You’re working too hard!

Can you simplify your life with static testing?

Chris Cowell
Senior Instructor, GitLab
PNSQC 2021
README for Chris

- Former developer & QA for ~20 years
  Accenture, Cambia, Jive, Puppet, Oracle
- Now an instructor at GitLab
- 4-time presenter at PNSQC
- Portland, Oregon
- Picture on NY Times obits page twice
Agenda

1. What is static testing?
2. TLDR
3. How do you perform static testing?
4. Areas static testing can improve
5. Areas static testing can’t improve
6. Downsides
7. Summary and recommendations
1. What is static testing?
What is static testing?

- A.K.A. “static code analysis” or “linting”
- Analyzes source code for problems
- In contrast to dynamic testing, which analyzes running code
- Also scans non-code files
- Categorizes problems by severity
- Might fix automatically
- Can diff results on default and non-default branches
2. TLDR
The lure of static testing

Can I replace these with a tool that requires no effort from me?

- Functional tests
- Performance tests
- Security tests

The answer is...
Static testing doesn’t replace dynamic testing.

It finds risks, not bugs.

Try it.
3. How do you perform static testing?
How do you perform static testing?

- Many options
  - Within IDE (IntelliJ IDEA)
  - Build tool plugin (Maven, Gradle, Rake)
  - CI/CD pipeline plugin (GitLab, Travis, Circle CI)
  - SaaS repo plugin (GitLab, GitHub, Bitbucket)
  - Browser extensions
  - CLI
- Free, paid, or hybrid
Some of the big players

Bandit
Code Climate
Coverity
LGTM
PMD
Semgrep
SpotBugs (nee FindBugs)
4. Areas static testing can improve
It depends on the tool and language.
Best practices

```java
import java.lang.String;

class Foo {
    String bar(boolean qux) {
        if (qux)
            return "Hello";
        else
            return "Hi";
    }
}
```

- unnecessary
- “if” statement needs braces
- use only 1 exit point
Naming and formatting conventions

1. **capitalize class names**
   ```java
class foo {
    boolean boolBar() { ... }
    void qux() { ... }
}
```

2. **omit data types from names**
3. **name boolean methods “isXYZ”**
4. **1 blank line between methods**
Duplication

```python
def foo():
    print("Hi")

def bar():
    print("Hi")

if qux:
    i += 1
else:
    i += 1
```

unnecessary method
unnecessary conditional
Unreachable code

```python
def foo():
    isBar = False
    if isBar:
        print("Hello")
    return True
    print("Hi")
```

*method never called*

*unreachable*
Performance

1. StringBuffer sb = new StringBuffer();
2. 
3. if (sb.toString().equals("")) {...}
4. if (sb.length() == 0) {...}
5. 
6. StringBuilder sb = new StringBuilder();
Security

1  require 'digest'
2  Digest::MD5.hexdigest 'abc'
3
4  file_name = gets
5  file_type = system("file #{file_name}")

weak hash
open to command injection
Complexity

```python
if i > j:
    if k > 15 or i < 1:
        ...
    else:
        ...
else:
    if i < 5 and j > 2:
        ...
    else:
        if not foo or k > 2:
            ...
        elif j > 15 and k > j:
            ...
        else:
            ...
```
Documentation

1 /**
2 * Add 2 integers
3 * @param i first addend
4 * @return
5 */
6 int sum(int i, int j) {
7    return i + j;  // TODO: is this right?
8 }
9 /** Guys, do we even need this method? */
Spelling and grammar

String optionsMsg = "Review all of you're options by June 1 2022";
5. Areas static testing can’t improve
Areas static testing can’t improve

- Algorithm doesn’t meet requirements
  - sort by last name instead of first name
- Correct algorithm, incorrect implementation
  - off-by-1 error
  - reversed arguments in a method call
  - logic error
- Failure to meet performance goals
- Integration errors
- Configuration errors
6. Downsides
Downsides to static testing

- False positives
- Slow and expensive to triage
- Slow to run
- No standard reporting format
7. Summary and recommendations
Summary and recommendations

Static testing doesn’t replace dynamic testing
- Blind to many categories of problems

Finds risks, not bugs
- Removing risks improves your code, but might take time to pay off

Try it
- Finds some problems before any other testing method
- Part of shifting left (especially with in-IDE tools)
- Tune your tools to limit false positives
- Triage ruthlessly
Thank you!

Go forth and test statically
Q & A

Chris Cowell
christopher.w.cowell@gmail.com
Photo credits

- Plasma ball: Image by [Milesl](https://www.pxfuel.com/en/free-photo-xjtgj) from Pixabay
- boy with static hair: https://commons.wikimedia.org/wiki/File:Static_slide.jpg, Creative Commons Attribution 2.0 Generic License
- facepalm: https://commons.wikimedia.org/wiki/File:Paris_Tuileries_Garden_Facepalm_statue.jpg, Creative Commons Attribution 2.0 Generic
- Dictionary: https://freesvg.org/dictionary-image, public domain
- At Your Own Risk: https://commons.wikimedia.org/wiki/File:AT_YOUR_OWN_RISK.svg, public domain
- Hello My Name is: https://commons.wikimedia.org/wiki/File:Emojione_1F4DB.svg, Creative Commons Attribution-Share Alike 4.0 International
- Twins: https://commons.wikimedia.org/wiki/File:The_McClure_Twins.jpg, Creative Commons Attribution-Share Alike 4.0 International
- Do Not Enter sign: https://pixabay.com/photos/do-not-enter-sign-traffic-road-600755/, Pixabay license
- Turtle: https://commons.wikimedia.org/wiki/File:Florida_Box_Turtle_Digon3.jpg, Creative Commons Attribution-Share Alike 3.0 Unported
- Sloth: https://commons.wikimedia.org/wiki/File:Cute_Sloth.jpg, Creative Commons CC0 1.0 Universal Public Domain Dedication
- Maze: https://freesvg.org/the-maze, Public Domain
- RTFM: https://commons.wikimedia.org/wiki/File:Demotivational_RTFM_Prudentia_Statue_on_the_Brussels_town_hall_(53522136).jpg, Creative Commons Attribution 2.0 Generic
- static boy on slide: https://commons.wikimedia.org/wiki/File:Static_slide.jpg, Creative Commons Attribution 2.0 Generic license