Roland Weisleder
Softwareentwicklung & Beratung

Legacy-Code mit drei Prinzipien in die Zukunft bringen

Roland Weisleder @Ro_Wei roland@rweisleder.de

About Me

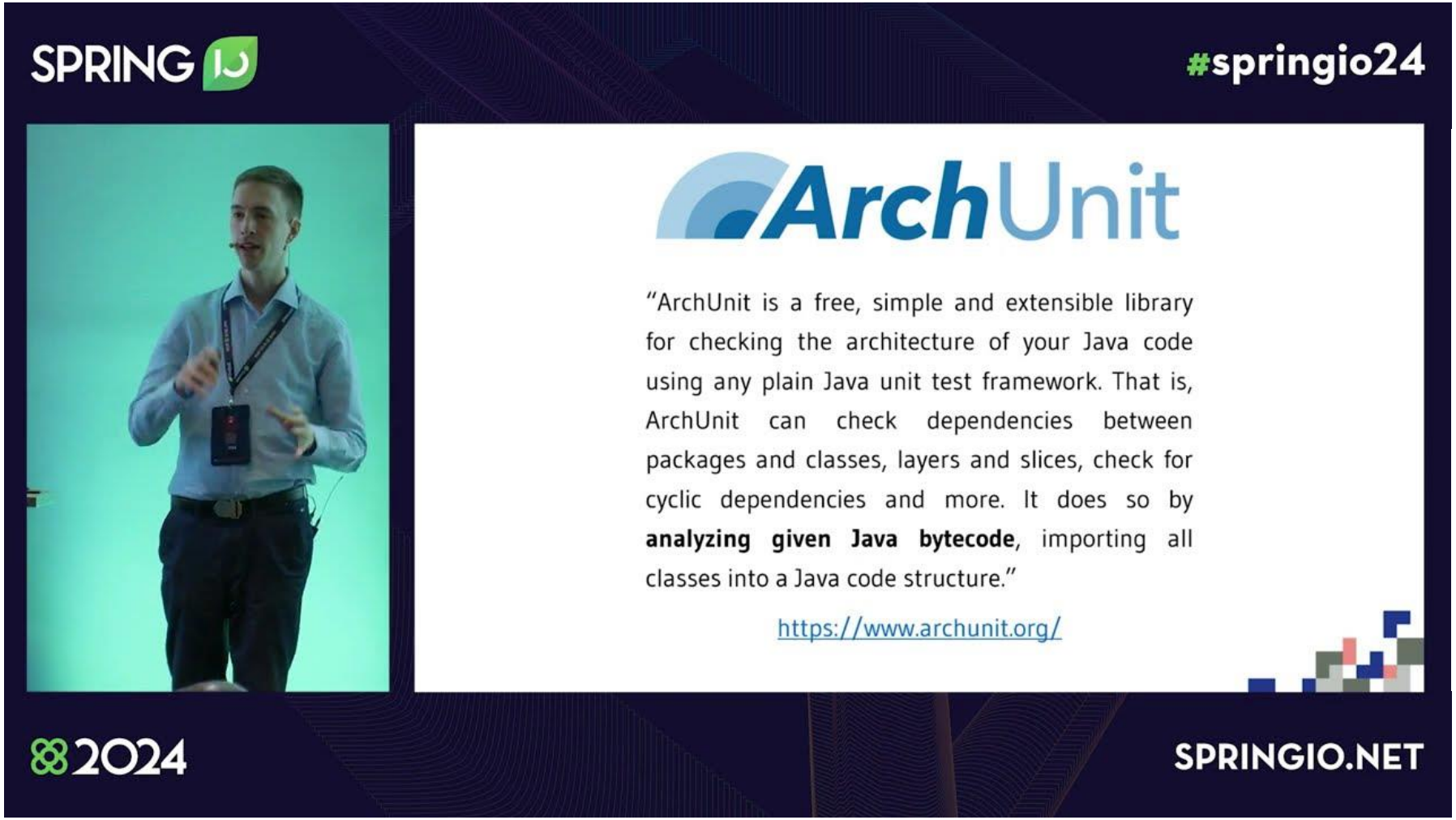
Roland Weisleder
Softwareentwicklung & Beratung

Independent Software Developer & Consultant
Bringing Legacy Systems Into the Future

Spring Boot Trainer @ workshops.de

Lecturer @ HTW Berlin

in/roland-weisleder
@rweisleder
@Ro_Wei
roland@rweisleder.de



Unit Test Your Spring Architecture With ArchUnit

https://www.youtube.com/watch?v=sGmhaizFcEA

What is Legacy Code?*

- Still in production use
- Relies on outdated technologies
- Has insufficient test coverage
- Lacks adequate documentation
- Expert knowledge unavailable

*According to ChatGPT

But First …

Every legacy system is unique.

Therefore, each one requires a unique approach.

Source: https://www.linkedin.com/feed/update/urn:li:activity:7196516808441360384/

Big Bang Rewrite?

Too time-consuming

What to improve?

What you work on anyway

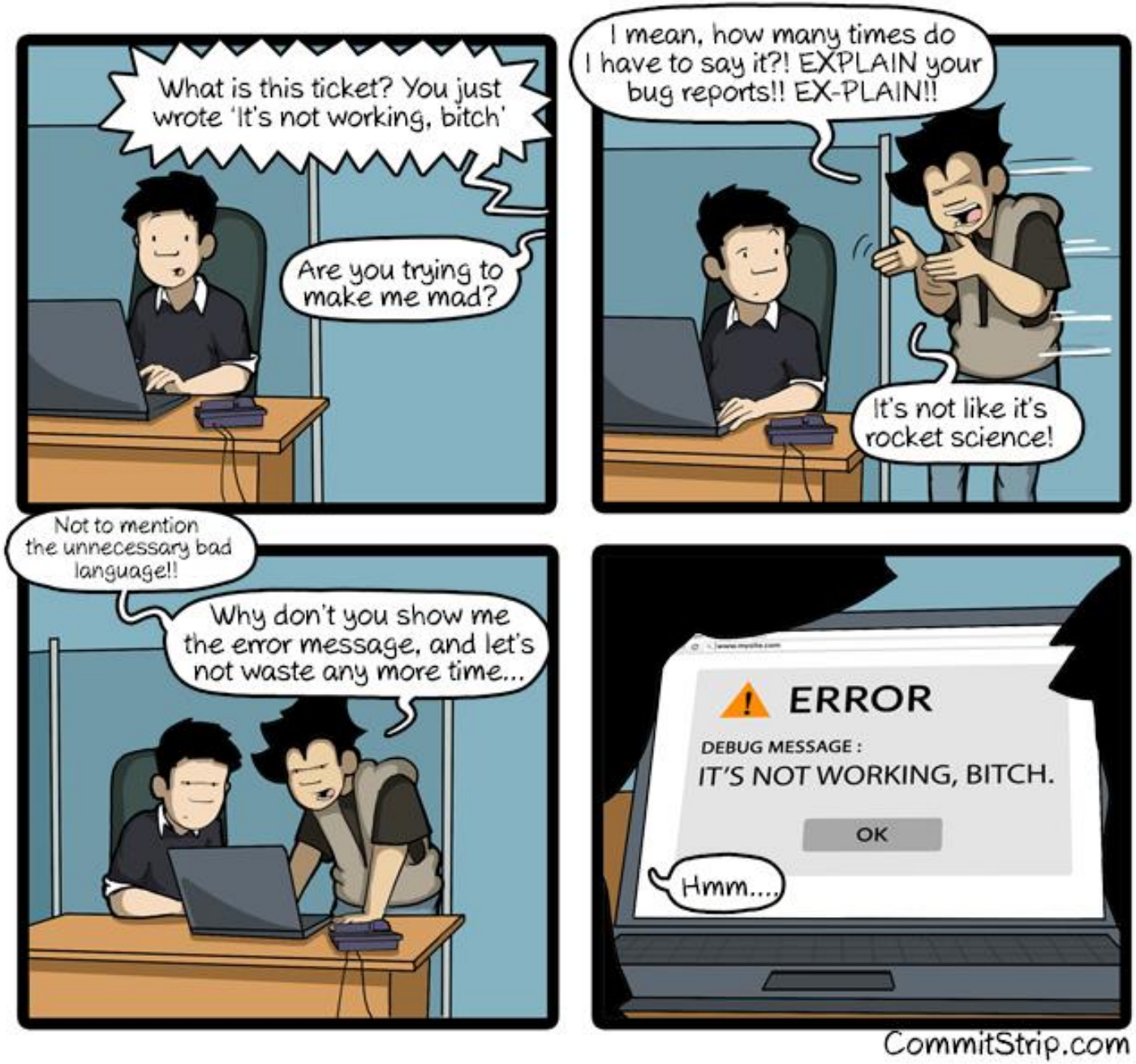
Rewrite or Refactor?

Rewrite only with test coverage

> Old bugs need to be added to the rewritten system.

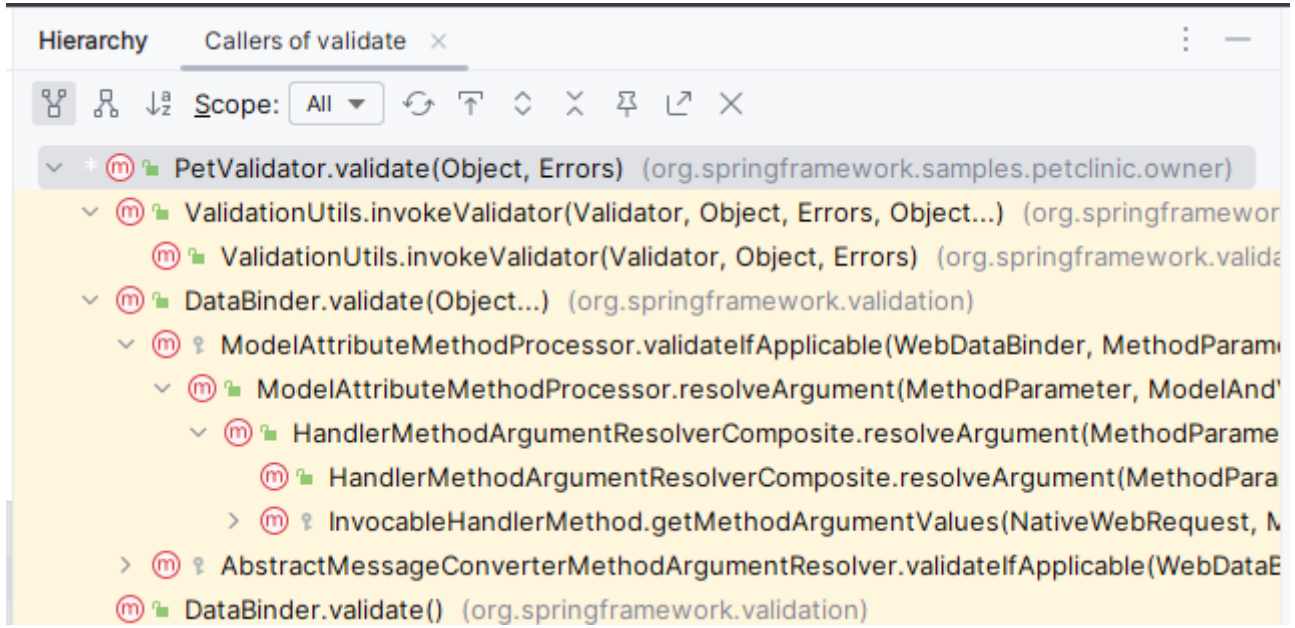
Source: https://martinfowler.com/bliki/StranglerFigApplication.html

Where to start?



Source: https://www.commitstrip.com/en/2016/03/03/its-not-working/

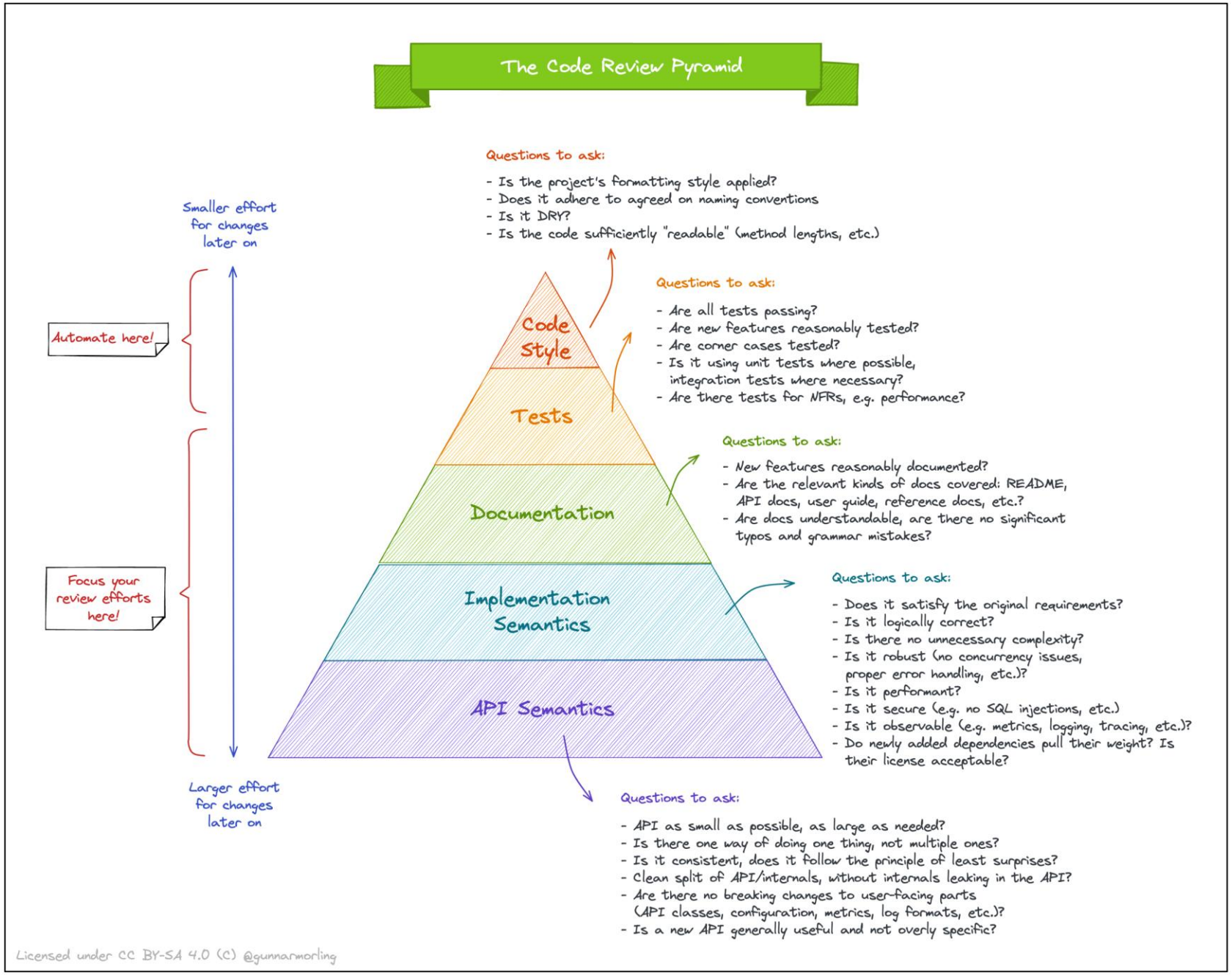
Call Hierarchy



Understand Code

Scratch Refactoring

- Learn through refactoring
- Make experiments
- Revert everything

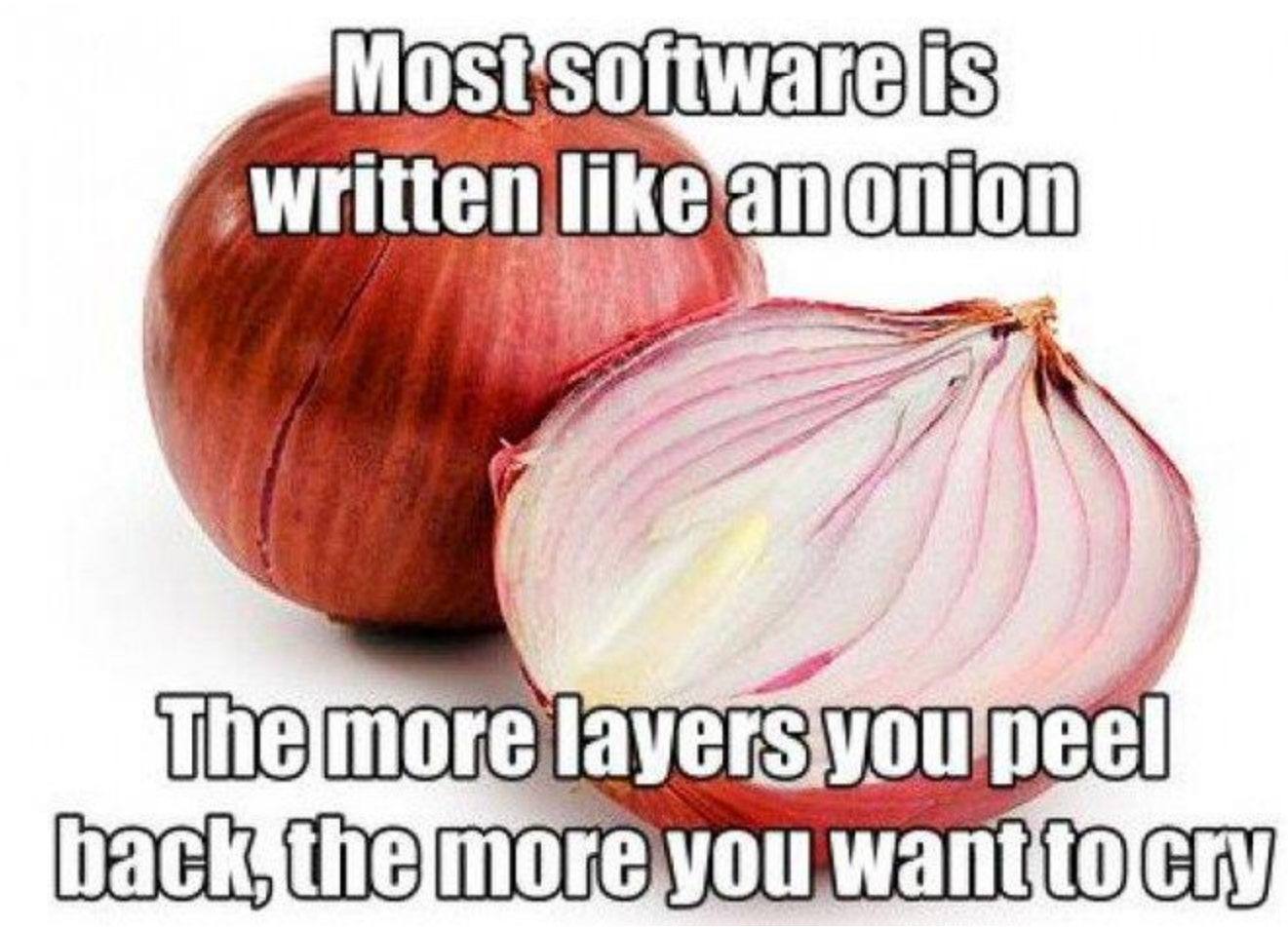


Source: https://www.morling.dev/blog/the-code-review-pyramid/

```java
@RestController
class DemoController {

    @Autowired
    private WebServiceTemplate webServiceTemplate;

    @Autowired
    private DataSource dataSource;

    @GetMapping
    public ResponseEntity<String> fetchAndSaveData() {
        String soapEndpoint = "http://example.com/soapEndpoint";
        String requestPayload = "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:ex=\"http://example.com/\">" +
                "<soapenv:Header/>" +
                "<soapenv:Body>" +
                "<ex:GetDataRequest>" +
                "<ex:parameter1>value1</ex:parameter1>" +
                "<ex:parameter2>value2</ex:parameter2>" +
                "</ex:GetDataRequest>" +
                "</soapenv:Body>" +
                "</soapenv:Envelope>";

        String soapResponse = (String) webServiceTemplate.marshalSendAndReceive(soapEndpoint, new StringSource(requestPayload));

        if (soapResponse == null || soapResponse.isEmpty()) {
            return ResponseEntity.notFound().build();
        }

        String userName = extractUserNameFromResponse(soapResponse);

        if (userName == null || userName.isEmpty()) {
            return ResponseEntity.badRequest().body("Invalid data received");
        }

        String sql = "INSERT INTO users (name) VALUES (?)";
        try (Connection conn = dataSource.getConnection();
             PreparedStatement ps = conn.prepareStatement(sql)) {
            ps.setString( parameterIndex: 1, userName);
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
            return ResponseEntity.status(500).body("Database error");
        }

        return ResponseEntity.ok( body: "Data successfully saved");
    }
```

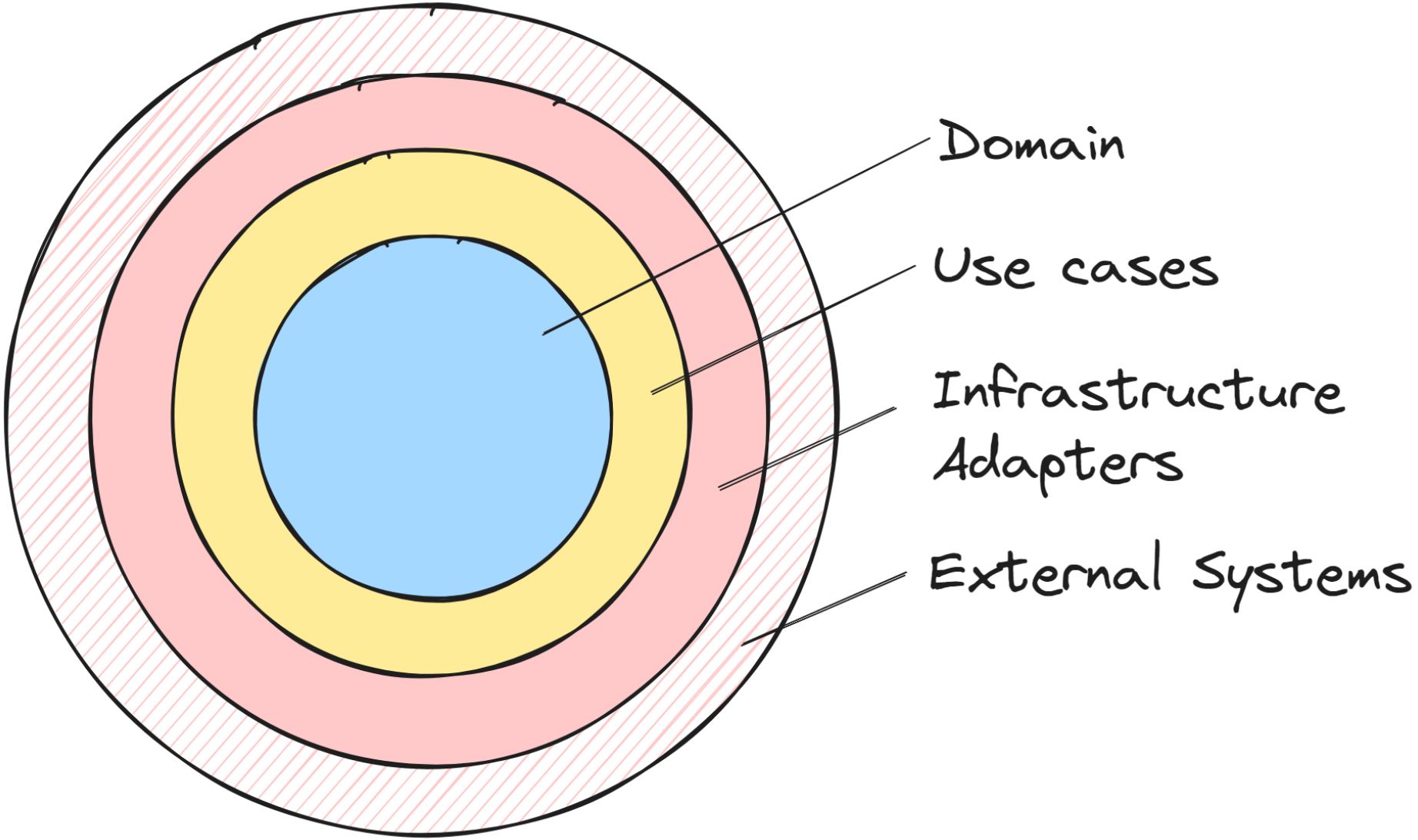
Separate Domain and Infrastructure

to simplify testing
and maintainability



Source: https://twitter.com/dev_humor/status/1226249378253484033

Clean Architecture



Document Code

“

Our code is self-documenting.

```
public static Date parseDate(String input) throws ParseException {
    return DateFormat.getInstance().parse(input);
}
```

```
public static Date parseDate(String input) throws ParseException {
    return DateFormat.getInstance().parse(input);
}
```

- What about the input format?
- What about malformed input?
- What about null?
- What about thrown exceptions?
- What about thread-safety?

```
public static String formatCurrency(double input) {
    return NumberFormat.getCurrencyInstance(Locale.GERMANY).format(input);
}
```

* don’t use double for money…

```java
public static String formatCurrency(double input) {
    return NumberFormat.getCurrencyInstance(Locale.GERMANY).format(input);
}
```

- What about the return value?

* don’t use double for money…

Suddenly a wild Java update appears

```
java.text.ParseException: Unparseable date: "14.10.2023 17:15"

    at java.base/java.text.DateFormat.parse(DateFormat.java:399)


        org.opentest4j.AssertionFailedError:
        Expected :0,00 €
        Actual   :0,00 €
```

“

Unit tests are our documentation.

Who has read the tests of their favorite language / framework / library to understand how to use it?

```java
/**
 * @param input a date with format "dd.MM.yy HH:mm", must not be null
 * @throws ParseException if the given input has a different format
 */
public static Date parseDate(String input) throws ParseException {
    return DateFormat.getInstance().parse(input);
}
```

```
Date date = parseDate( input: "14.10.2023 17:15");
assertNotNull(date)
```

com.gildedrose.ParserAndFormatterTest.ParserAndFormatter

```
public static Date parseDate(
    String input
)
throws ParseException
```

Params: input – a date with format "dd.MM.yy HH:mm", must not be null

Throws: ParseException – if the given input has a different format

gilded-rose-kata

not only

Don‘t **R**epeat **Y**ourself

but also

Documentation **R**eflects **I**mplementation

```java
@RestController
class DemoController {

    @Autowired
    private WebServiceTemplate webServiceTemplate;

    @Autowired
    private DataSource dataSource;

    @GetMapping
    public ResponseEntity<String> fetchAndSaveData() {
        String soapEndpoint = "http://example.com/soapEndpoint";
        String requestPayload = "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:ex=\"http://example.com/\">" +
                "<soapenv:Header/>" +
                "<soapenv:Body>" +
                "<ex:GetDataRequest>" +
                "<ex:parameter1>value1</ex:parameter1>" +
                "<ex:parameter2>value2</ex:parameter2>" +
                "</ex:GetDataRequest>" +
                "</soapenv:Body>" +
                "</soapenv:Envelope>";

        String soapResponse = (String) webServiceTemplate.marshalSendAndReceive(soapEndpoint, new StringSource(requestPayload));

        if (soapResponse == null || soapResponse.isEmpty()) {
            return ResponseEntity.notFound().build();
        }

        String userName = extractUserNameFromResponse(soapResponse);

        if (userName == null || userName.isEmpty()) {
            return ResponseEntity.badRequest().body("Invalid data received");
        }

        String sql = "INSERT INTO users (name) VALUES (?)";
        try (Connection conn = dataSource.getConnection();
             PreparedStatement ps = conn.prepareStatement(sql)) {
            ps.setString( parameterIndex: 1, userName);
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
            return ResponseEntity.status(500).body("Database error");
        }

        return ResponseEntity.ok( body: "Data successfully saved");
    }
```

Oliver B. Fischer
@sweblogtweets

Writing documentation is a kind of a unit test for your software.

If you are able to describe how it works, there is a good chance that it behaves like this.

3:58 PM · Apr 13, 2020

Source: https://twitter.com/sweblogtweets/status/1249698475249467393

Test Code

What should I test?

The documented behavior.


```
/**
 * @param input a date with format "dd.MM.yy HH:mm", must not be null
 * @throws ParseException if the given input has a different format
 */
```

Use Descriptive Test Names

```
@Test
void quality_of_Sulfuras_does_not_change() {...}

@Test
void sellIn_of_Sulfuras_does_not_change() {...}
```

Concise Implementation

Keep relevant stuff together,
hide irrelevant stuff

```
@Test
void quality_of_Sulfuras_does_not_change() {
    // Arrange
    Item sulfuras = sulfuras();
    int originalQuality = sulfuras.quality;

    // Act
    keepItemForFiveDays(sulfuras);

    // Assert
    assertEquals(originalQuality, sulfuras.quality);
}
```

I want to test private methods

Private methods are independent units in disguise.

Extract, document and test.

The code contains hard-to-mock dependencies

Current Time

Database

External Systems

Declare dependencies as parameters

```
public List<String> loadAllUsernames() throws Exception {
    DataSource dataSource = (DataSource) new InitialContext()
        .lookup( name: "jndi:/datasource");
    try (Connection connection = dataSource.getConnection()) {
        return ...;
    }
}
```

vs

```
private final DataSource dataSource;

public UserService(DataSource dataSource) {
    this.dataSource = dataSource;
}
```

Code Coverage

- Find untested code
- Find dead code

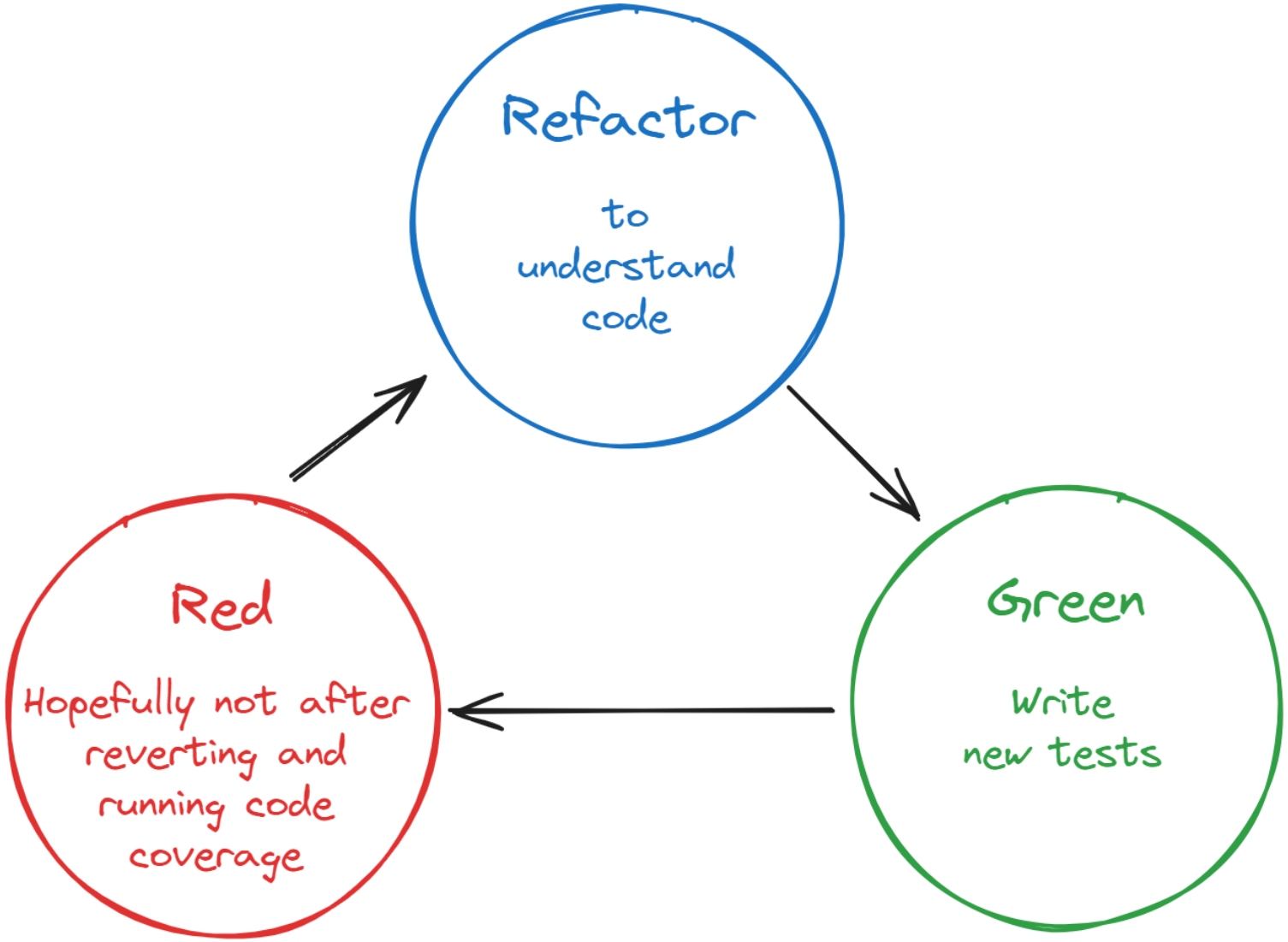
```
10      public void updateQuality() {
11          for (int i = 0; i < items.length; i++) {
12              if (!items[i].name.equals("Aged Brie") && !items[i].
13                  if (items[i].quality > 0) {
14                      if (!items[i].name.equals("Sulfuras, Hand of
15                          items[i].quality = items[i].quality - 1;
16                      }
17                  }
18              } else {
19                  if (items[i].quality < 50) {
20                      items[i].quality = items[i].quality + 1;
21
22                      if (items[i].name.equals("Backstage passes t
23                          if (items[i].sellIn < 11) {
```

Still doing Scratch Refactoring?

Revert everything, except the tests.

Congratulations! Time for TDD.

Inverted TDD



Kent Beck 🌻

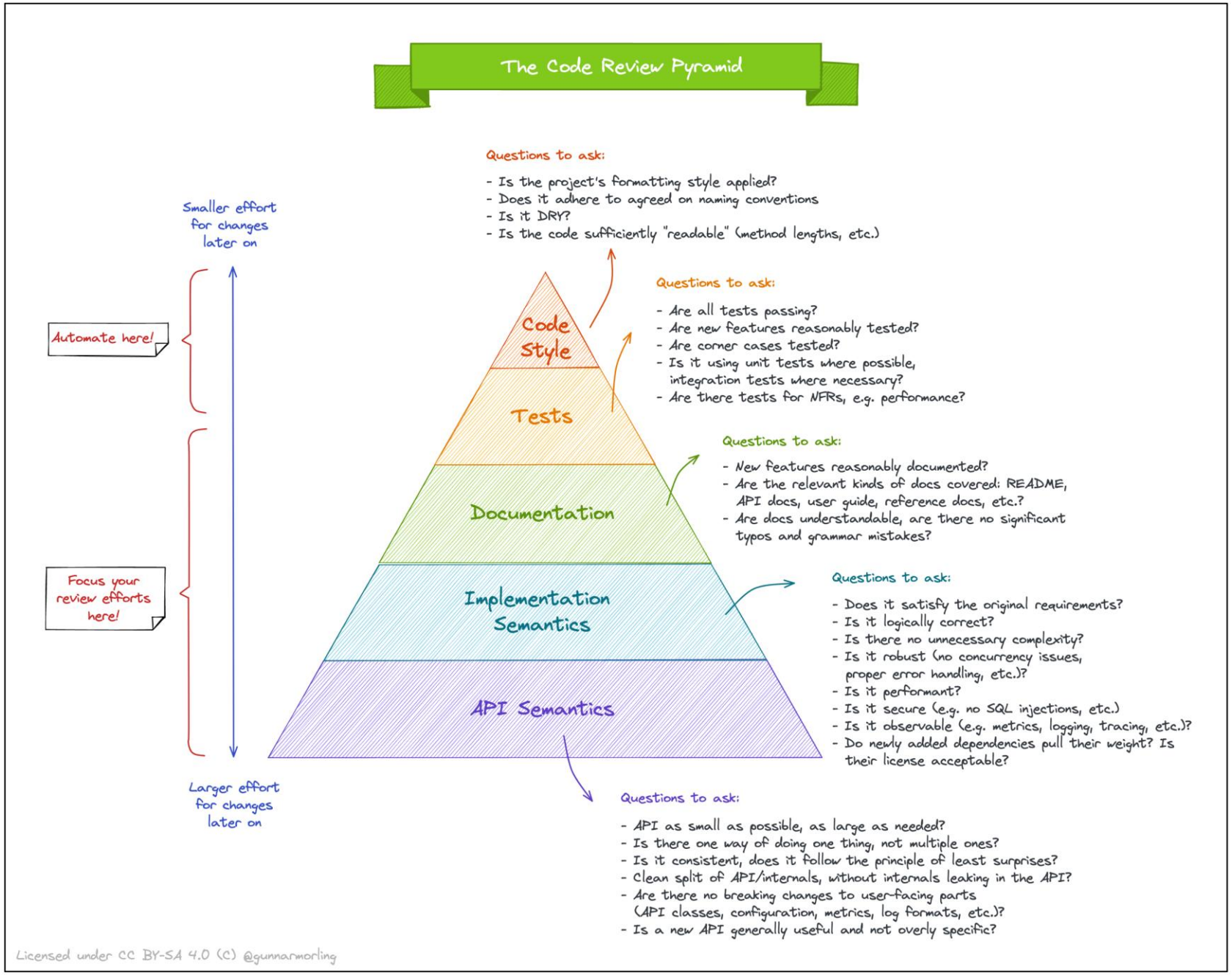
@KentBeck

for each desired change, make the change easy (warning: this may be hard), then make the easy change

1:07 AM · Sep 26, 2012

Source: https://twitter.com/kentbeck/status/250733358307500032

Improve Code



Source: https://www.morling.dev/blog/the-code-review-pyramid/

Define Clean Interfaces

Input

Process

Output

```php
function processUserAndPosts($userId) {
    $sessionValue = $_SESSION['session_value'] ?? '';
    $postValue = $_POST['post_value'] ?? '';

    $userData = DB::select(DB::raw("
        SELECT *
        FROM users
        WHERE id = '$userId'
        AND session_value = '$sessionValue'
    "));

    if (empty($userData)) {
        return false;
    }

    $postsData = [];
    foreach ($userData as $key => $value) {
        if (strpos($key, 'post_') === 0) {
            $postId = str_replace('post_', '', $key);
            $postData = DB::select(DB::raw("
                SELECT *
                FROM posts
                WHERE id = '$postId'
                AND post_value = '$postValue'
            "));
            $postsData[] = $postData;
        }
    }

    return [$userData, $postsData];
}
```

Source: https://www.linkedin.com/feed/update/urn:li:activity:7193713431269584896/

Invalid objects ...

```
public class Person {

    private String firstName;

    private String lastName;

    // getter, setter ...

    public void sayYourName() {
        System.out.println("Hi, my name is " + firstName + " " + lastName);
    }

    public static void main(String[] args) {
        Person person = new Person();
        person.sayYourName();
    }
}
```

… vs valid objects

```java
public class Person {

    private final String firstName;

    private final String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    // getter ...

    public void sayYourName() {
        System.out.println("Hi, my name is " + firstName + " " + lastName);
    }

    public static void main(String[] args) {
        Person person = new Person("Roland", "Weisleder");
        person.sayYourName();
    }
}
```

What about null?

```
public Person(@NotNull String firstName, @NotNull String lastName) {
    this.firstName = firstName;
    this.lastName = lastName;
}

public static void main(String[] args) {
    Person person = new Person(null, null);
    person.sayYourName();
}
```

Passing 'null' argument to parameter annotated as @NotNull

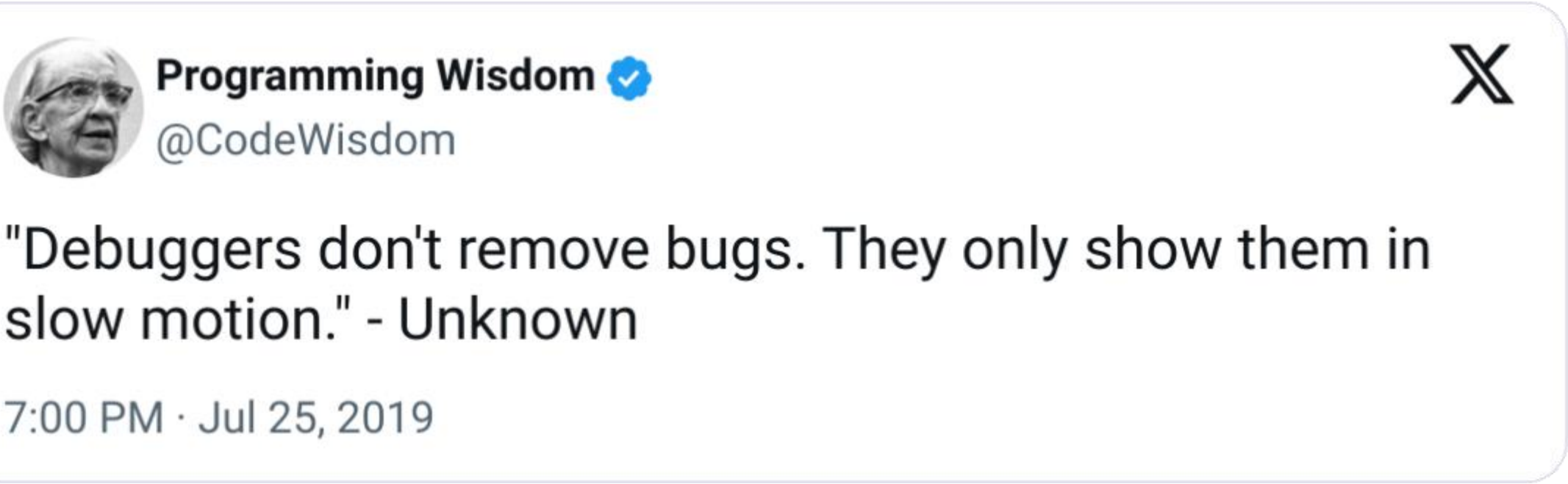
© com.gildedrose.Person

Primitive Obsession

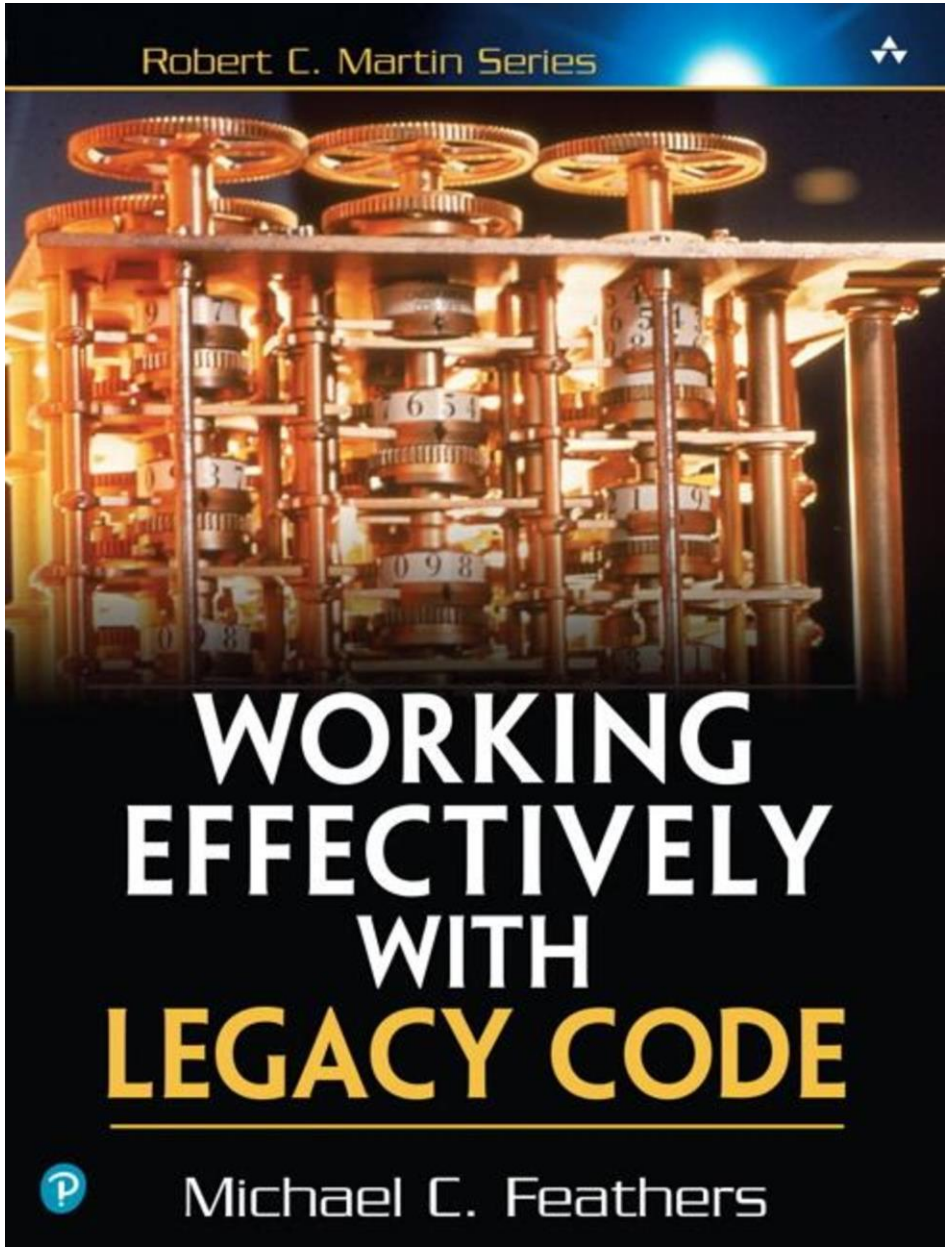
```
/**
 * @throws IllegalArgumentException
 *         if the given mail address has an invalid syntax
 */
public void sendWelcomeMail(String mailAddress) {
    // ...
}
```

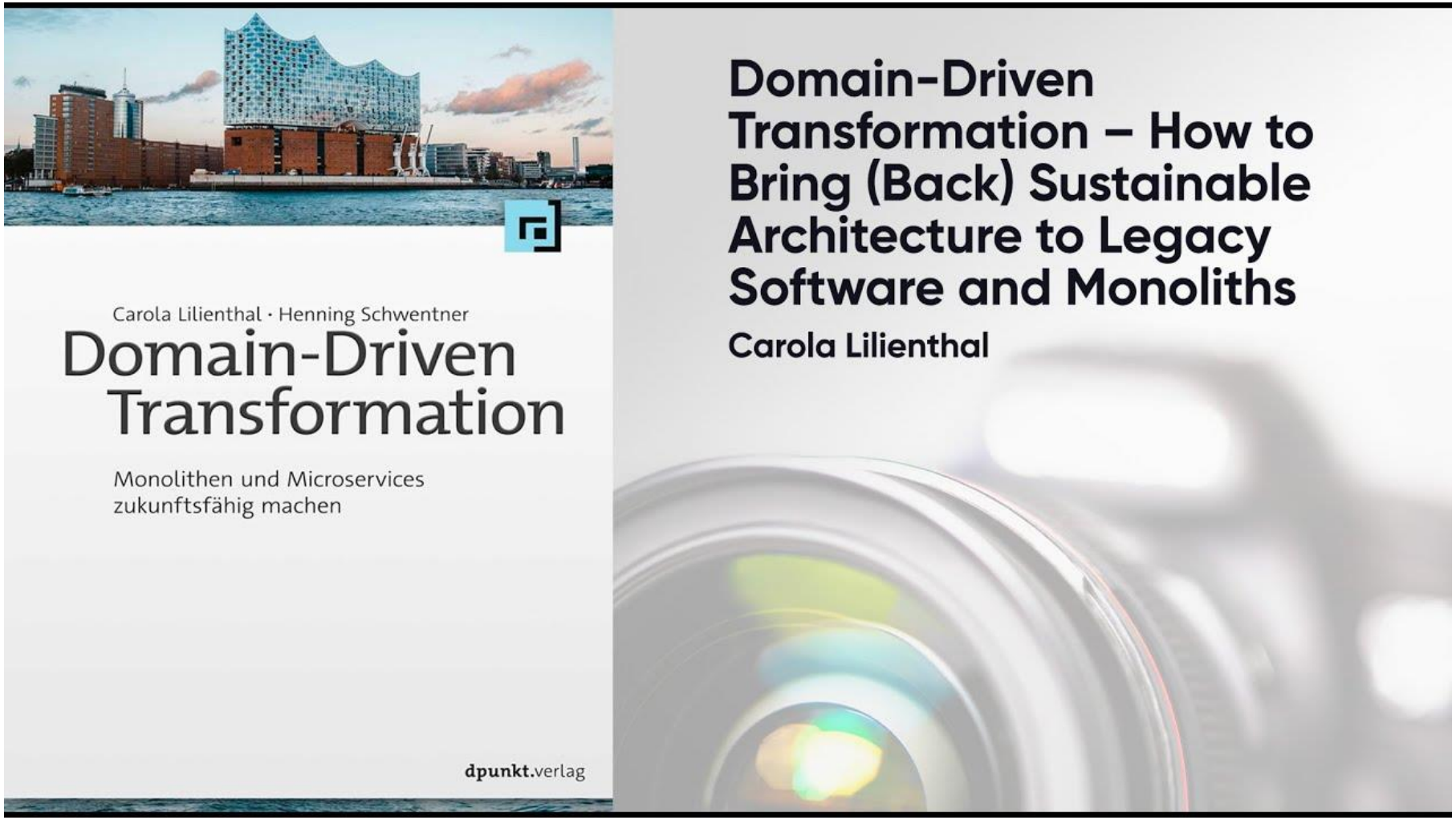
vs

```
public void sendWelcomeMail(MailAddress mailAddress) {
    // ...
}
```



Source: https://twitter.com/codewisdom/status/1154436195126009858


Robert C. Martin Series
WORKING EFFECTIVELY WITH LEGACY CODE
Michael C. Feathers



https://www.youtube.com/watch?v=OGZ1JkC-cQc

Key Takeaways

Separate Domain and Infrastructure

Documentation Reflects Implementation

Input Process Output

Bringing Legacy Code Into the Future

Roland Weisleder
Softwareentwicklung & Beratung

Slides:

speakerdeck.com/rweisleder

Gilded Rose Kata:

github.com/emilybache/GildedRose-Refactoring-Kata

Need help with your Legacy Code? Contact me!

roland@rweisleder.de
in/roland-weisleder
@rweisleder
@Ro_Wei

Feedback 🍅 👏
Questions ⁉️
Connect 🔗