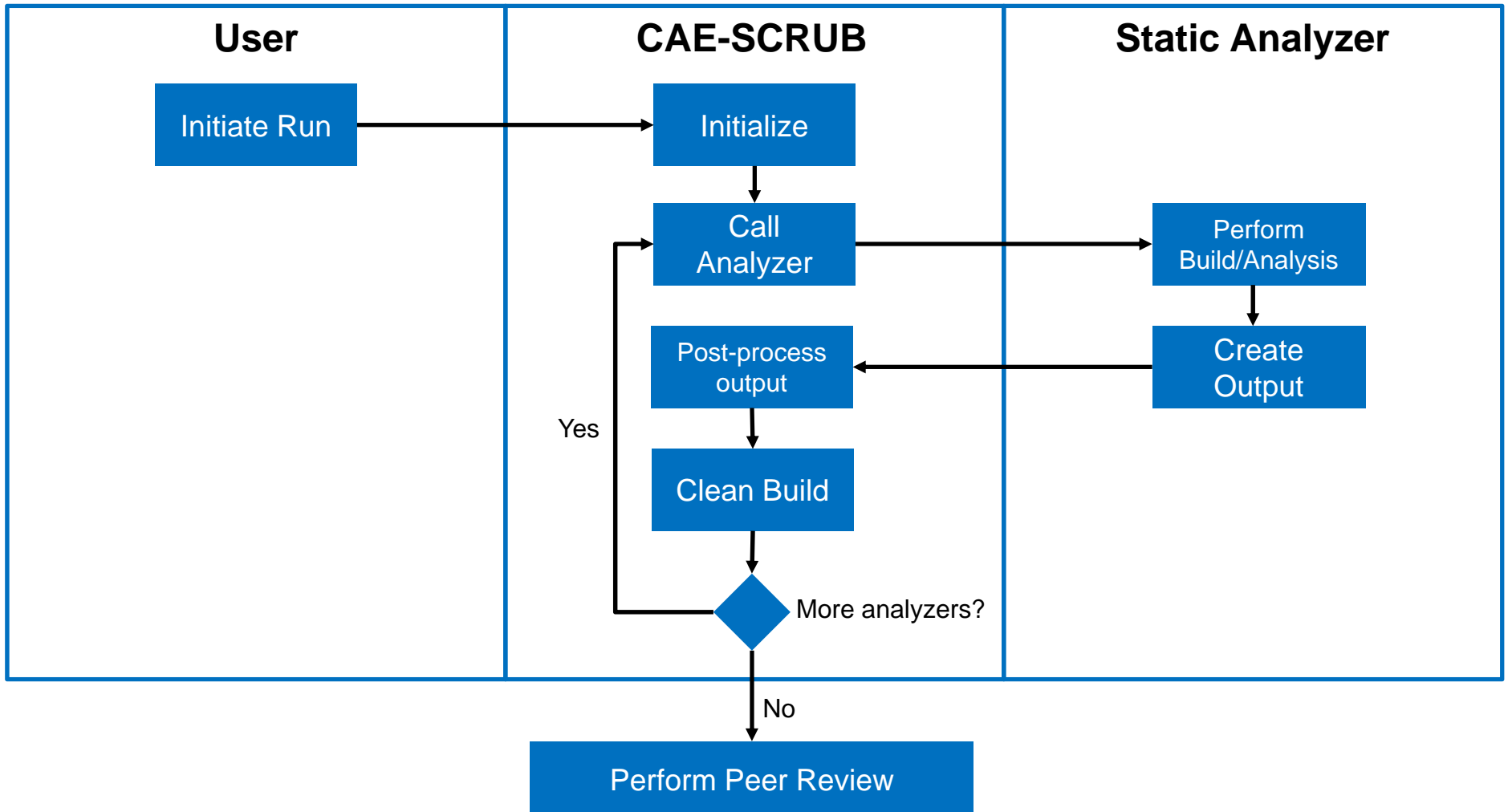
# Evolution of CAE-SCRUB

- Inherited a very well established version of SCRUB, but it was not suitable for large-scale deployment
- Refactored backend code
  - Improved architecture and stability
  - Simplified setup process via configuration file based setup
  - Improved error handling capabilities
- Improved installation guide and user guide
- Transitioned to git for version control

# Architecture



# Program Flow

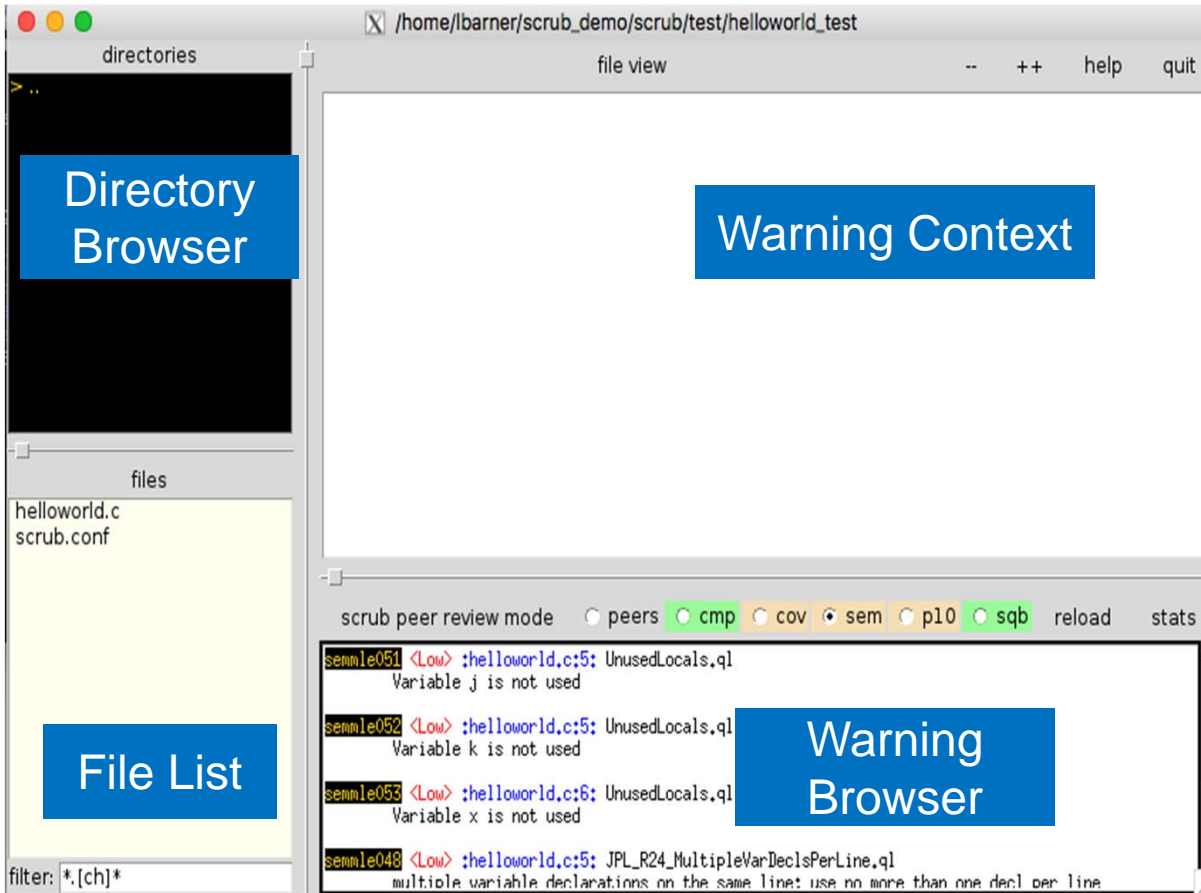




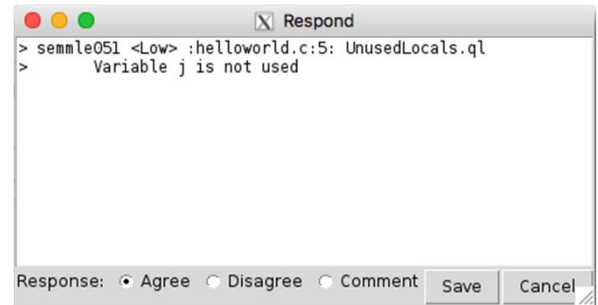
# Typical Usage Example

1. CAE-SCRUB is run on desired revision/branch of source code
  - Either manually or via system automation
2. Peer reviewers are notified of new results
3. Reviewers Agree/Disagree/Discuss results asynchronously
4. Lead developer analyzes peer review results and organizes peer review if necessary
  - Items where peer reviewers concur are not discussed
  - Solutions are proposed where applicable
  - False positives are noted and filtered out
5. Synchronous peer review is held to disposition remaining warnings

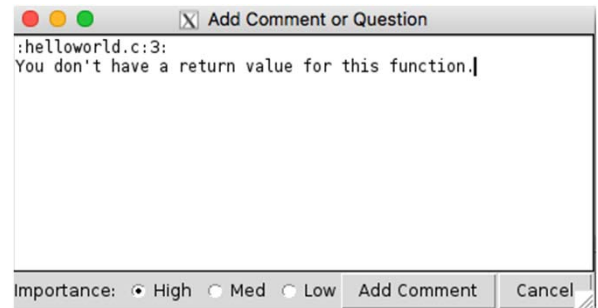
# GUI Overview



Reviewer can see each warning in context to help with discussion and disposition



Reviewers “vote” and add/edit comments



Reviewers disposition each warning

## Things CAE-SCRUB Does Well

- Provides a framework for static code analysis aggregation
- Provides a standardization of error types
- Streamlines the static analysis review process
- Implements a repeatable review process that can be integrated into development lifecycle

# Areas for Improvement

- Difficult to deploy
  - Requires detailed knowledge of how to configure multiple static analyzers
- Currently no integration with CM tools
- Number of warnings can be overwhelming
- Quality of results is highly dependent upon configuration
- No severity ranking information

# The Path Forward

- Investigate integration with other code review tools
  - Integration with COTS peer review tools can mitigate the need to maintain local deployments of CAE-SCRUB
- Create baseline set of queries to be run for each static analyzer
- Create ranking system for types of warnings
- General stability improvements for backend
- Customizable query lists for static analysis tools

# Summary

- CAE-SCRUB is a tool for integrating static analysis results into the peer review process
- It creates an extensible framework for connecting with static analysis tools
- Extensive work has been done to make large-scale deployment a possibility
- Integration with other software engineering tools is a top priority going forward



**Jet Propulsion Laboratory**  
California Institute of Technology

---

[jpl.nasa.gov](http://jpl.nasa.gov)

# Current Areas of Investigation

- Integration with CM tools such as git
- Integration with continuous integration tools such as Jenkins
- Integration with code review tools such as Collaborator



# Implementation

- Backend realization
  - Collection of bash scripts handle running the static analyzers
  - Collection of Python scripts handle post-processing of data from static analyzers
- Frontend GUI written in Tcl/Tk
  - Frontend handles viewing and commenting on the results from the static analyzers

# What is Static Analysis

- Identifies patterns in code that indicate refactoring opportunities to make code more maintainable
- Code reviews are not a feasible way to review millions of lines of code
- Provides automated checks against JPL coding standards and best practices
- Static analysis can perform verification, but not validation