

USEFUL PROGRAM ANALYSIS ECOSYSTEM

Emma Söderberg
Dept. of Computer Science



LUND
UNIVERSITY

LTH
FACULTY OF
ENGINEERING

WASP | WALLENBERG AI,
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

Software@LTH, April 25, 2019

USEFUL PROGRAM ANALYSIS ECOSYSTEM

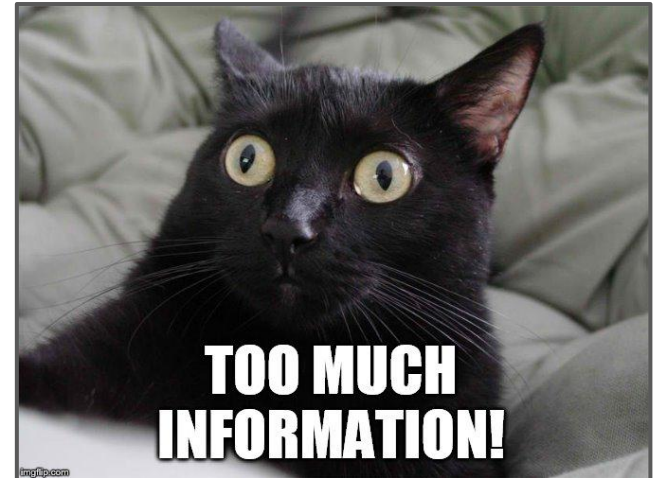
WHAT?

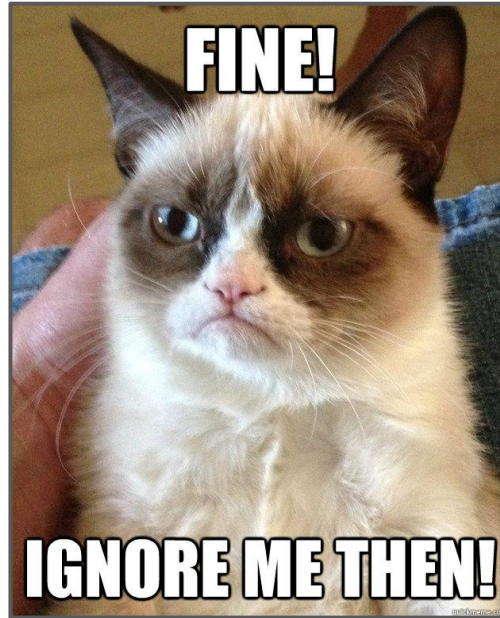


ISN'T PROGRAM ANALYSIS ALWAYS USEFUL?

No, because of ...

- False positives
- Too many results
- Bad integration
- Incomprehensible results



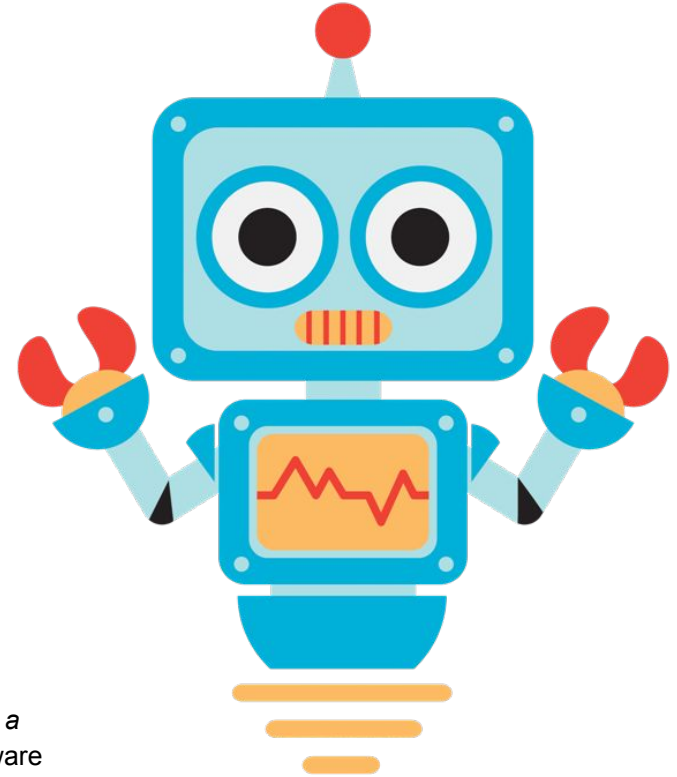


Top question on StackOverflow about static analysis: how to ignore it!

HOW TO MAKE PROGRAM ANALYSIS USEFUL?

Think about the user!

- Static analysis pipelines at Google
 - Integrates analysis results into developer tools
 - Collects feedback
 - Meta-analyzer system
- Tricorder - Internal
- Tricium - Chromium



Sadowski, C., van Gogh, J., Jaspan, C., Söderberg, E., and Winter, C. (2015). *Tricorder: Building a Program Analysis Ecosystem*. In proceedings of ICSE'15: 37th International Conference on Software Engineering.

Söderberg, E. (2016). *Tricium - Tricorder for Chromium*. <https://bit.ly/tricium-early-design>

"THE TRICORDER PRINCIPLES"

- “No” false positives
- Data-driven usability improvements
- Workflow integration
- Empower users to contribute
- Project customization



WHAT DOES IT LOOK LIKE IN CODE REVIEW?

```
package com.google.devtools.statianalysis;
public class Test {
    public boolean foo() {
        return getString() == "foo".toString();
    }
    public String getString() {
        return new String("foo");
    }
}
```

Lint

Missing a Javadoc comment

Java
1:02 AM, Aug 21

[Please fix](#)

[Not useful](#)

• Lint Missing a Javadoc comment

Java
1:02 AM, Aug 21

[Please fix](#)

[Not useful](#)

```
public boolean foo() {
    return getString() == "foo".toString();
}
```

• ErrorProne String comparison using reference equality instead of value equality
(see <http://code.google.com/p/error-prone/wiki/StringEquality>)

StringEquality
1:02 AM, Aug 21

[Please fix](#)

[Not useful](#)

Suggested fix attached: [show](#)

ErrorProne

String comparison using reference equality instead of value equality
(see <http://code.google.com/p/error-prone/wiki/StringEquality>)

StringEquality
1:09 AM, Aug 21

[Please fix](#)

Suggested fix attached: [show](#)

[Not useful](#)

WHAT DOES IT LOOK LIKE IN CODE REVIEW?

ErrorProne

StringEquality
1:09 AM, Aug 21

String comparison using reference equality instead of value equality
(see <http://code.google.com/p/error-prone/wiki/StringEquality>)

[Please fix](#)

Suggested fix attached: [show](#)

[Not useful](#)

```
package com.google.devtools.staticanalysis;
public class Test {
    // ...
}
ErrorProne
StringEquality
1:09 AM, Aug 21
Please fix
Suggested fix attached
```

```
package com.google.devtools.staticanalysis;

public class Test {
    public boolean foo() {
        return getString() == "foo".toString();
    }

    public String getString() {
        return new String("foo");
    }
}
```

```
package com.google.devtools.staticanalysis;
package java.util.Objects;

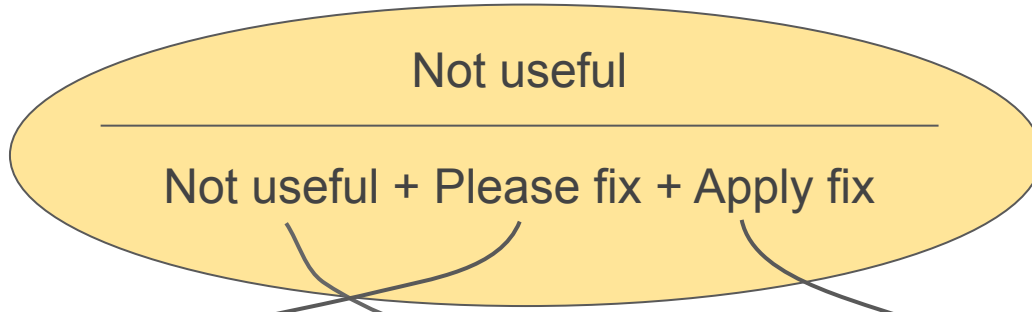
public class Test {
    public boolean foo() {
        return Objects.equals(getString(), "foo".toString());
    }

    public String getString() {
        return new String("foo");
    }
}
```

Apply

Cancel

THE NOT-USEFUL RATE



ErrorProne
StringEquality
1:09 AM, Aug 21

String comparison using reference equality instead of value equality
(see <http://code.google.com/p/error-prone/wiki/StringEquality>)

[Please fix](#)

Suggested fix attached: [show](#)

[Not useful](#)

```
package com.google.devtools.staticana

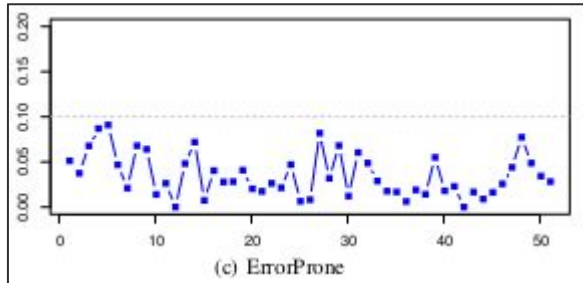
public class Test {
    public boolean foo() {
        return getString() == "foo".toSt
    }

    public String getString() {
        return new String("foo");
    }
}
```

Apply Cancel

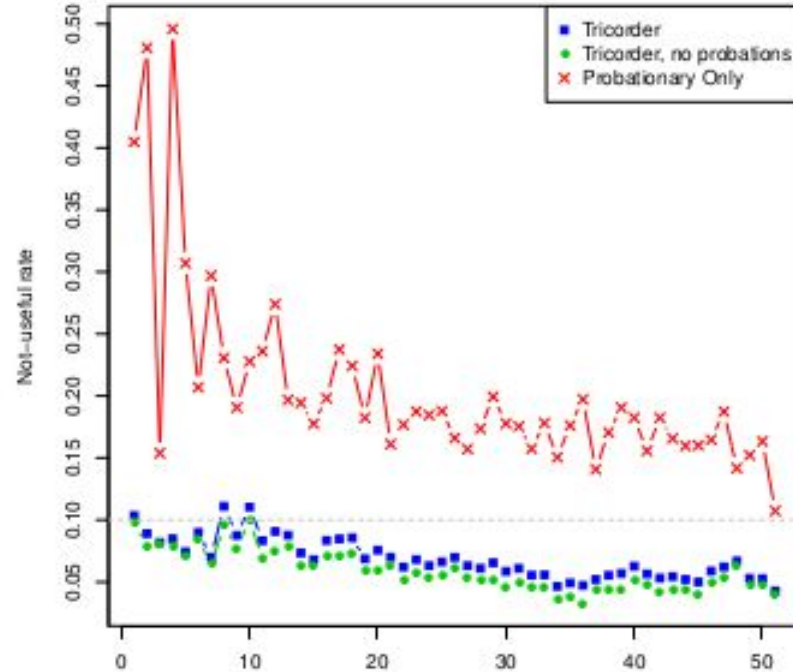
THE NOT-USEFUL RATE

The ErrorProne Analyzer



Weeks in 2014

The Tricorder System



Weeks in 2014

FUN TRICORDER RESULTS

Enabled in
Tricorder

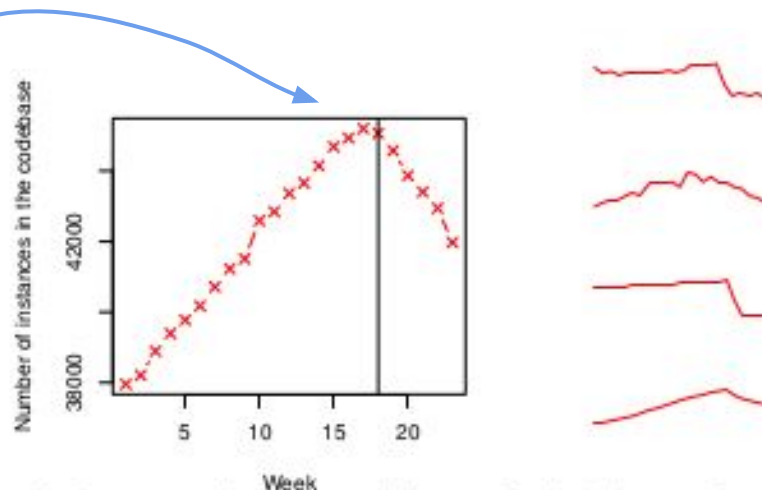
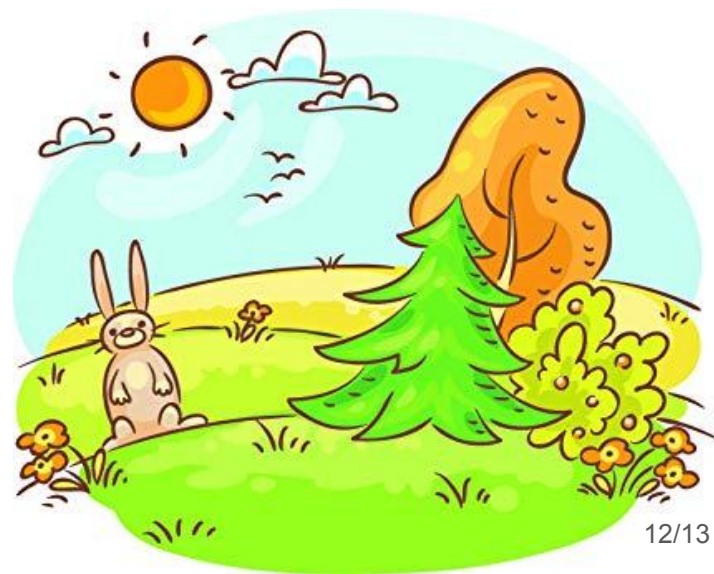


Fig. 8: Instances in the codebase of violations of several ClangTidy checks. The large graph shows the number of violations in the codebase of the check `misc-redundant-smartptr-get`, the smaller graphs show similar trends for other ClangTidy Checks.

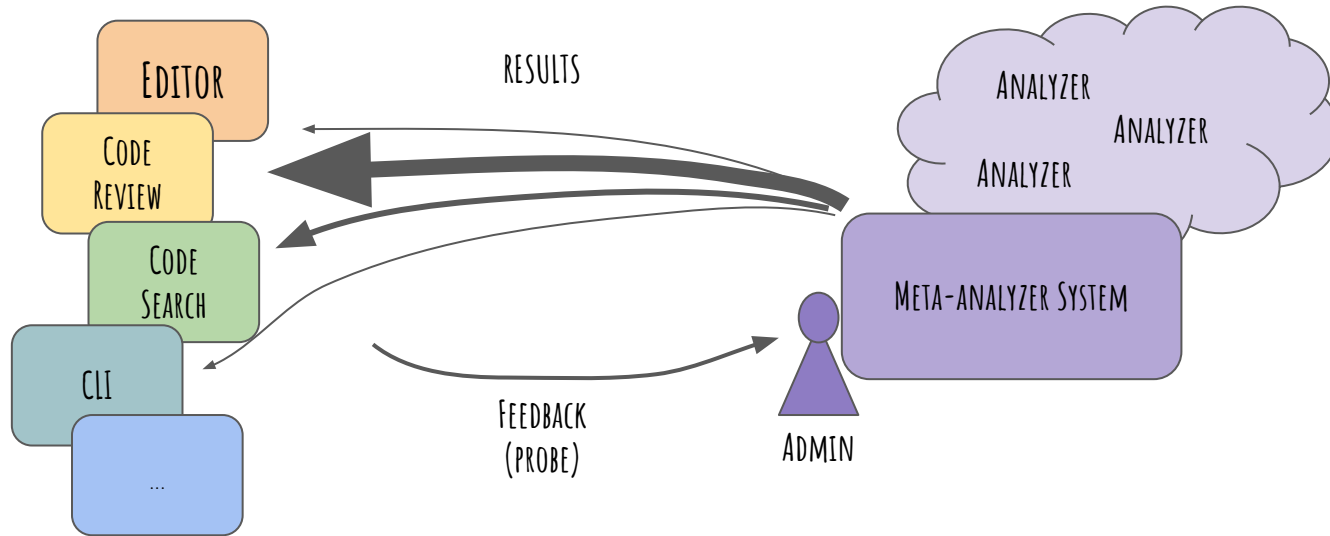
MAKING IT INTO AN ECOSYSTEM

Users - System - Contributors

- Make it easy to contribute
- Enforce analyzer contracts
 - 10% not-useful rate
 - Naughty list
 - Be responsive
- 100+ analyzers, most maintainers outside



AN ADAPTIVE SYSTEM



Going further:

- Not just probes, sensors
- Automation