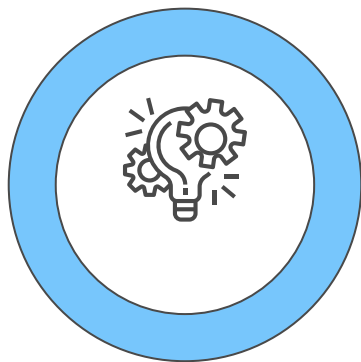


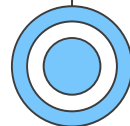
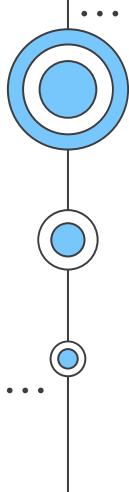
Eclipse Plugin for Potential Failure Analysis

by Josefin Wetterstrand
and Blenda Öhman



Introduction

The main goal was to create an Eclipse plug-in to collect, save and present data metrics from JUnit tests.



Background



PoFA

Analyses dynamic data and employs machine learning techniques to train prediction models



AspectJ

AspectJ was used to “listen” to the JUnit test and gather relevant data



Relevant work

Oskar Andersson did a similar project using the terminal. We used one file from his project.

Collected data metrics



Pass or fail

If the test passes or fails



Execution time

Execution time for test and called methods



Required memory

The required memory for both test and specific methods



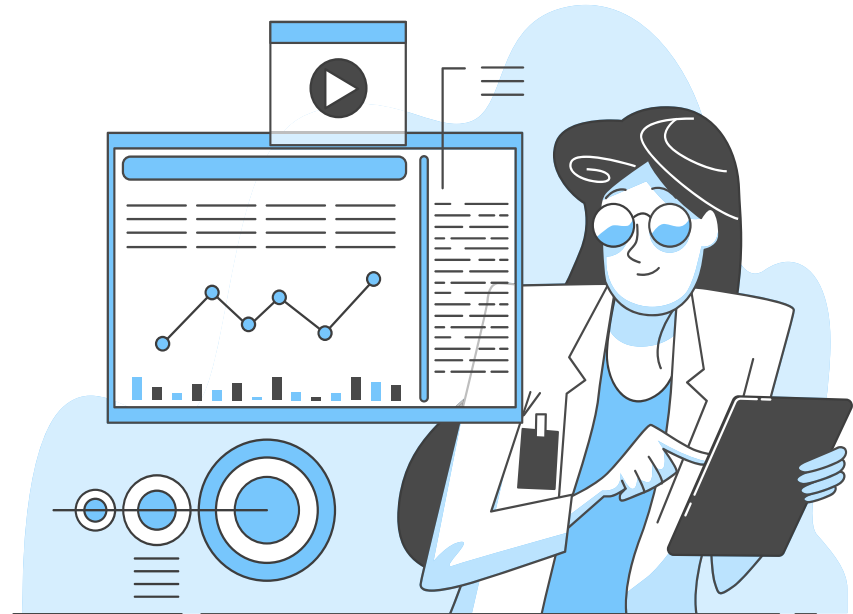
Read/Write to hard drive

Number of read/writes to hard drive



Classes called

Number of classes called outside the package





How to use the plug-in

01

Run the Plug-in

Creates a new window where you can import your project.

02

Run your JUnit Test

Our plug-in will collect relevant data about the tests.

03

Update the View

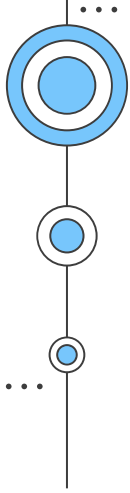
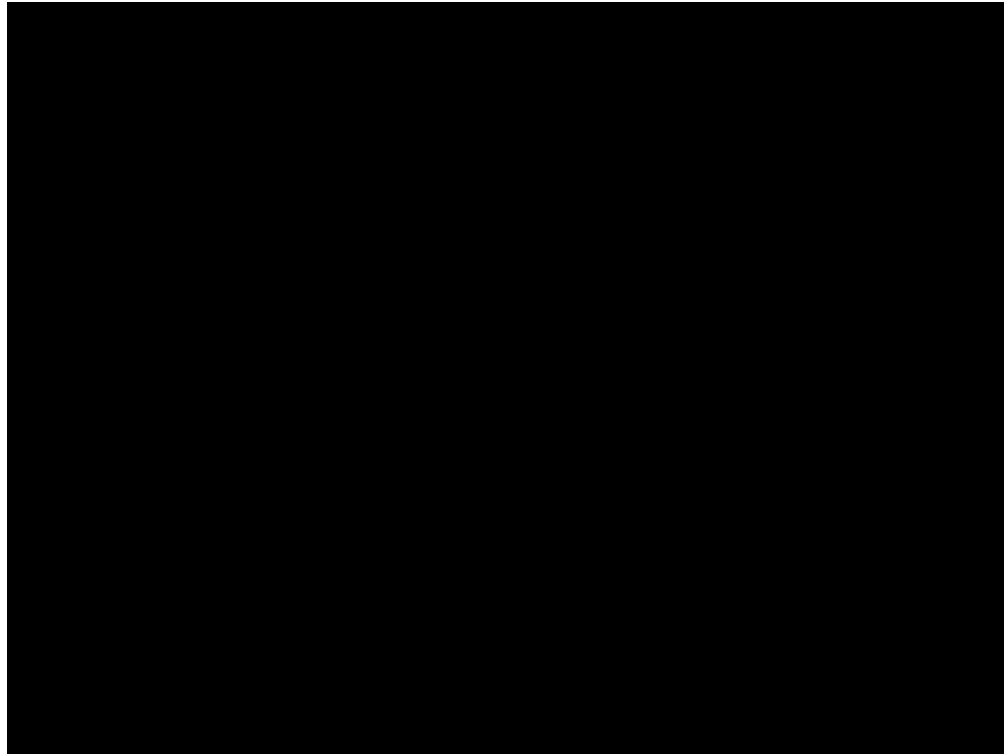
The newest data is presented in a table in the plug-in view.

04

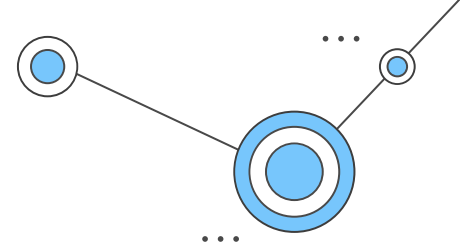
Use Collected Data

The test results are saved in a file named with the timestamp, that can be used for PoFA

Demo



Future Work



Portability

A .jar file with installation would be optimal for better portability

Usability

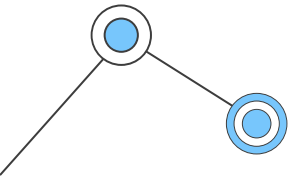
Cuts were made to only implement the most important features in the view

File Directory

File path is hard coded

Metrics

Saving number of execution times





Takeaway

We have created a working plug-in and our hope is that it can be used in the future for collecting data and using it for PoFA-analyzes.

**Thank you
for listening!**

