Improving Code Quality with Sonarqube!

Sonarqube

What is Sonarqube?

- Server based, static code analyzer
- Work with maven, gradle
- Provides useful measures on Code Quality
- Act as a "code reviewer", including advice on how to fix problem

What about other static analyzer tools?

- Some overlap with other tools (Checkstyle, Findbugs, Fortify)
- Best practice is to use more than one tool
- Highly likely that independent reviewer will run sonarqube on your code

Sonarqube Metrics

- Reliabillity (bugs)
- Security (vulnerabilities)
- Maintainability (code smells, debt)
- Coverage (test)
- Duplications (copy and paste)
- Size (code)
- Complexity ()
- issues (bugs + vulnerabilities + code smells)

Nice features

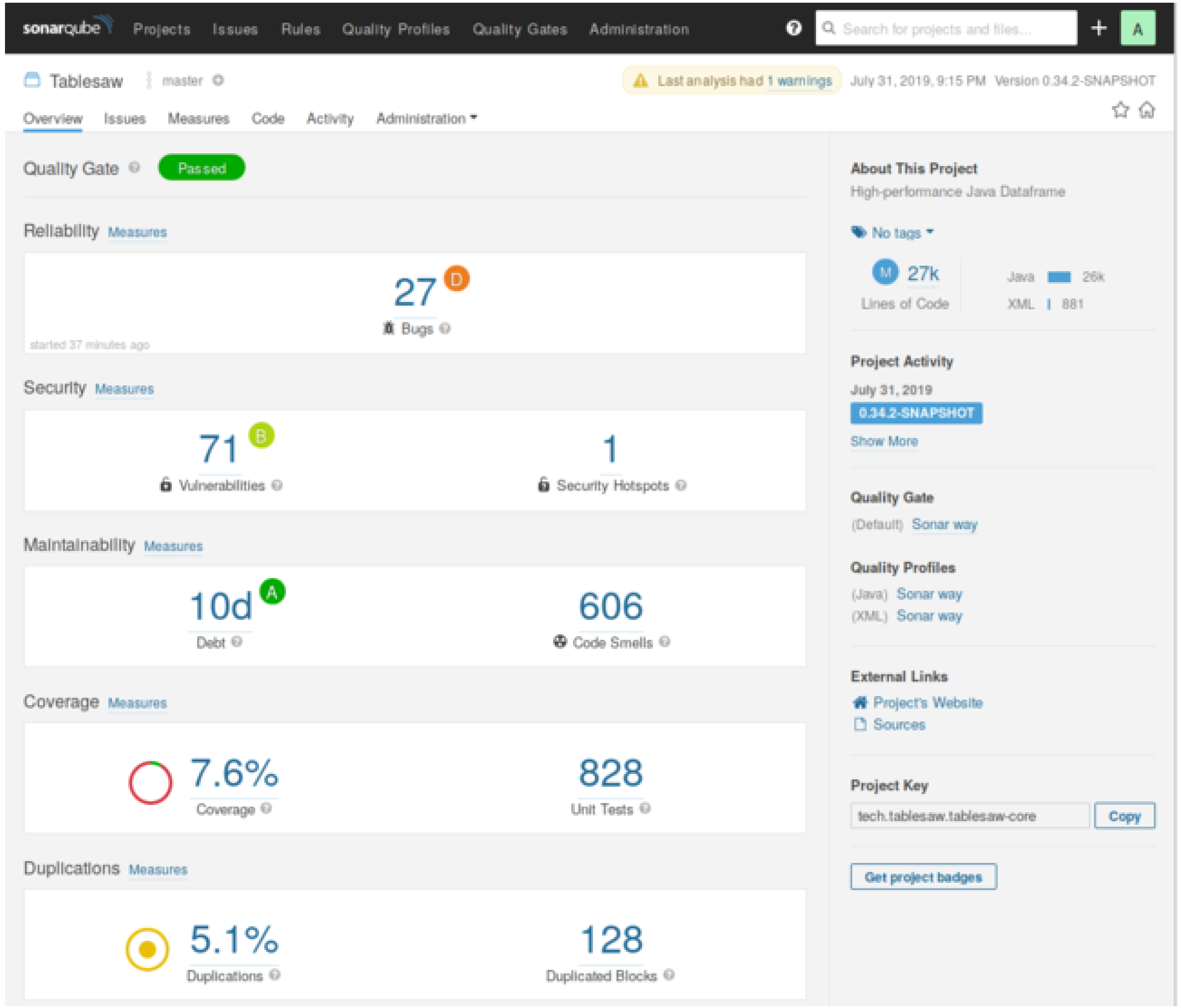
- Interactive graphic representation
- Each metrics shows total vs. new
    - Did you introduce problems recently?
- Each problem is associated with a suggested fix

Advices when using Sonarqube

- Think before making a change
  - false positive?
  - will I break some logic?
- Learning tool
  - patterns for most common issues
  - learn to recognize issue and how to fix it
- The boy scout rule
  - "Always leave the code you're editing a little better than you found it" (Robert C. Martin - Uncle Bob)
  - Clean up technical debt as you go
- As a project, agree on a Quality Profile
  - Quality Profile = set of rules run against your project

Project Dashboard

- Shows main metrics
- Grade each metrics
  - A-F
- Grade project
  - Passed / Failed



Tablesaw | master

Last analysis had 1 warnings July 31, 2019, 9:15 PM Version 0.34.2-SNAPSHOT

Quality Gate Passed

Reliability Measures

27 D Bugs

started 37 minutes ago

Security Measures

71 B Vulnerabilities

1 Security Hotspots

Maintainability Measures

10d A Debt

606 Code Smells

Coverage Measures

7.6% Coverage

828 Unit Tests

Duplications Measures

5.1% Duplications

128 Duplicated Blocks

About This Project

High-performance Java Dataframe

No tags

M 27k Lines of Code

Java 26k

XML 881

Project Activity

July 31, 2019

0.34.2-SNAPSHOT

Show More

Quality Gate

(Default) Sonar way

Quality Profiles

(Java) Sonar way

(XML) Sonar way

External Links

Project's Website

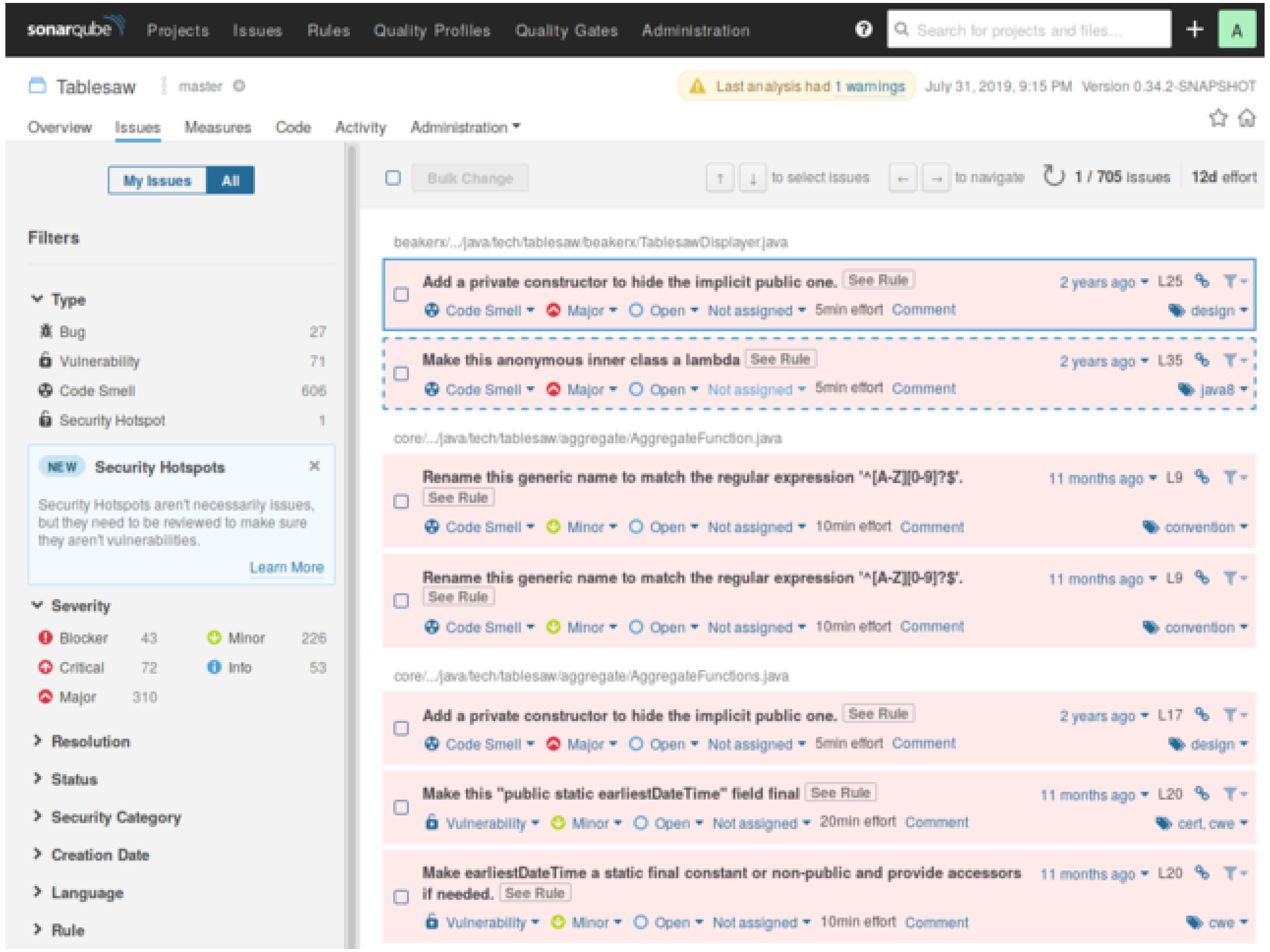
Sources

Project Key

tech.tablesaw.tablesaw-core

Issues

- All in one:
  - Bugs
  - Vulnerabilities
  - Code smells
- Severity:
  - Blocker
  - Critical
  - Major
  - Minor
  - Info
- Additive Filters
- Remember some may be False Positives



Security Hotspots aren't necessarily issues, but they need to be reviewed to make sure they aren't vulnerabilities.

beakerx/.../java/tech/tablesaw/beakerx/TablesawDisplayer.java

Issue	Type	Severity	Status	Assignee	Effort	Age	Line	Tags
Add a private constructor to hide the implicit public one.	Code Smell	Major	Open	Not assigned	5min effort	2 years ago	L25	design
Make this anonymous inner class a lambda	Code Smell	Major	Open	Not assigned	5min effort	2 years ago	L35	java8

core/.../java/tech/tablesaw/aggregate/AggregateFunction.java

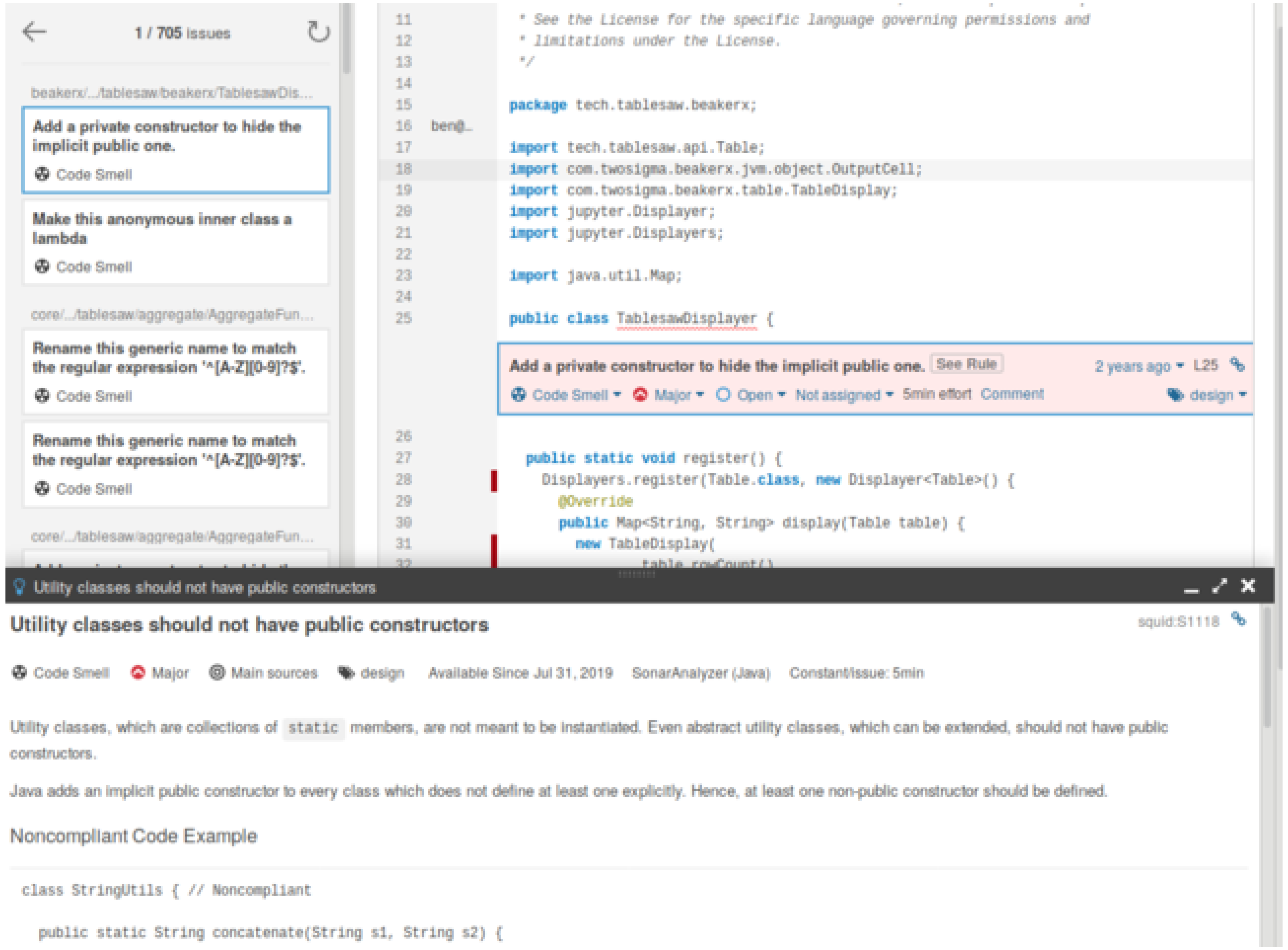
Issue	Type	Severity	Status	Assignee	Effort	Age	Line	Tags
Rename this generic name to match the regular expression '^[A-Z][0-9]?$'.	Code Smell	Minor	Open	Not assigned	10min effort	11 months ago	L9	convention
Rename this generic name to match the regular expression '^[A-Z][0-9]?$'.	Code Smell	Minor	Open	Not assigned	10min effort	11 months ago	L9	convention

core/.../java/tech/tablesaw/aggregate/AggregateFunctions.java

Issue	Type	Severity	Status	Assignee	Effort	Age	Line	Tags
Add a private constructor to hide the implicit public one.	Code Smell	Major	Open	Not assigned	5min effort	2 years ago	L17	design
Make this "public static earliestDateTime" field final	Vulnerability	Minor	Open	Not assigned	20min effort	11 months ago	L20	cert, cwe
Make earliestDateTime a static final constant or non-public and provide accessors if needed.	Vulnerability	Minor	Open	Not assigned	10min effort	11 months ago	L20	cwe

Issue Detail

- Show source code
- Show rule and advice to fix
- Can add comment
  - ex: False Positive
- Can assign to developer



1 / 705 issues

beakerx/.../tablesaw/beakerx/TablesawDis...

Add a private constructor to hide the implicit public one.
Code Smell

Make this anonymous inner class a lambda
Code Smell

core/.../tablesaw/aggregate/AggregateFun...

Rename this generic name to match the regular expression '^[A-Z][0-9]?$'.
Code Smell

Rename this generic name to match the regular expression '^[A-Z][0-9]?$'.
Code Smell

core/.../tablesaw/aggregate/AggregateFun...

```
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package tech.tablesaw.beakerx;
                                                   ben@...
import tech.tablesaw.api.Table;
import com.twosigma.beakerx.jvm.object.OutputCell;
import com.twosigma.beakerx.table.TableDisplay;
import jupyter.Displayer;
import jupyter.Displayers;

import java.util.Map;

public class TablesawDisplayer {
```

Add a private constructor to hide the implicit public one. See Rule — 2 years ago — L25
Code Smell · Major · Open · Not assigned · 5min effort · Comment · design

```
  public static void register() {
    Displayers.register(Table.class, new Displayer<Table>() {
      @Override
      public Map<String, String> display(Table table) {
        new TableDisplay(
```

Utility classes should not have public constructors

Utility classes should not have public constructors squid:S1118

Code Smell · Major · Main sources · design · Available Since Jul 31, 2019 · SonarAnalyzer (Java) · Constant/issue: 5min

Utility classes, which are collections of `static` members, are not meant to be instantiated. Even abstract utility classes, which can be extended, should not have public constructors.

Java adds an implicit public constructor to every class which does not define at least one explicitly. Hence, at least one non-public constructor should be defined.

Noncompliant Code Example

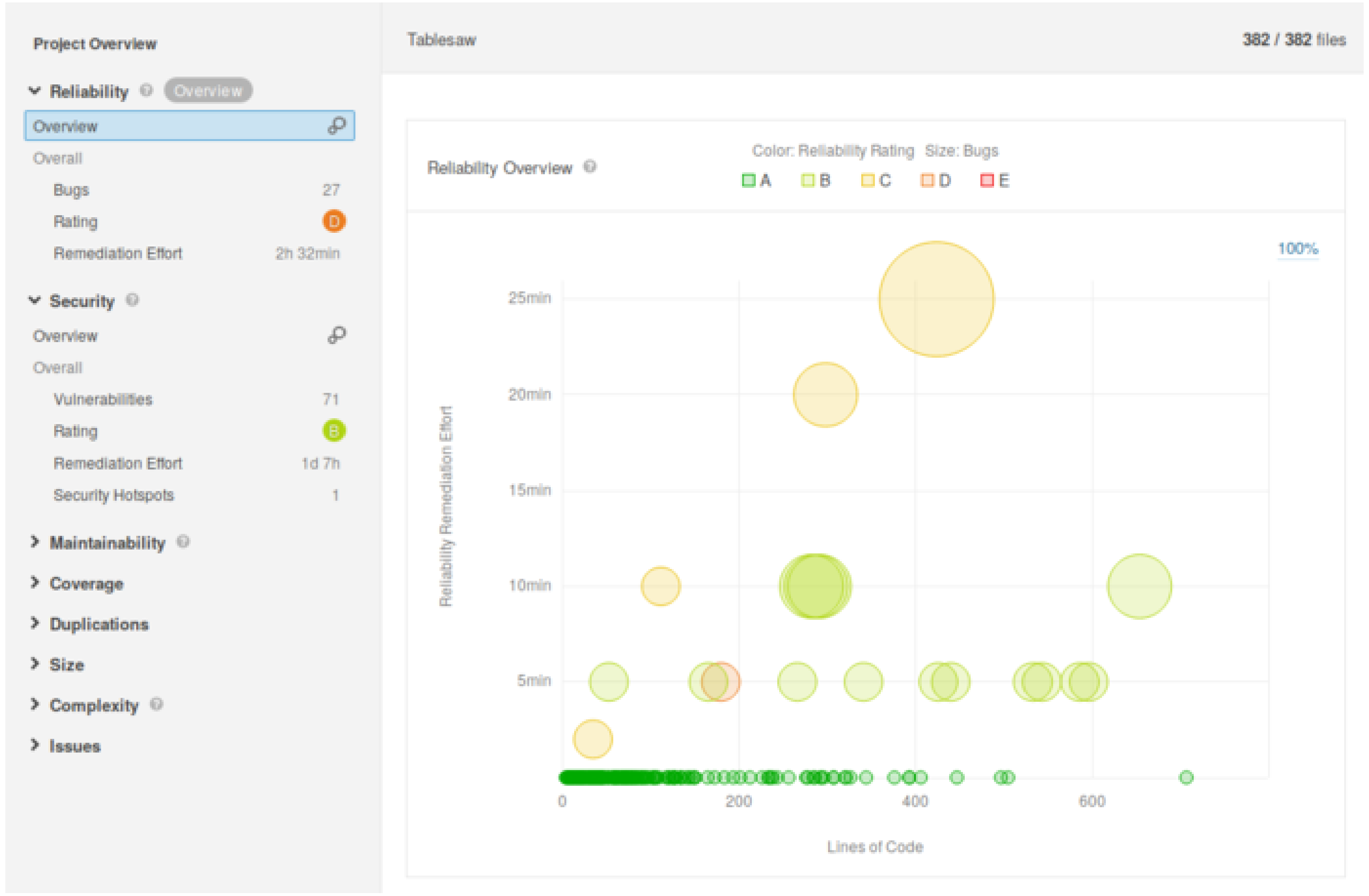
```
class StringUtils { // Noncompliant

  public static String concatenate(String s1, String s2) {
```

Reliability

Security

Maintainability

- Shows files with most issues
- Shows gravity of issues
- Interactive
  - Click on a bubble to jump to the file



Project Overview

Reliability Overview

Overview

Overall

Bugs	27
Rating	D
Remediation Effort	2h 32min

Security

Overview

Overall

Vulnerabilities	71
Rating	B
Remediation Effort	1d 7h
Security Hotspots	1

Maintainability

Coverage

Duplications

Size

Complexity

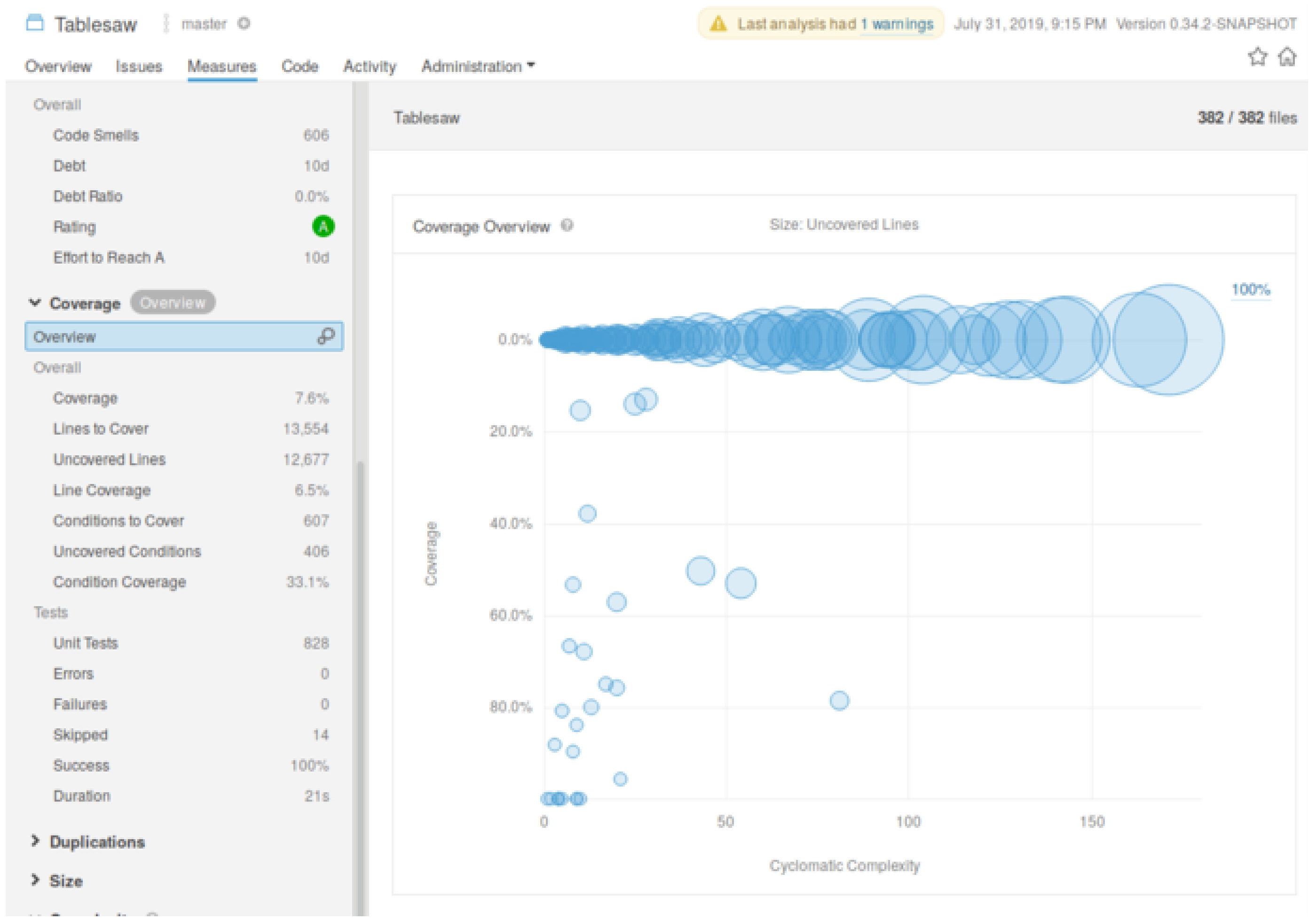
Issues

Tablesaw

382 / 382 files

Code Coverage

- Shows test failures
- Shows files with least coverage
- Interactive
  - Click on a bubble to jump to the file
  - margin green/red indicator for lines covered



Tablesaw | master

Last analysis had 1 warnings   July 31, 2019, 9:15 PM   Version 0.34.2-SNAPSHOT

Overview   Issues   Measures   Code   Activity   Administration

Tablesaw   382 / 382 files

Overall	
Code Smells	606
Debt	10d
Debt Ratio	0.0%
Rating	A
Effort to Reach A	10d
Coverage	Overview
Overview	
Overall	
Coverage	7.6%
Lines to Cover	13,554
Uncovered Lines	12,677
Line Coverage	6.5%
Conditions to Cover	607
Uncovered Conditions	406
Condition Coverage	33.1%
Tests	
Unit Tests	828
Errors	0
Failures	0
Skipped	14
Success	100%
Duration	21s
Duplications	
Size	

Coverage Overview   Size: Uncovered Lines

Complexity

- Cyclomatic Complexity
  - represents how difficult the method is to test
  - based on number of paths through the method
    - simple method (no conditional, loop) = 1
    - method with 1 if statement = 2
  - Best practice is to keep Cyclomatic Complexity below 10
  - Pathological cases: switch statement with multiple return
    - high cyclomatic complexity
    - easy to test
- Cognitive Complexity
  - represents how difficult the method is to understand
  - impact maintainability
  - metrics specific to sonarqube
  - see whitepaper
    - https://www.sonarsource.com/docs/CognitiveComplexity.pdf

Tips to reduce Cyclomatic Complexity

- Extract methods
  - extract complex code fragment into method
  - modified and extracted methods are:
    - easier to understand
    - easier to unit test (separation of concerns)
- complex if conditions
  - `&&` and `||` in statement tests (`if`, `while`) count as additional paths
  - compute complex condition before test
  - somewhat artificial
- Know when to leave it as is
  - switch statement when code is clear
- If you must
  - How to reduce cyclomatic complexity in 15 articles
    - http://codinghelmet.com/articles/reduce-cyclomatic-complexity-null-object

Demo

"The code depicted in this demo is fictitious. Any similarity to actual source code, living or dead is merely coincidental."

Attributions

- This presentation is using the excellent remark framework to convert markdown to HTML slides.