

# EclipsePro

Salahuddin Choudhary, Yanka Li,  
Colin Rothwell, Key Shin



## Features

- EclipsePro Audit
  - Code Audit
  - Code Metrics
- EclipsePro Test
  - JUnit Test Generation
  - Code Coverage Analysis





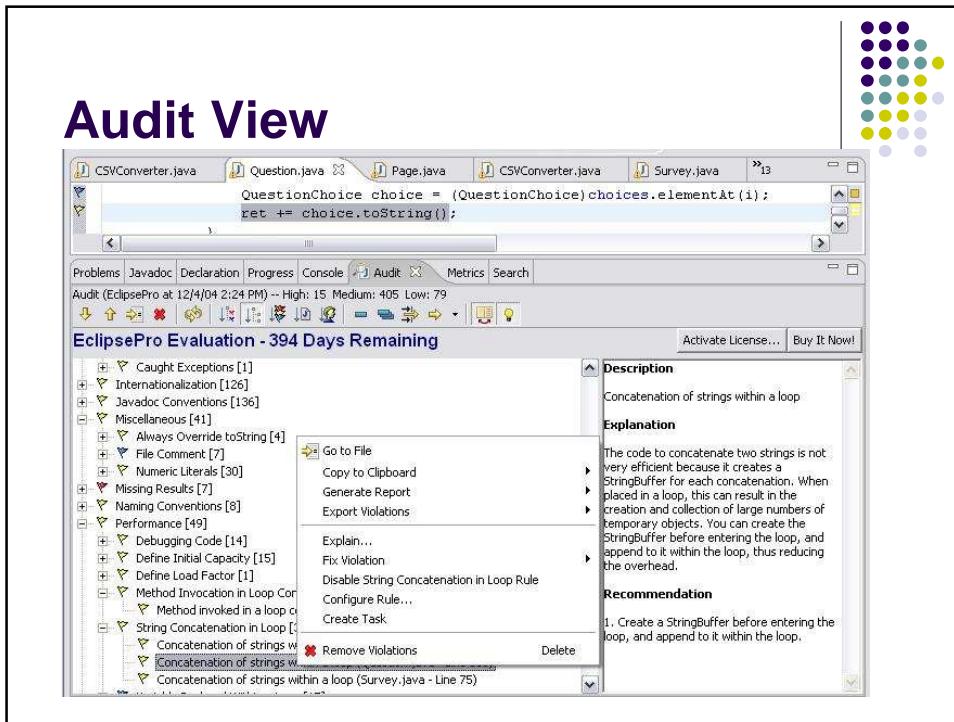
## Code Audit

- Directly plugs into eclipse
  - Also usable by IBM WebSphere
- Runnable on file/package/project
- Detects style/convention violations
  - Specified by a set of rules
  - Completely configurable
- Generates a list of issues (Audit View)
  - HTML/XML Report is also an option



## Violations

- Violations are specified by an audit rule
- Each rule is defined within a category
- Classified by severity (high, med, low)
- QuickFix is usable to correct some errors
- Violations are displayed in the Audit View
  - File/Line location of violation
  - Explanation
  - Recommendation for correction



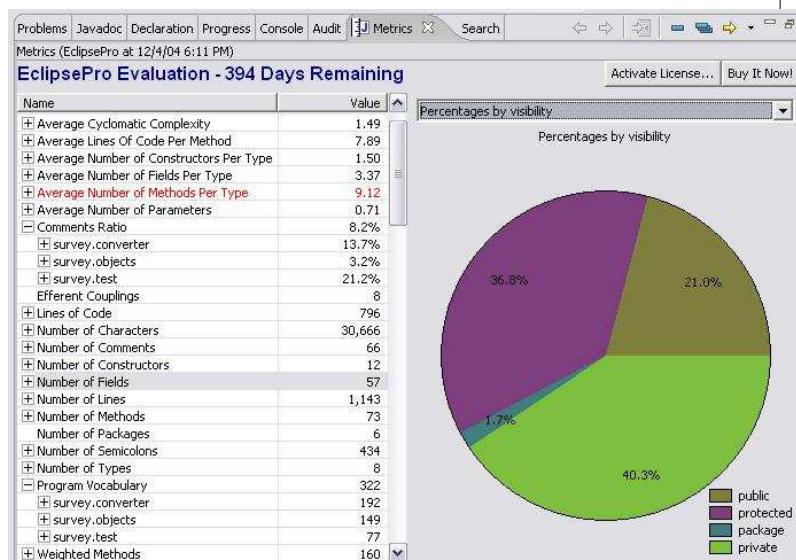
## Rules

- close() inside finally
- Appending strings
- Catching Exception
- Loop method calls
- Debugging code (XP)
- Internationalization
- Complexity
- FP comparison
- Efficient code
- Package names
- Import order
  - Conflicts with Eclipse!
- Null comparisons
- equals(), toString()
- Style issues
  - Whitespace
  - Brace use
  - Fields
  - For instead of While

## Code Metrics

- Measures key indicators of source code
- Some seem pretty useful
  - Average Cyclomatic Complexity
  - Comment Ratio
  - Program Vocabulary
- Others not so useful
  - Number of characters
  - Average number of methods per type

## Metric View





## Testing

- We focused mainly on Code Audit
- Some of the metric data are interesting
  - Not entirely clear what to do with it
- Audited our entire project
  - Actually corrected the more useful errors
  - Ran it on our technical review code
    - Audit found mostly style issues
    - Peer review found more execution errors
    - Both found package naming issue



## Benefits

- Easy to install and setup
  - Doesn't rely on dependencies
- Customize which rules to use
  - Can even create your own
- Can be set to run statically or dynamically
- Clearly explains violations
- Organizes by severity/category/author
- Doesn't need to execute the code

## Drawbacks

- Buggy
  - Multiple runs sometimes return different errors
- The default configuration is poor
  - Tedious to create your own
- Issues with Eclipse
  - Javadoc redundancy
  - Style conflicts (import order)
- Conflicting rules from different styles
- Difficult to write your own audit rules
- Expensive (\$400)

## Applicability

- Production code
  - Debugging code
  - Internationalization
  - Style
- Large scale
  - Performance
  - Benefits outweigh initial configuration cost
- Long term
  - Portability (interface use)