

# Lecture 9: Software Quality in Practice

17-313: Foundations of Software Engineering

Rohan Padhye, Michael Hilton, Chris Timperley, and Daye Nam

# Administrivia

- Missing lecture policy (see Syllabus on website)
- HW2: In-person presentation during this week's recitation
- HW3A: plan due on Thursday
  - Implementation due October 6th
  - Reflection due October 13th
- Midterm is on October 11th
  - review will take place in the week before the midterm

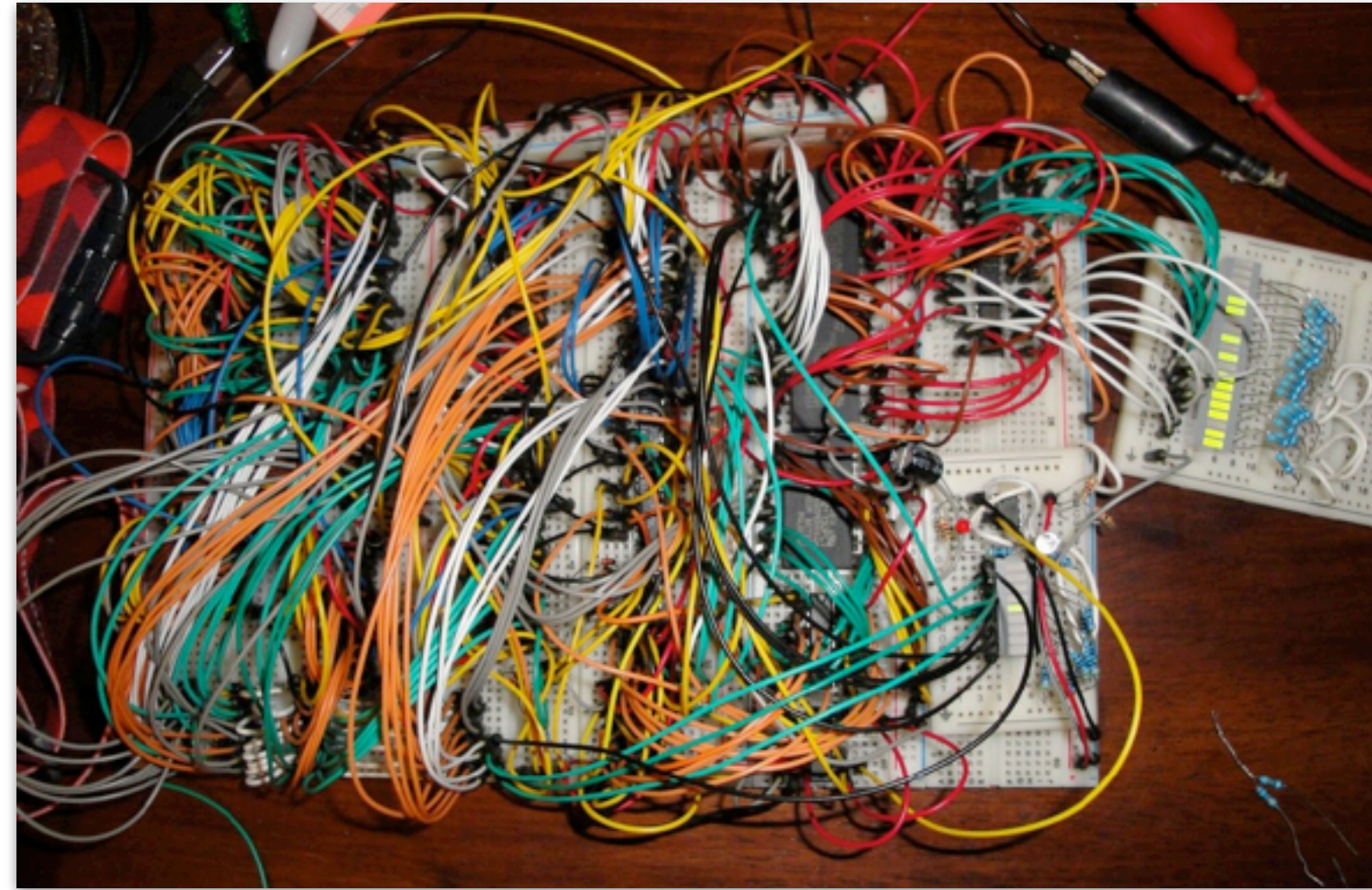
# Learning Goals

- Understand the concepts of software quality and technical debt
- Reflect on personal experiences of technical debt
- Learn best practices for proactively ensuring quality
- Design an explicit QA process for your project
- Learn techniques for reactively dealing with quality problems

# Software Quality



## Internal Quality



- Is the code well structured?
- Is the code understandable?
- How well tested is the code?

## External Quality

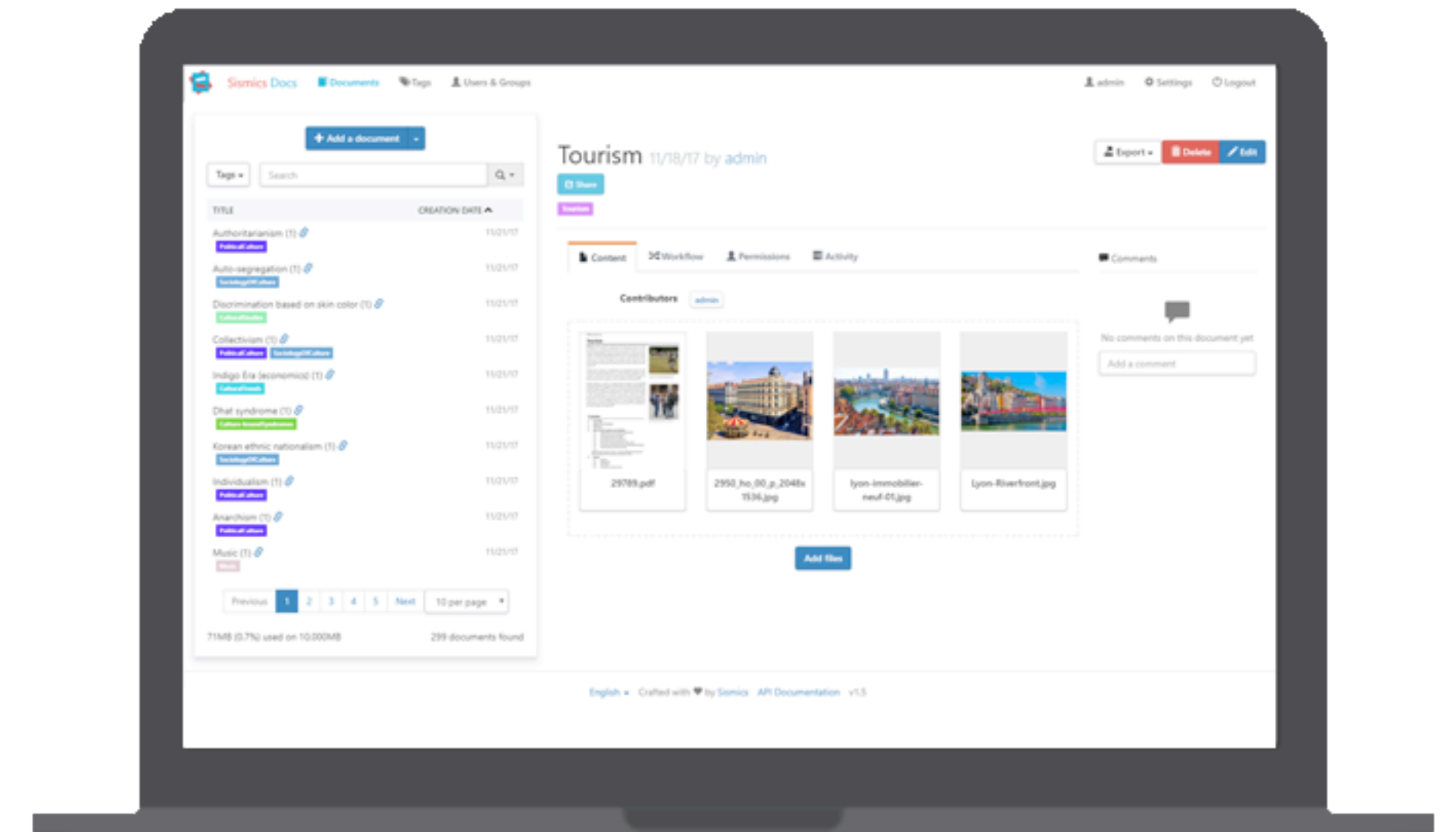


- Does the software crash?
- Does it meet the requirements?
- Is the UI well designed?

# How would you rate Teedy?

teedy

The screenshot shows the GitHub repository page for `sismics/docs`. The repository is public and has 1.3k stars, 41 watchers, and 216 forks. It is currently on the `master` branch with 2 other branches and 11 tags. The repository description states: "Lightweight document management system packed with all the features you can expect from big expensive solutions". The repository includes a README, a license (GPL-2.0), and 11 releases, with the latest being `v1.10` on Jan 2. The file list includes `.github`, `docs-android`, `docs-core`, `docs-importer`, `docs-web-common`, `docs-web`, `.gitattributes`, `.gitignore`, `CODE_OF_CONDUCT.md`, `COPYING`, `Dockerfile`, `README.md`, `docs.xml`, and `pom.xml`.

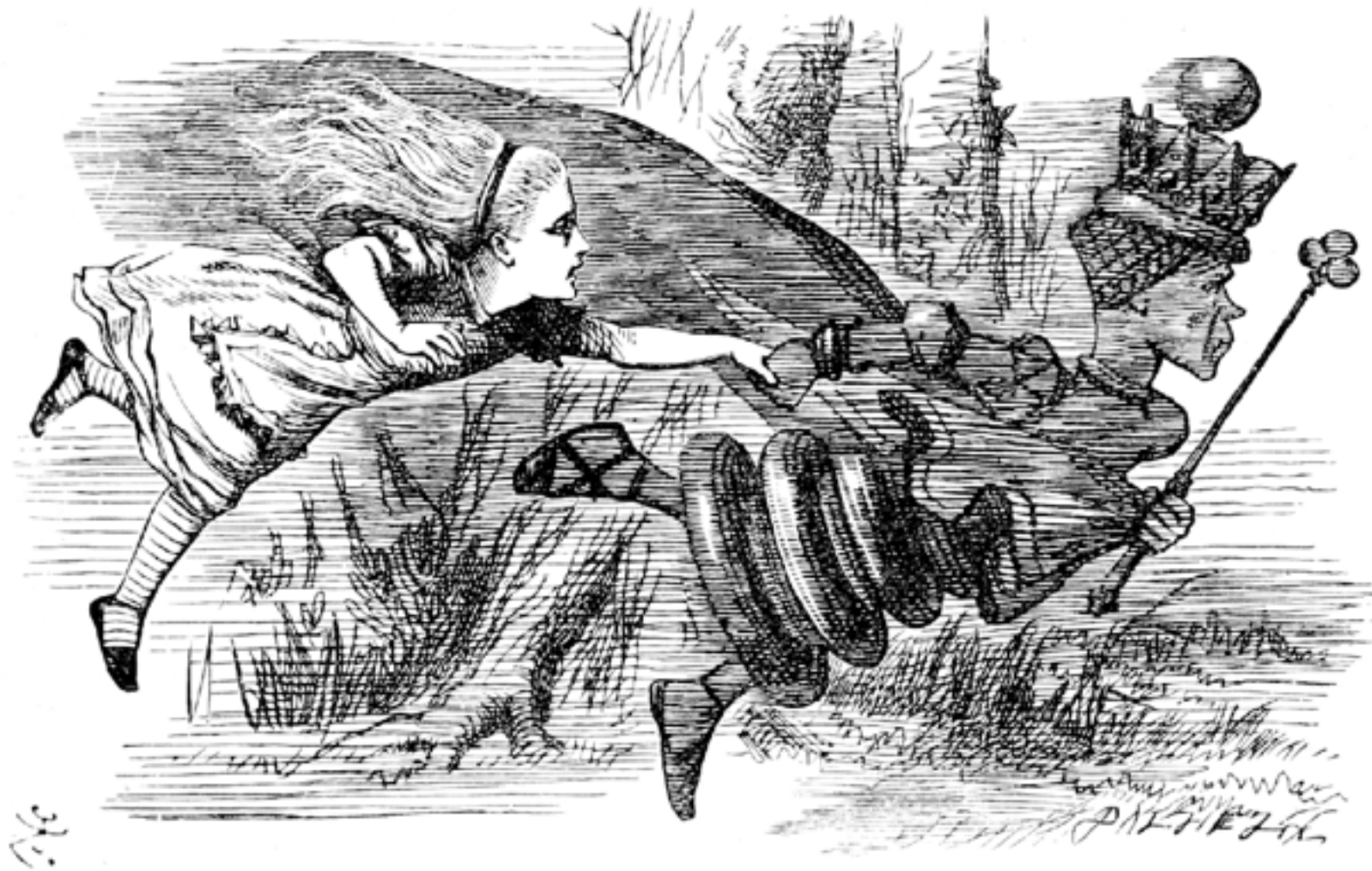


**How did we get here?**

# Software entropy

“As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.”

*Meir Manny Lehman*



“Now, here, you see, it takes all the running you can do just to keep in the same place. If you want to get somewhere else, you must run at least twice as fast!”

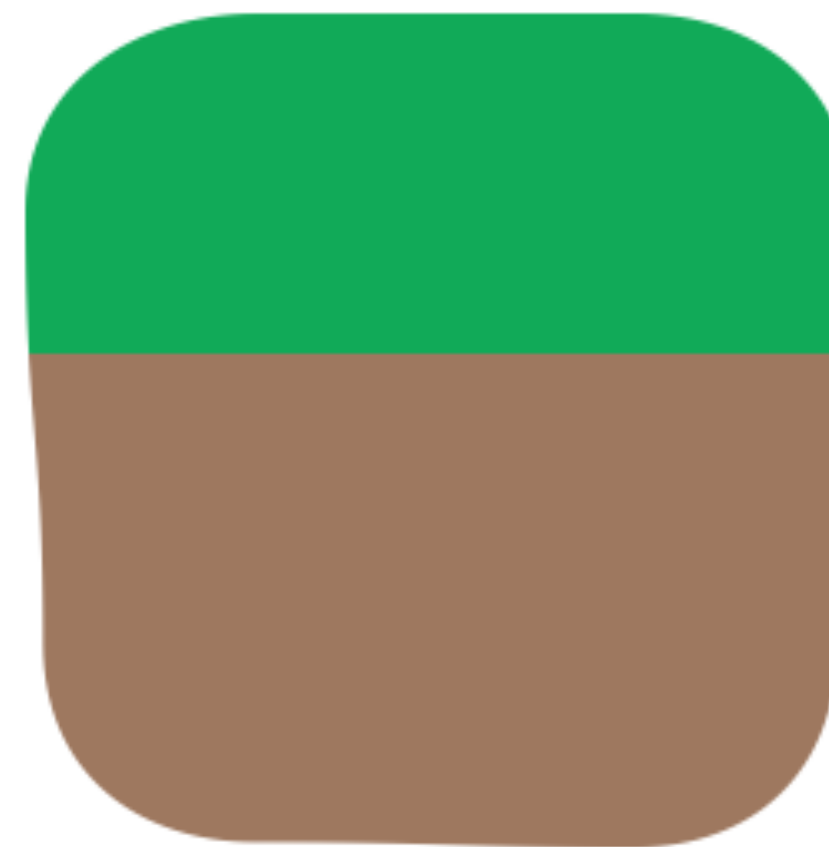
*Through the Looking Glass*



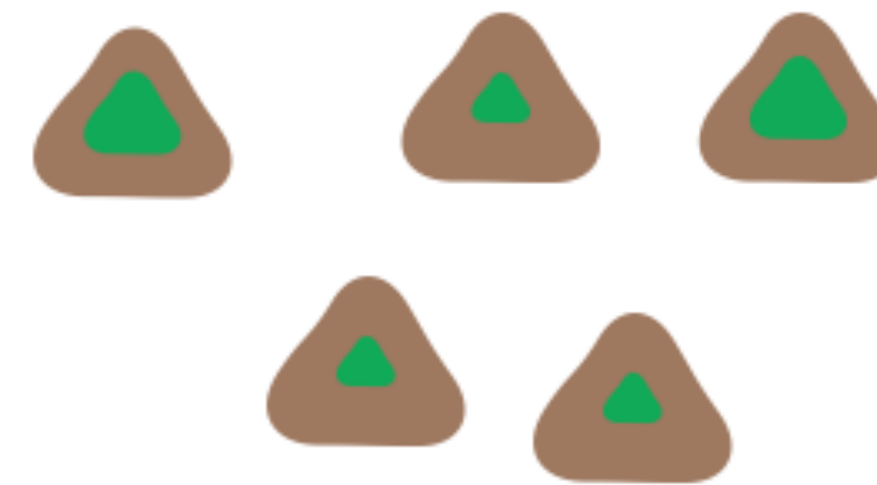
# Technical debt

Any software system has a certain amount of **essential** complexity required to do its job...

... but most systems contain **crudt** that makes it harder to understand.



Crudt causes changes to take **more effort**



The technical debt metaphor treats the crudt as a debt, whose interest payments are the extra effort these changes require.

<https://martinfowler.com/bliki/TechnicalDebt.html>

# A better analogy: Pollution

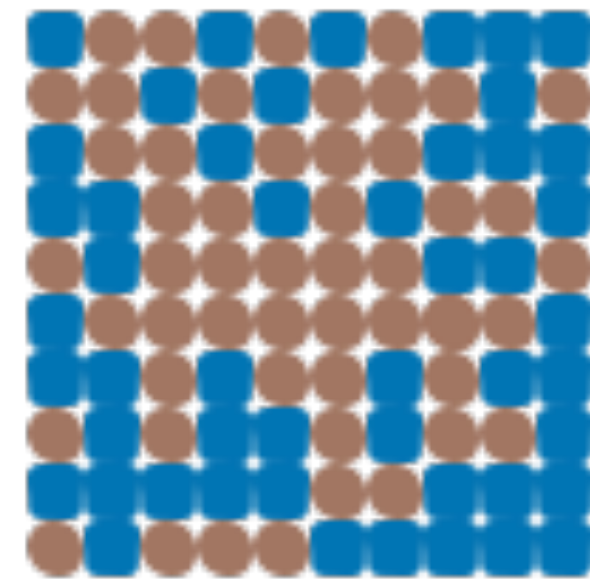


AQI Basics for Ozone and Particle Pollution			
Daily AQI Color	Levels of Concern	Values of Index	Description of Air Quality
Green	Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Yellow	Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Red	Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Purple	Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

<https://www.airnow.gov/aqi/aqi-basics>

# Internal quality makes it easier to add features

*If we compare one system with a lot of cruft...*

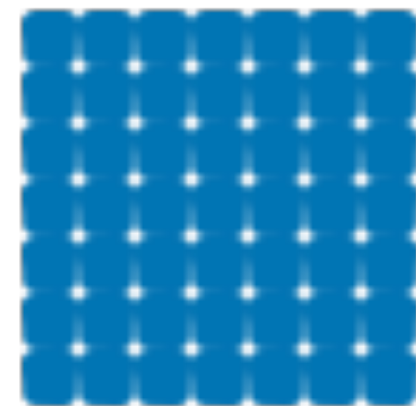


*the cruft means new features take longer to build*



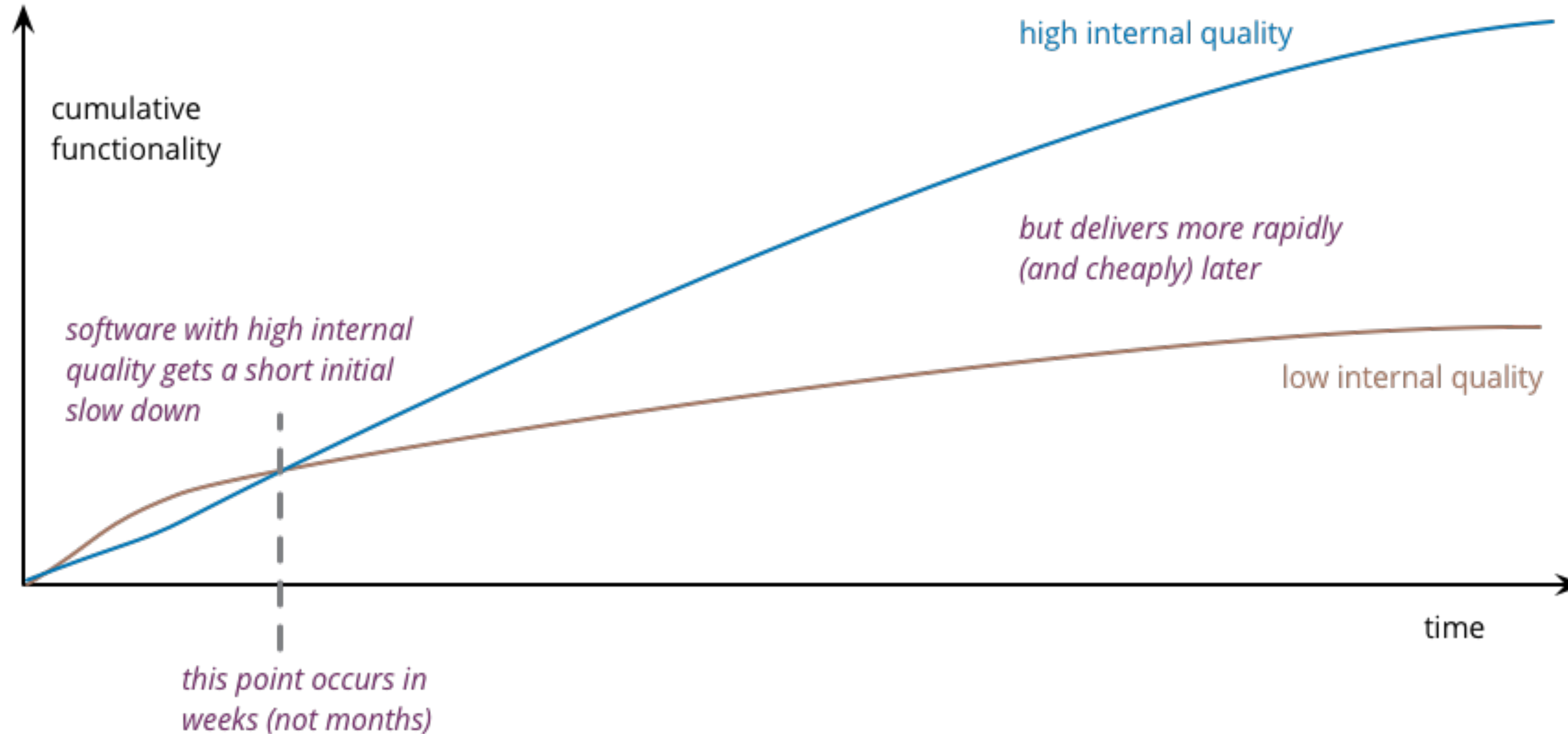
*this extra time and effort is the cost of the cruft, paid with each new feature*

*...to an equivalent one without*



*free of cruft, features can be added more quickly*

# High internal quality is an investment



# What causes technical debt?

# What causes technical debt?

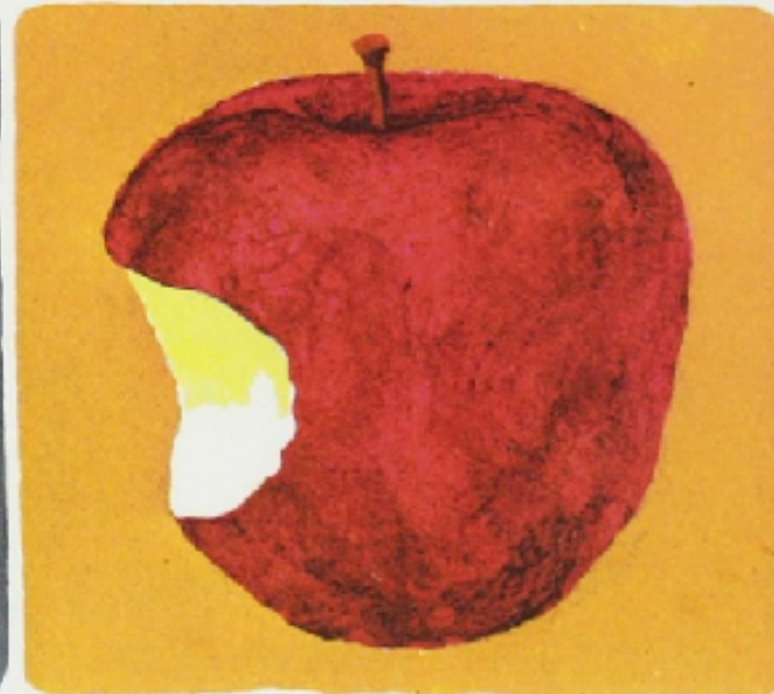
- Tightly-coupled components
- Poorly-specified requirements
- Business pressure
- Lack of process
- Lack of documentation
- Lack of automated testing
- Lack of knowledge
- Lack of ownership
- Delayed refactoring
- Multiple, long-lived development branches
- ...

# Not all technical debt is the same

	<b>Reckless</b>	<b>Prudent</b>
<b>Deliberate</b>	<i>“We don’t have time for design”</i>	<i>“We must ship now and deal with consequences (later)”</i>
<b>Inadvertent</b>	<i>“What’s layering?”</i>	<i>“Now we know how we should have done it”</i>

<https://martinfowler.com/bliki/TechnicalDebtQuadrant.html>

# EVERYONE CREATES TECHNICAL DEBT





# Bad: Too much technical debt

- Bad code can be demoralizing
- Conversations with the client become awkward
- Team infighting
- Atrophied skills
- Turnover and attrition
- ...



**When should we reduce technical debt?**

# Dealing with technical debt: Fixing broken windows



# Alternative: Putting out fires is expensive!



# Analogy: Cleaning your dryer



**How can we avoid technical debt?**

# Reflection: Homework 2

17-313 Foundations of SOFTWARE ENGINEERING

ASSIGNMENTS  
HW01 HELLO TEEDY  
HW02 LEARNING THE CODEBASE  
HW03 TEAMWORK

RECITATIONS  
CALENDAR  
SCHEDULE  
STAFF  
SYLLABUS

Search 17-313

Assignments / HW02 Learning the Codebase

## 17-313: FOUNDATIONS OF SOFTWARE ENGINEERING

### HOMEWORK 2: LEARNING THE CODEBASE

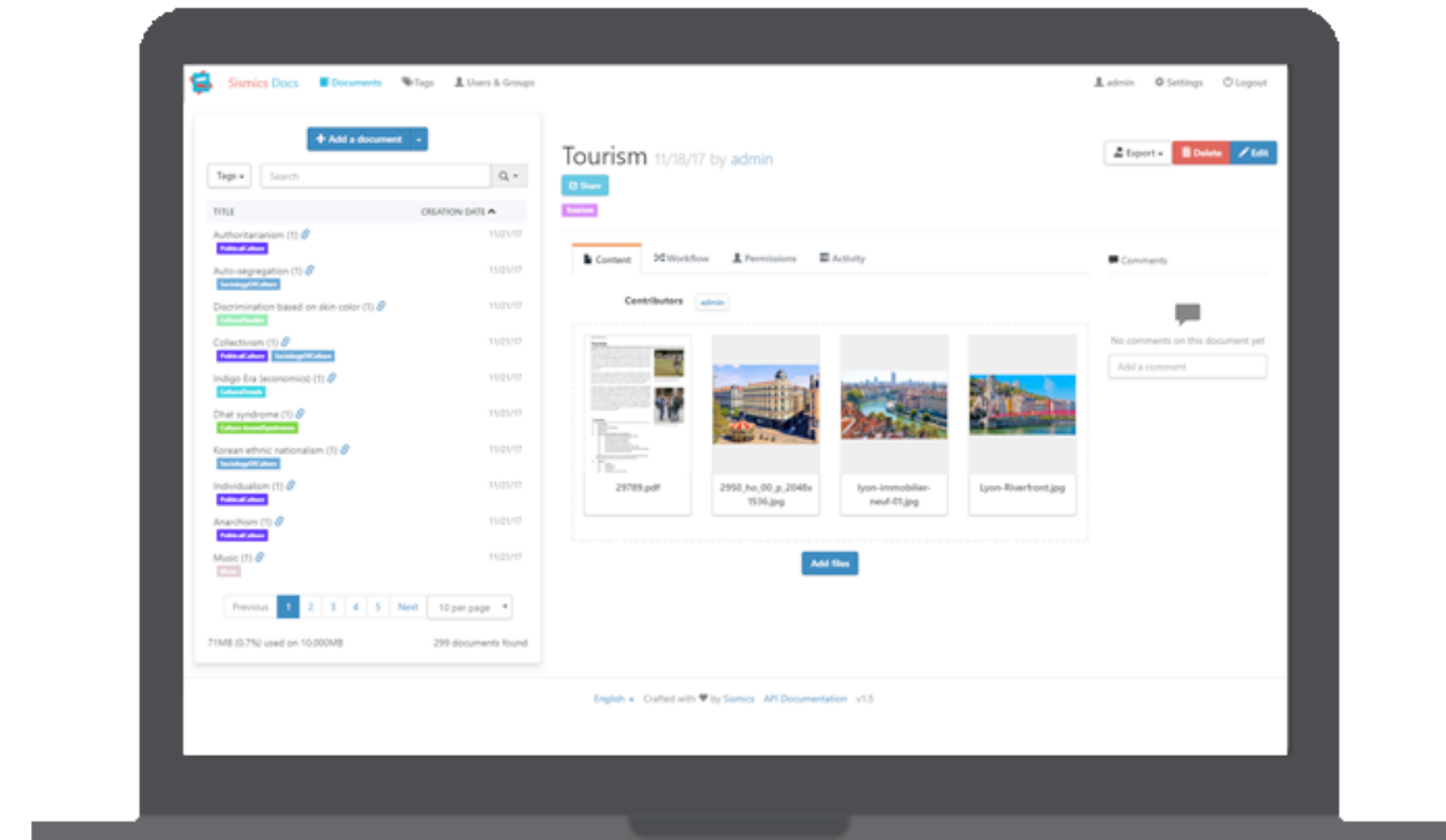
#### LEARNING GOALS

- Learn about the design and implementation of an existing software project that you will build upon this semester
- Collaborate with your teammates to design and implement a small feature addition or modification to that software project
- Practice version control and development best practices within the context of a group assignment

#### SPECIFICATION

Now that you have onboarded to 17-313, setup your development environment, and met your team, upper management has tasked you with building a graduate student admissions system over the course of the semester (described below). Rather than creating a new admissions system from scratch, management has decided that you and your team will repurpose and adapt an existing document management system, Teedy. Beyond being a pretty good document management platform, Teedy provides complex features that you expect will be useful, like tagging, workflows, and fine-grained permissions and ACLs (access control lists).

teedy



# Common Anti-Patterns

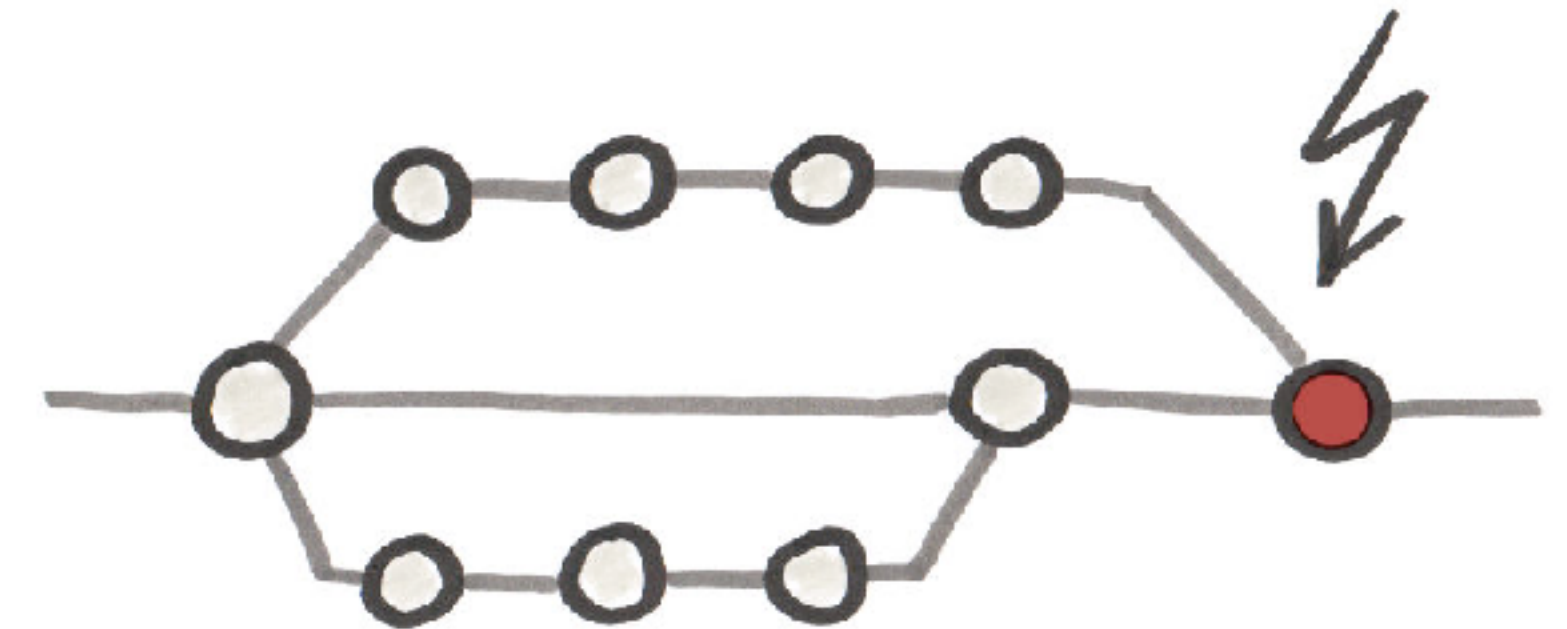
- Not having a QA process! Or no-one follows it





# Common Anti-Patterns

- Not having a QA process! Or no-one follows it
- Bad version control practices
  - Everyone commits to the main branch
  - Long-lived feature branches
  - Huge PRs



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

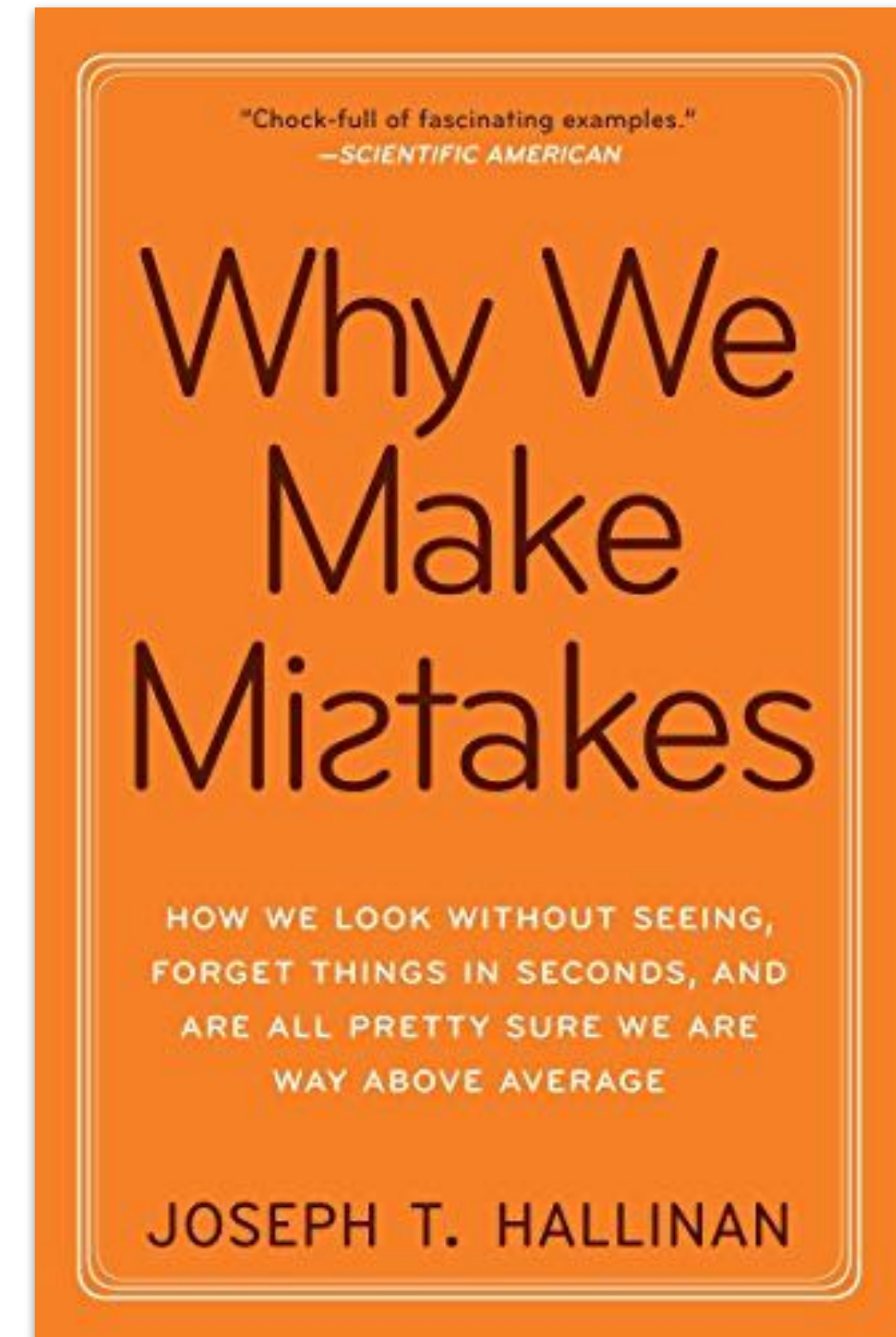
# Common Anti-Patterns

- Not having a QA process! Or no-one follows it
- Bad version control practices
- Slow and encumbering QA processes
  - changes take forever to get merged
  - time could be better spent on new features



# Common Anti-Patterns

- Not having a QA process! Or no-one follows it
- Bad version control practices
- Slow and encumbering QA processes
- Reliance on repetitive manual labor
  - focused on superficial problems rather than structural ones
  - results may vary (e.g., manual testing)
  - **mistakes will happen!**



# Case Study: Knight Capital

## Knightmare: A DevOps Cautionary Tale

D7 DevOps April 17, 2014 6 Minutes

I was speaking at a conference last year on the topics of DevOps, Configuration as Code, and Continuous Delivery and used the following story to demonstrate the importance making deployments fully automated and repeatable as part of a DevOps/Continuous Delivery initiative. Since that conference I have been asked by several people to share the story through my blog.



In laymen's terms, Knight Capital Group realized a \$460 million loss in 45-minutes. Remember, Knight only has \$365 million in cash and equivalents. **In 45-minutes Knight went from being the largest trader in US equities and a major market maker in the NYSE and NASDAQ to bankrupt.**

# Good: Trunk-Based Development

Small changes are easier to develop, review, and integrate



# Tip: Squash and Merge

## Every commit on main should represent a meaningful change

### Added coverage workflow to GitHub actions #3

Edit <> Code

Merged ChrisTimperley merged 24 commits into main from add\_coverage 23 days ago

Conversation 15 Commits 24 Checks 0 Files changed 4 +66 -60

Commits on Aug 24, 2022

add coverage and run via actions

MichaelHilton committed on Aug 24

5626aee <>

configure multi-modules

MichaelHilton committed on Aug 24

5162da9 <>

Autogenerated JaCoCo coverage badge

YOUR NAME HERE authored and YOUR NAME HERE committed on Aug 24

e673062 <>

Update README.md

MichaelHilton committed on Aug 24

Verified

b529f0b <>

merge main changes

MichaelHilton committed on Aug 24

Merge branch 'add\_coverage' of github.com:CMU-313/Teedy into add\_cove...

MichaelHilton committed on Aug 24

Autogenerated JaCoCo coverage badge

YOUR NAME HERE authored and YOUR NAME HERE committed on Aug 24

All checks have passed

1 successful check

Show all checks

This branch has no conflicts with the base branch

Merging can be performed automatically.

Squash and merge

You can also open this in GitHub Desktop or view command line instructions.

Commits on Sep 3, 2022

Added coverage workflow to GitHub actions (#3)



MichaelHilton committed 23 days ago

Verified

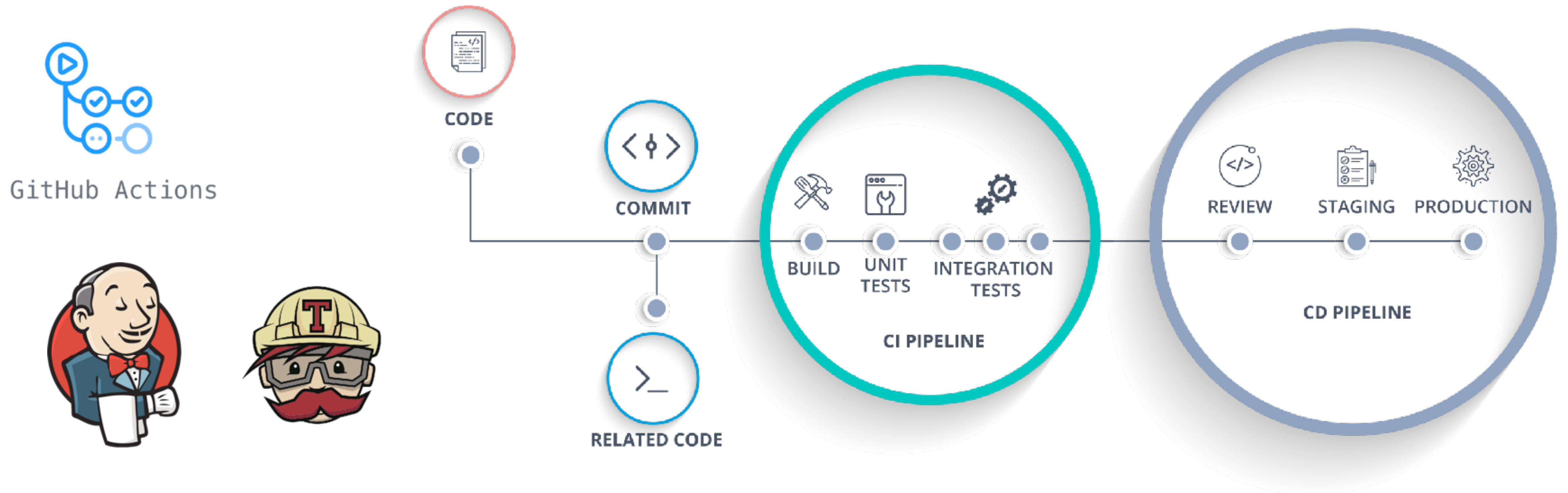


e3a72ad



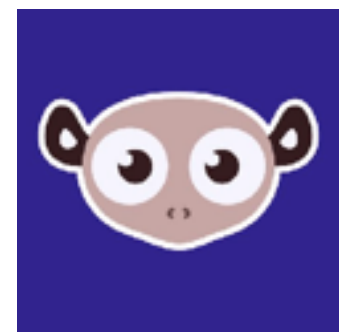
# Good: Continuous Integration and Deployment

Find and fix mistakes as early as possible when they're easier to address



# Good: Automate as much as possible!

Humans are expensive; compute time is cheap



: my[py]

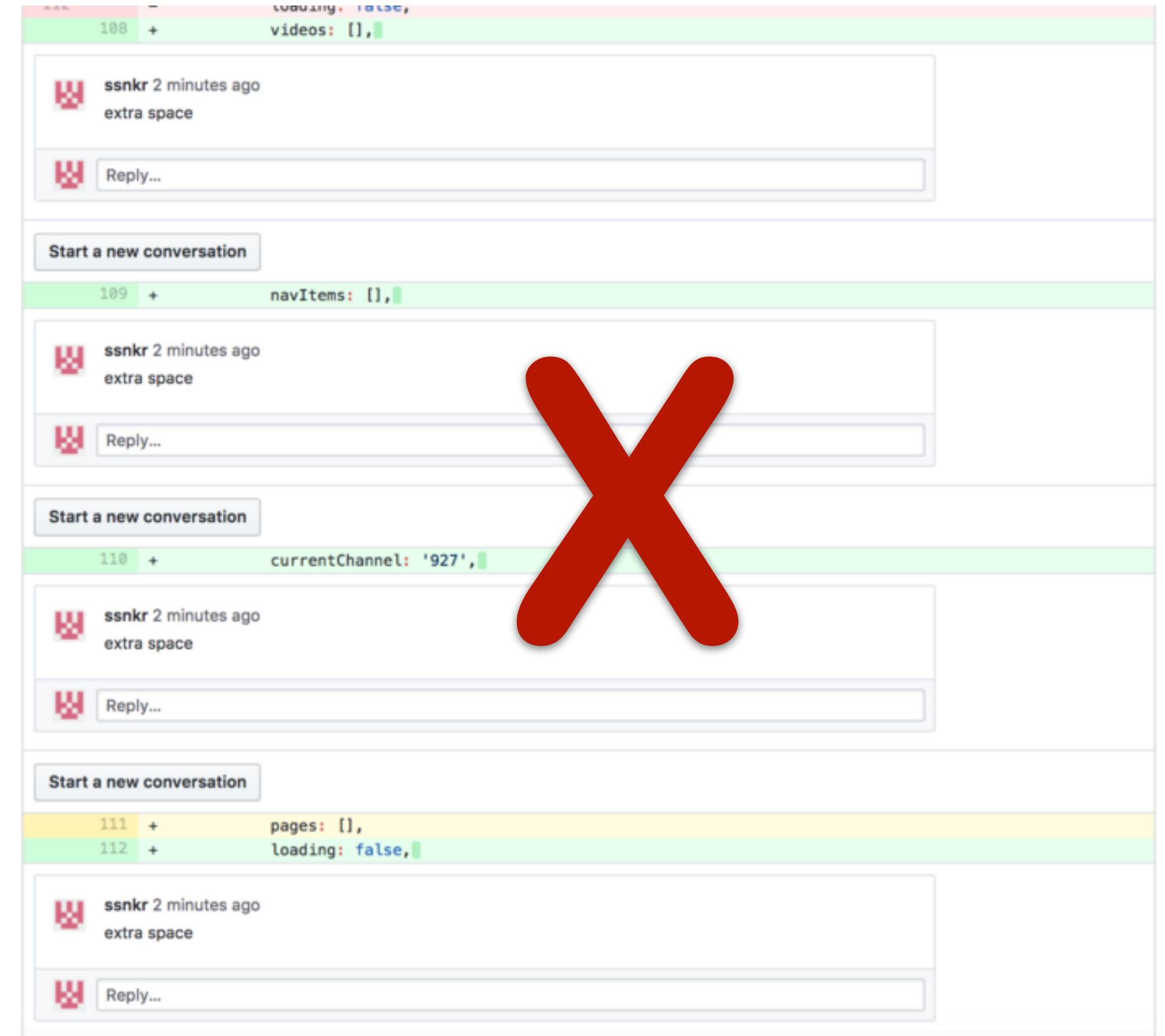


snyk



# Good: Let humans review the hard stuff!

- **Good:** Is this code understandable? Is this going to be a problem to maintain? Is this going to constrain us in the future? Is there a more elegant / simpler / more efficient solution?
- **Bad:** Reviewers shouldn't run the code or tests or make any changes
- **Bad:** Reviewing superficial changes



# Good: Write automated tests

Manual testing is unreliable and unsustainable

## Automated Testing

- + Reproducible
- + Some upfront effort
- + Zero marginal effort
- + Runs on every commit
- + Finds regressions!

JUnit 

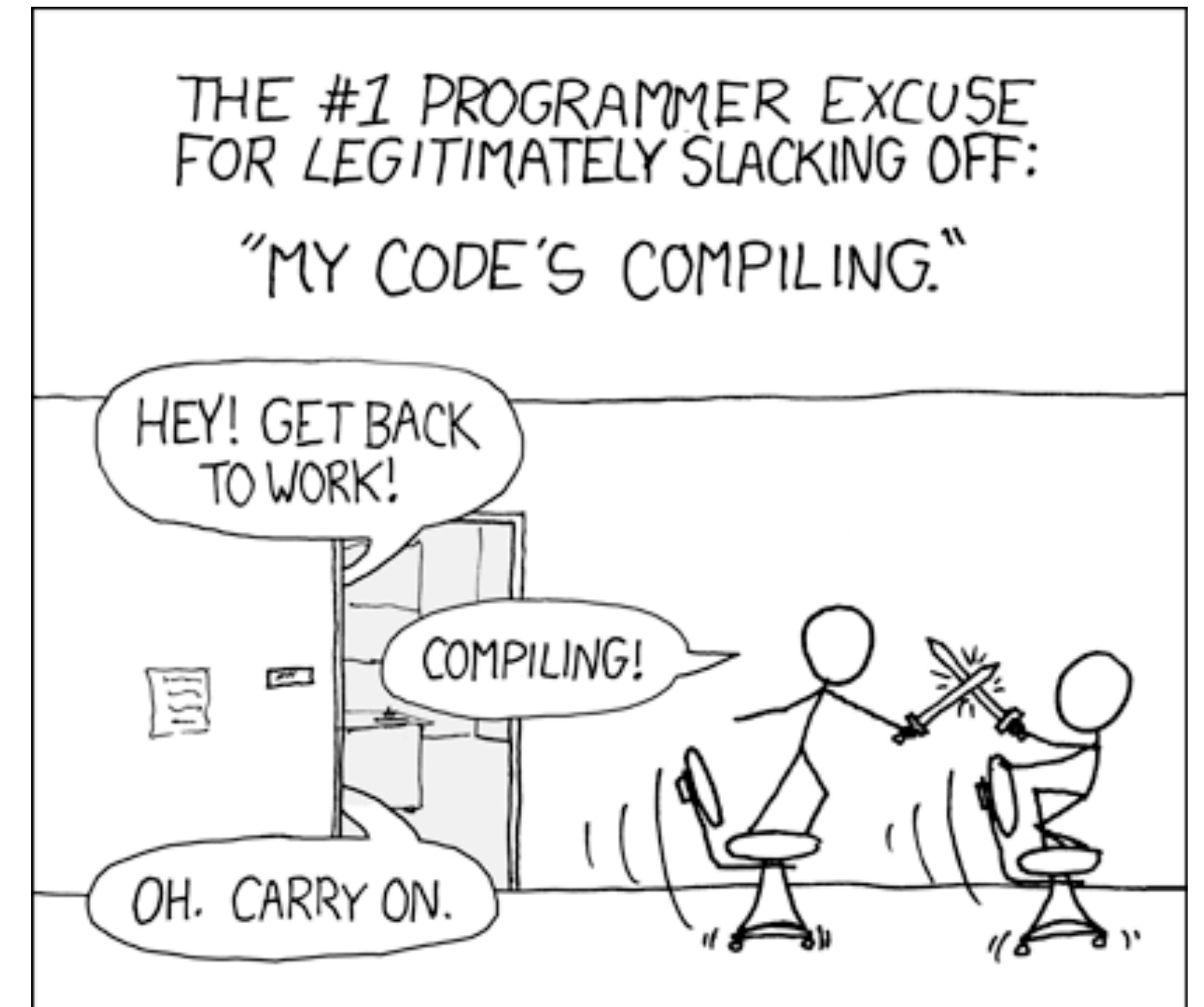
## Manual Testing

- Unreproducible
- Low upfront effort
- High marginal effort
- Runs when you remember
- Unsustainable



# Tips for Effective CI/CD

- Your CI pipeline should run quickly

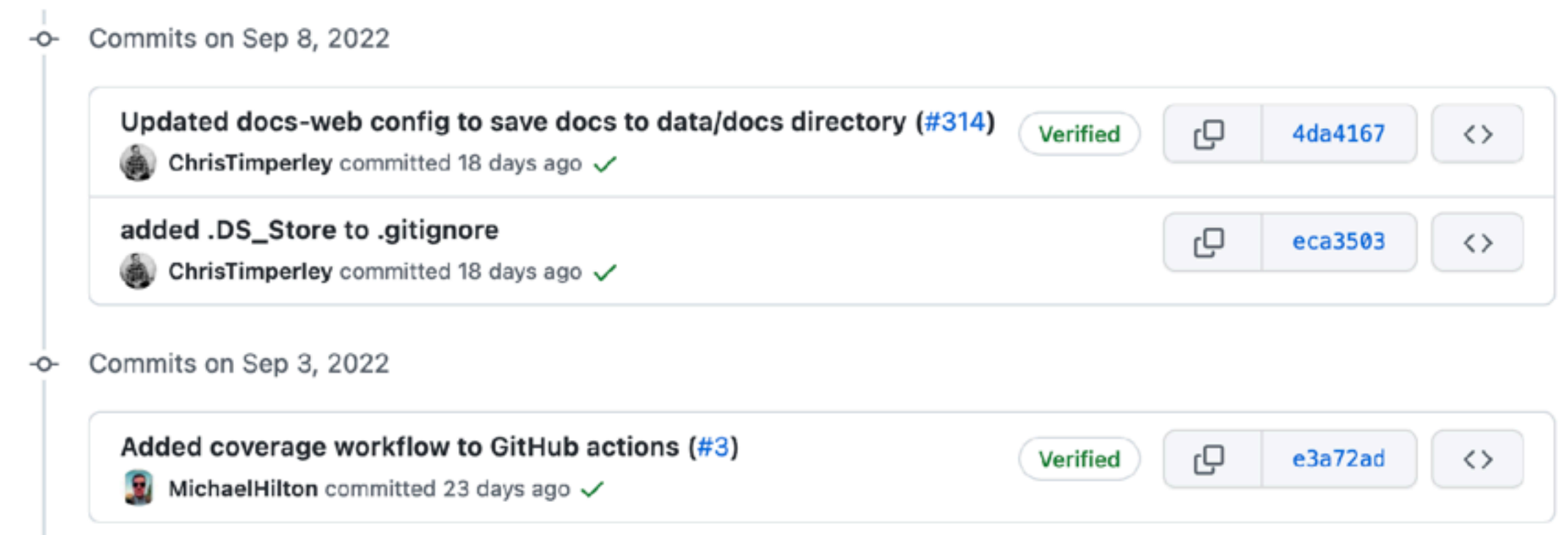


# Tips for Effective CI/CD

- Your CI pipeline should run quickly
- Your CI pipeline should run on every change
  - Only bother the reviewer once all checks pass



A screenshot of a GitHub Actions status card. On the left is a green square icon with a white branching diagram. To its right is a white box with a green border. At the top left of the box is a green circle with a white checkmark. Next to it is the text "All checks have passed" in bold, followed by "1 successful check" in a smaller font. Below this is a grey bar with a green checkmark, a GitHub Actions icon, and the text "Tox lint checking / build (pull\_request) Successful in 1m — build".



A screenshot of a GitHub commit history. It shows a vertical timeline with two sections. The first section is titled "Commits on Sep 8, 2022" and contains two commit entries. The first entry is "Updated docs-web config to save docs to data/docs directory (#314)" by ChrisTimperley, committed 18 days ago, with a green checkmark, a "Verified" badge, a copy icon, the commit hash "4da4167", and a code icon. The second entry is "added .DS\_Store to .gitignore" by ChrisTimperley, committed 18 days ago, with a green checkmark, a copy icon, the commit hash "eca3503", and a code icon. The second section is titled "Commits on Sep 3, 2022" and contains one commit entry: "Added coverage workflow to GitHub actions (#3)" by MichaelHilton, committed 23 days ago, with a green checkmark, a "Verified" badge, a copy icon, the commit hash "e3a72ad", and a code icon.

# Tips for Effective CI/CD

- Your CI pipeline should run quickly
- Your CI pipeline should run on every change
- Your main branch should always be green
  - Otherwise CI loses its value!

Tools: `extract_features.py`: correct define name for `AP_RPM_ENABLED`


 peterbarker committed 5 days ago ❌

AP\_Mission: prevent use of uninitialised stack data ...

 peterbarker committed 5 days ago ❌

2

AP\_HAL\_ChibiOS: disable DMA on I2C on bdshot boards to free up DMA ch... ...

 andyp1per authored and tridge committed 6 days ago ❌

SITL: Fixed rounding lat/lng issue when running JSBSim SITL ...

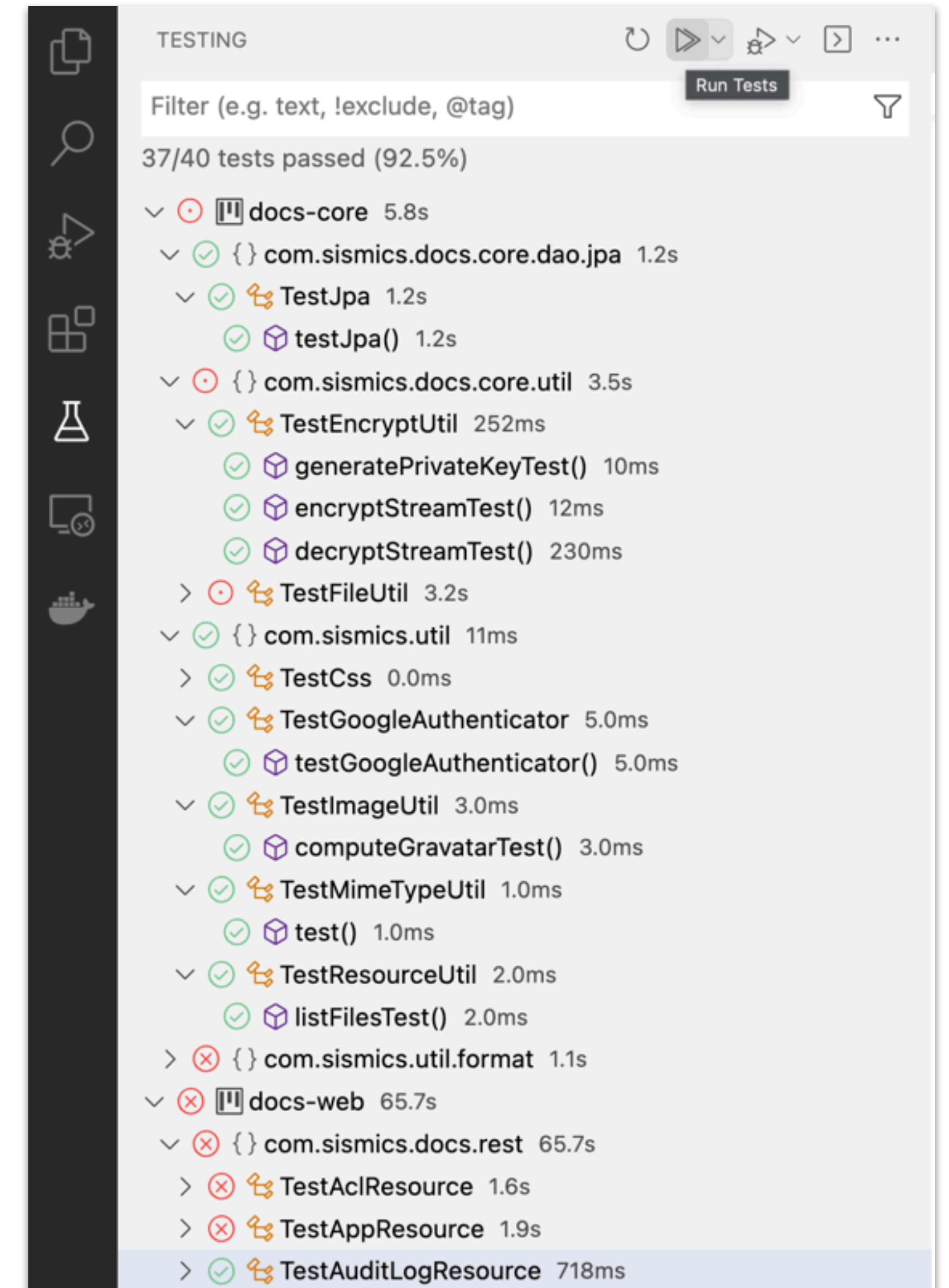
 ShivKhanna authored and tridge committed 6 days ago ❌

AP\_HAL\_ChibiOS: define skyviper short board names

 yuri-rage authored and tridge committed 6 days ago ❌

# Tips for Effective CI/CD

- Your CI pipeline should run quickly
- Your CI pipeline should run on every change
- Your main branch should always be green
- Your CI shouldn't replace your local environment
  - you don't have to re-run every test



# Activity: Let's design a good QA process

- Work in teams of two or three
- Upload to Gradescope

**When it all goes wrong...**





# Diagnose the problem with a checklist

- Have you recompiled the project?
- Have you pushed/pulled all of the changes?
- Can you reproduce the problem on another machine?
- Can you find a previous version of the project that worked?
- Does the query sent by the frontend match the backend's expectations?
- Does the database contain your changes?
- ...

# Collect evidence and report the issue

## Log files, stack traces, terminal outputs, examples, screenshots

youkidearitai commented 5 hours ago

### Description

The following code:

```
<?php
var_dump(mb_stripos("AあいうえおBBcC", "?", 0, "ISO-2022-JP"));
var_dump(mb_stripos("AあいうえおBBcC", "?", 0, "ISO-2022-JP"));
var_dump(mb_stripos("AあいうえおBBcC", "?", 0, "SJIS"));
var_dump(mb_stripos("AあいうえおBBcC", "?", 0, "SJIS"));
```

Resulted in this output:

```
int(1)
int(5)
int(1)
int(5)
```

But I expected this output instead:

```
bool(false)
bool(false)
bool(false)
bool(false)
```

note: mb\_stripos is affect mb\_substitute\_character, but mb\_stripos is not affect it is (3v4l: <https://3v4l.org/872nB> )

### PHP Version

PHP 8.1.x

### Operating System

Ubuntu 20.04 on WSL2

The screenshot shows the IntelliJ IDEA interface. The Log Viewer window displays a list of log messages with columns for ID, MESSAGE, MARKER, LEVEL, and LOGGER. The selected log entry is:

ID	MESSAGE	MARKER	LEVEL	LOGGER
16	SecurityManager is now: null	LogUI	INFO	LogUI

The Stack Trace Analyzer window is open, showing the following stack trace:

```
W: java.lang.Throwable: stack dump
W:   at java.lang.Thread.dumpStack(Thread.java:490)
W:   at com.google.samples.apps.topeka.activity.SignInActivity.onCreate(SignInActivity.java:6237)
W:   at android.app.Activity.performCreate(Activity.java:6237)
W:   at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1107)
W:   at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2369)
W:   at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2476)
W:   at android.app.ActivityThread.-wrap11(ActivityThread.java)
W:   at android.app.ActivityThread$$Method.invoke(Message(ActivityThread.java:1344)
W:   at android.os.Handler.dispatchMessage(Handler.java:102)
W:   at android.os.Looper.loop(Looper.java:148)
W:   at android.app.ActivityThread.main(ActivityThread.java:5417)
W:   at java.lang.reflect.Method.invoke(Native Method)
W:   at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
W:   at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
```

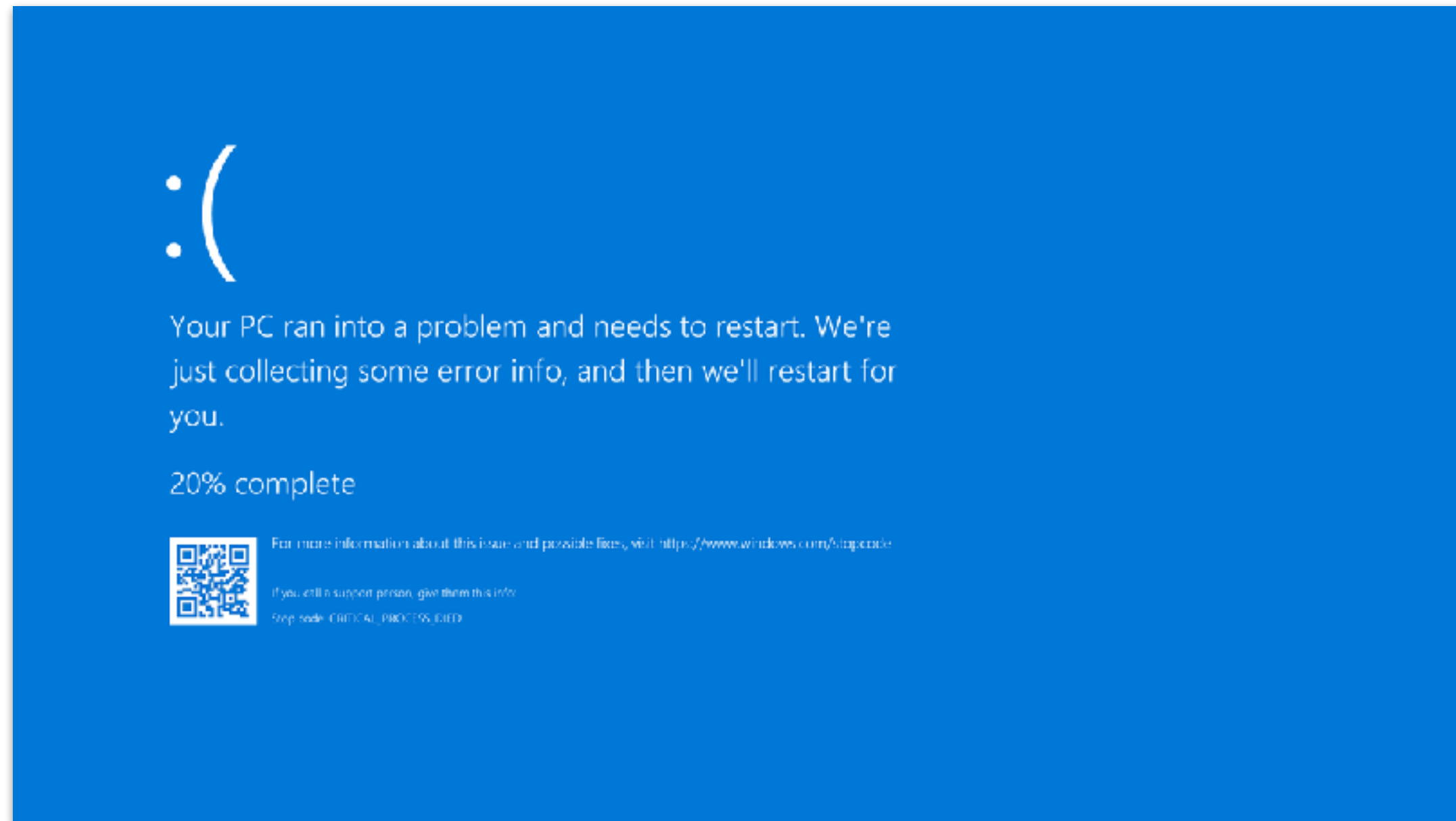
# Write a test to reliably reproduce the failure

## Now your teammates can help out, too!

```
/**
 * Test related to https://github.com/sismics/docs/issues/373.
 */
@Test
public void testIssue373() throws Exception {
    PdfFormatHandler formatHandler = new PdfFormatHandler();
    String content = formatHandler.extractContent(
        language: "deu",
        Paths.get(ClassLoader.getResource("file/issue373.pdf").toURI())
    );
    Assert.assertTrue(content.contains("Aufrechterhaltung"));
    Assert.assertTrue(content.contains("Außentemperatur"));
    Assert.assertTrue(content.contains("Grundumsatzmessungen"));
    Assert.assertTrue(content.contains("ermitteln"));
}
```

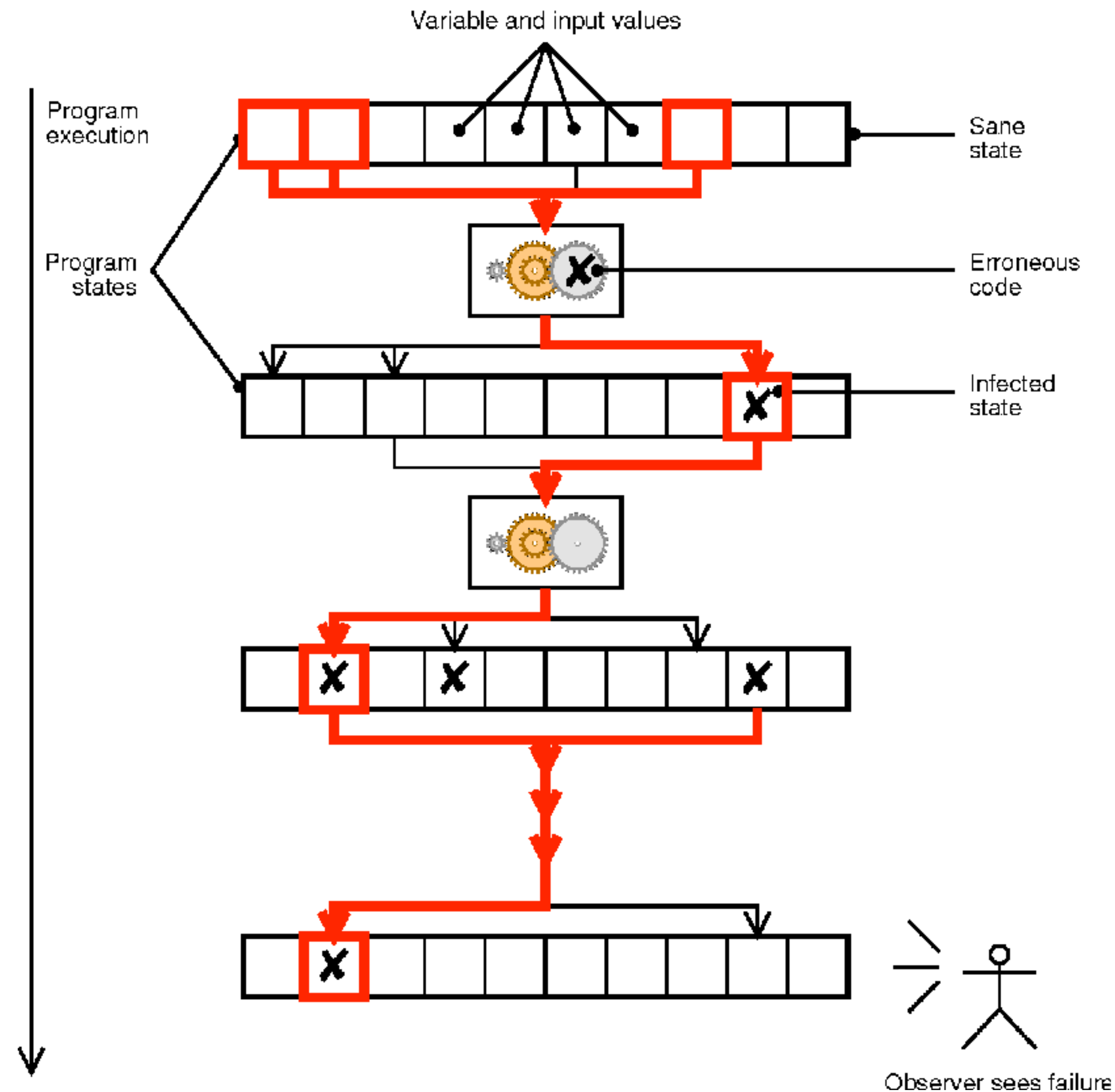
```
Viewed ...
1 + --TEST--
2 + Bug #75917 SplFileObject::seek broken with CSV flags
3 + --FILE--
4 + <?php
5 + $expected = [
6 +     ['john', 'doe', 'john.doe@example.com', '0123456789'],
7 +     ['jane', 'doe', 'jane.doe@example.com'],
8 + ];
9 +
10 + $tmp = new SplTempFileObject();
11 + foreach ($expected as $row) {
12 +     $tmp->fputcsv($row);
13 + }
14 + $tmp->setFlags(0);
15 + $tmp->seek(23);
16 + var_dump($tmp->current());
17 +
18 + $tmp->setFlags(SplFileObject::READ_CSV |
19 +     SplFileObject::SKIP_EMPTY);
20 + $tmp->seek(23);
21 + var_dump($tmp->current());
22 + ?>
23 + --EXPECT--
24 + bool(false)
25 + bool(false)
```

# Problem: What code caused the bug?



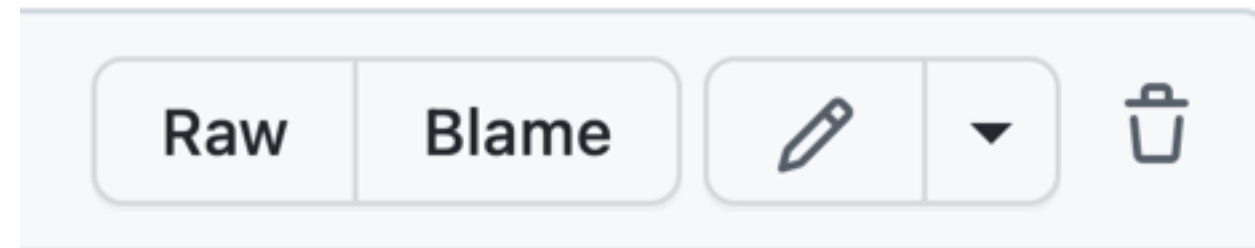


# Defects cause program state to be infected



# Locate the defect: Git History

## Defects tend to be located in recently changed files



Teedy / docs-web / src / main / java / com / sismics / docs / rest / resource /  
BaseResource.java



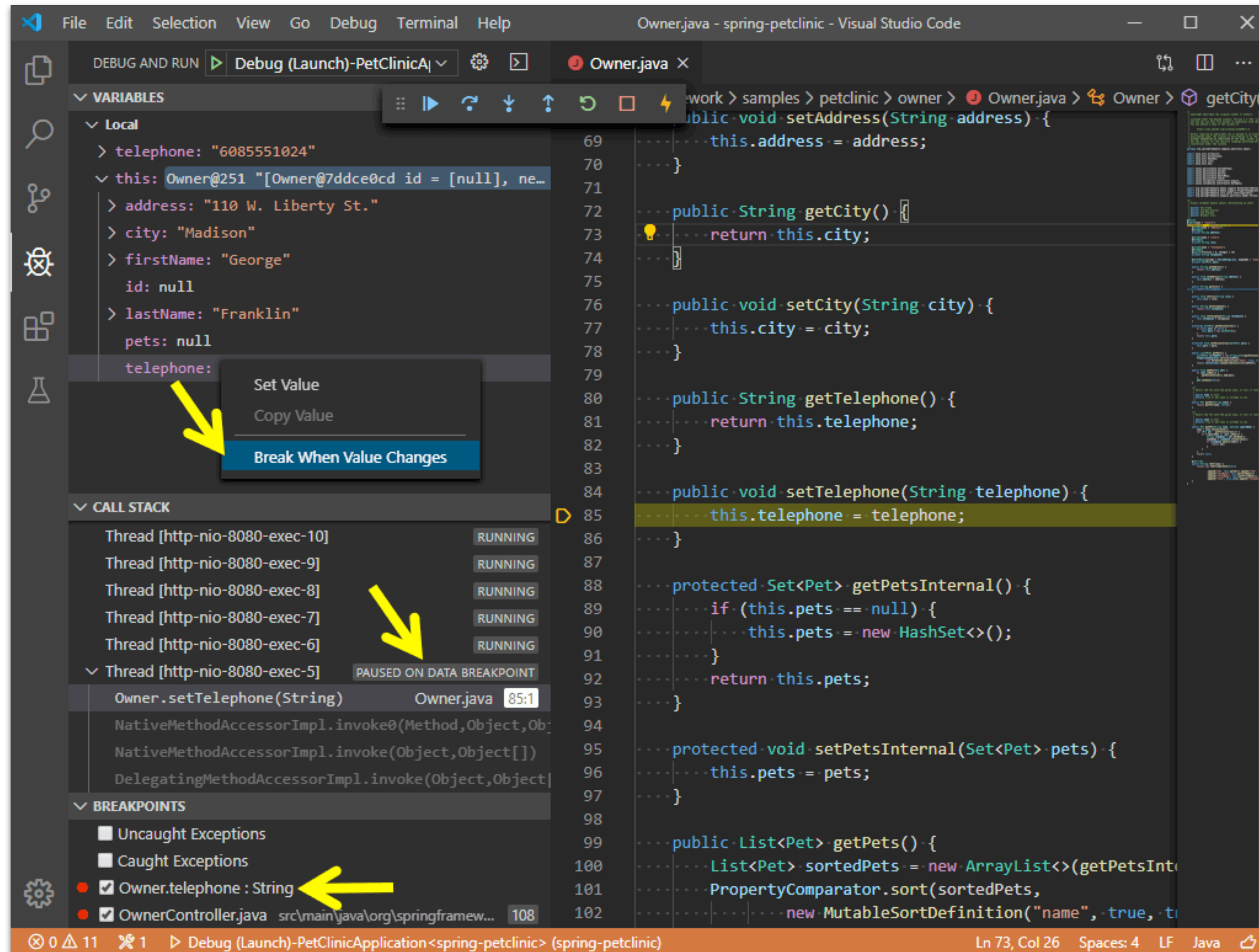
Newer  Older

Commit	Author	Date	Line	Code
100644			124 lines (110 sloc)	3.3 KB
Initial commit		9 years ago	1	<code>package com.sismics.docs.rest.resource;</code>
			2	
#18: ACL check for groups		7 years ago	3	<code>import com.google.common.collect.Lists;</code>
Initial commit		9 years ago	4	<code>import com.sismics.docs.rest.constant.BaseFunction;</code>
			5	<code>import com.sismics.rest.exception.ForbiddenClientException;</code>
			6	<code>import com.sismics.security.IPrincipal;</code>
			7	<code>import com.sismics.security.UserPrincipal;</code>
Closes #106: Header base authentication		6 years ago	8	<code>import com.sismics.util.filter.SecurityFilter;</code>
			9	
			10	<code>import javax.servlet.http.HttpServletRequest;</code>
upgrade to java 11 + upgrade libraries		2 years ago	11	<code>import javax.ws.rs.Consumes;</code>
Download zip of files not in same document (#591)		5 months ago	12	<code>import javax.ws.rs.Produces;</code>
Closes #106: Header base authentication		6 years ago	13	<code>import javax.ws.rs.QueryParam;</code>
			14	<code>import javax.ws.rs.core.Context;</code>
upgrade to java 11 + upgrade libraries		2 years ago	15	<code>import javax.ws.rs.core.MediaType;</code>
Closes #106: Header base authentication		6 years ago	16	<code>import java.security.Principal;</code>
			17	<code>import java.util.List;</code>
			18	<code>import java.util.Set;</code>

main Teedy / docs-web / src / main / java / com / sismics / docs / rest / resource / Go to file Add file ...

Commit	Author	Date	Message
5e7f060	jendib	on May 16	keep filename in temporary file
..			
			AcIResource.java Closes #313: remove administrators from ACL targets search 3 years ago
			AppResource.java Closes #632: validate POST /app/config_inbox and update documentation 5 months ago
			AuditLogResource.java Closes #257: admin users can see all logs 4 years ago
			BaseResource.java Download zip of files not in same document (#591) 5 months ago
			CommentResource.java Refactor documents and files indexing 5 years ago
			DocsMessageBodyWriter.java Download zip of files not in same document (#591) 5 months ago
			DocumentResource.java Add doc for search syntax (#634) 5 months ago
			FileResource.java keep filename in temporary file 4 months ago
			GroupResource.java Allow the . (dot) and @ (at) character in usernames (#637) 5 months ago
			MetadataResource.java #300: custom metadata fields: UI read/write 3 years ago
			RouteModelResource.java Closes #252: route model permissions 4 years ago
			RouteResource.java Fixed sending workflow emails to previous assignee (#422) 2 years ago
			ShareResource.java Closes #509: guest users cannot share and unshare 2 years ago
			TagResource.java Closes #379: spaces and colons not allowed in tag name 3 years ago
			ThemeResource.java sismics docs -> teedy 4 years ago
			UserResource.java rename 5 months ago
			VocabularyResource.java #300: custom metadata fields: API admin 3 years ago
			WebhookResource.java Closes #243: webhooks UI 4 years ago

# Locate the defect: Interactive Debugging



See: <https://github.com/CMU-313/Teedy/wiki/Using-the-Debugger-via-VSCode>



# Locate the defect: Assertions

Reduce the distance between the fault and failure

```
public static int findMax(int v[]) {
    int max = Integer.MIN_VALUE;

    // Precondition: v[] is not empty
    assert v.length > 0 : "v[] is empty";
    // Precondition: max <= v[i] for every i
    for (int i = 0; i < v.length; i++)
        assert max <= v[i] : "Found value < MIN_VALUE";

    // Locate the real maximum value
    for (int i = 0; i < v.length; i++)
        if (v[i] > max)
            max = v[i];

    // Postcondition: max >= v[i] for every i
    for (int i = 0; i < v.length; i++)
        assert max >= v[i] : "Found value > MIN_VALUE";
    return max;
}
```

Detect illegal program states

# Write smaller unit tests as failure hypotheses

## Go from large integration tests to small unit tests

```
/**
 * Exhaustive test of the user resource.
 *
 * @author jtremaux
 */
public class TestUserResource extends BaseJerseyTest {
    /**
     * Test the user resource.
     */
    @Test
    public void testUserResource() {
        // Check anonymous user information
        JsonObject json = target().path(path: "/user").request()
            .acceptLanguage(Locale.US)
            .get(responseType: JsonObject.class);
        Assert.assertTrue(json.getBoolean("is_default_password"));

        // Create alice user
        clientUtil.createUser(username: "alice");

        // Login admin
        String adminToken = clientUtil.login(username: "admin", password: "admin", remember: false);
    }
}
```



```
/**
 * Test of {@link PdfFormatHandler}
 *
 * @author bgamard
 */
public class TestPdfFormatHandler {
    /**
     * Test related to https://github.com/sismics/docs/issues/373.
     */
    @Test
    public void testIssue373() throws Exception {
        PdfFormatHandler formatHandler = new PdfFormatHandler();
        String content = formatHandler.extractContent(language: "deu", Paths.ge
        Assert.assertTrue(content.contains("Aufrechterhaltung"));
        Assert.assertTrue(content.contains("Außentemperatur"));
        Assert.assertTrue(content.contains("Grundumsatzmessungen"));
        Assert.assertTrue(content.contains("ermitteln"));
    }
}
```

# Write a patch and use your tests to validate

## If your tests pass, you've fixed the bug \*

[3.6] bpo-29104: Fixed parsing backslashes in f-strings. (GH-490) #1812

<> Code ▾

Merged serhiy-storchaka merged 1 commit into python:3.6 from serhiy-storchaka:backport-0cd7a3f-3.6 on May 25, 2017

Conversation 0

Commits 1

Checks 0

Files changed 3

+48 -21

Changes from all commits ▾ File filter ▾ Conversations ▾ ⚙

0 / 3 files viewed ⓘ

Review changes ▾

Filter changed files

Lib/test

test\_fstring.py

Misc

NEWS

Python

ast.c

14 Lib/test/test\_fstring.py

Viewed ...

```
@@ -361,6 +361,20 @@ def test_backslashes_in_string_part(self):
```

```
361 self.assertEqual(f'2\x203', '2 3')
362 self.assertEqual(f'\x203', ' 3')
363
```

```
361 self.assertEqual(f'2\x203', '2 3')
362 self.assertEqual(f'\x203', ' 3')
363
```

```
364 + with self.assertWarns(DeprecationWarning): # invalid
    +     escape sequence
365 +     value = eval(r"f'\{6*7}'")
366 +     self.assertEqual(value, '\\42')
367 +     self.assertEqual(f'\\{6*7}', '\\42')
368 +     self.assertEqual(fr'\{6*7}', '\\42')
```

```
369 +
370 +     AMPERSAND = 'spam'
371 +     # Get the right unicode character (&), or pick up
    +     local variable
372 +     # depending on the number of backslashes.
373 +     self.assertEqual(f'\N{AMPERSAND}', '&')
374 +     self.assertEqual(f'\\N{AMPERSAND}', '\\Nspam')
375 +     self.assertEqual(fr'\N{AMPERSAND}', '\\Nspam')
376 +     self.assertEqual(f'\\N{AMPERSAND}', '\\&')
```

```
364 def test_misformed_unicode_character_name(self):
365     # These test are needed because unicode names are
    +     parsed
366     # differently inside f-strings.
```

```
378 def test_misformed_unicode_character_name(self):
379     # These test are needed because unicode names are
    +     parsed
380     # differently inside f-strings.
```

# Takeaways

- Technical debt is a fact of life
- It's easier to preemptively avoid issues and maintain quality than to accumulate technical debt and address it later
- But you can't avoid every issue! We can use techniques to diagnose, debug, and fix those issues that do occur