Test-Driven Development (TDD)

TDD: Description

- Developer writes an automated test case prior to writing the product code implementing the feature under test
- Most frequently used as an enhancement to unit-level testing
- May also be used with acceptance testing

TDD Process

Green

Red

Refactor

https://martinfowler.com/articles/workflowsOfRefactoring/#tdd

TDD Process

Add a test for yet-to-be-build functionality. This test will fail (test suite becomes red).

Green

Red

Refactor

https://martinfowler.com/articles/workflowsOfRefactoring/#tdd

TDD Process

Add a test for yet-to-be-build functionality. This test will fail (test suite becomes red).

Make the failing test pass (go green) by implementing the necessary functionality simply but crudely.

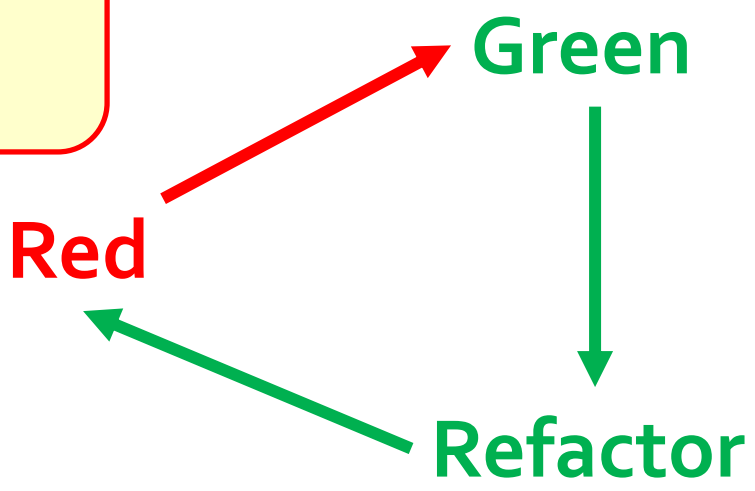
Red → **Green** → **Refactor**

https://martinfowler.com/articles/workflowsOfRefactoring/#tdd

TDD Process

Make the failing test pass (go green) by implementing the necessary functionality simply but crudely.

Add a test for yet-to-be-build functionality. This test will fail (test suite becomes red).



Use refactoring to ensure the overall code base is as clean and well-designed as possible for currently-implemented functionality.

https://martinfowler.com/articles/workflowsOfRefactoring/#tdd

TDD: Measuring Defect Removal Effectiveness

- Direct Measurement: No known method (because TDD may prevent as well as remove defects)

- Indirect Measurement:
  - Measure the reduction in down-stream defect density in a large code population before/after introducing TDD
  - Published Results: 20..91% (Williams, Brown, et al.)

TDD: Why it Works (Potential Explanation)

- Thinking about test cases before writing the code helps incrementally design the software
  - Writing test cases before code requires writing stubs/mocks that interact with other objects through interfaces designed incrementally

- TDD encourages more thorough testing (which remove more defects). Evidence: TDD seems to increase:
  - LinesOfTestCode / LinesOfProductCode ratio and
  - test coverage

TDD: Advantages

- May reduce time to find a defect because problem is known to lie within just a few lines of new code

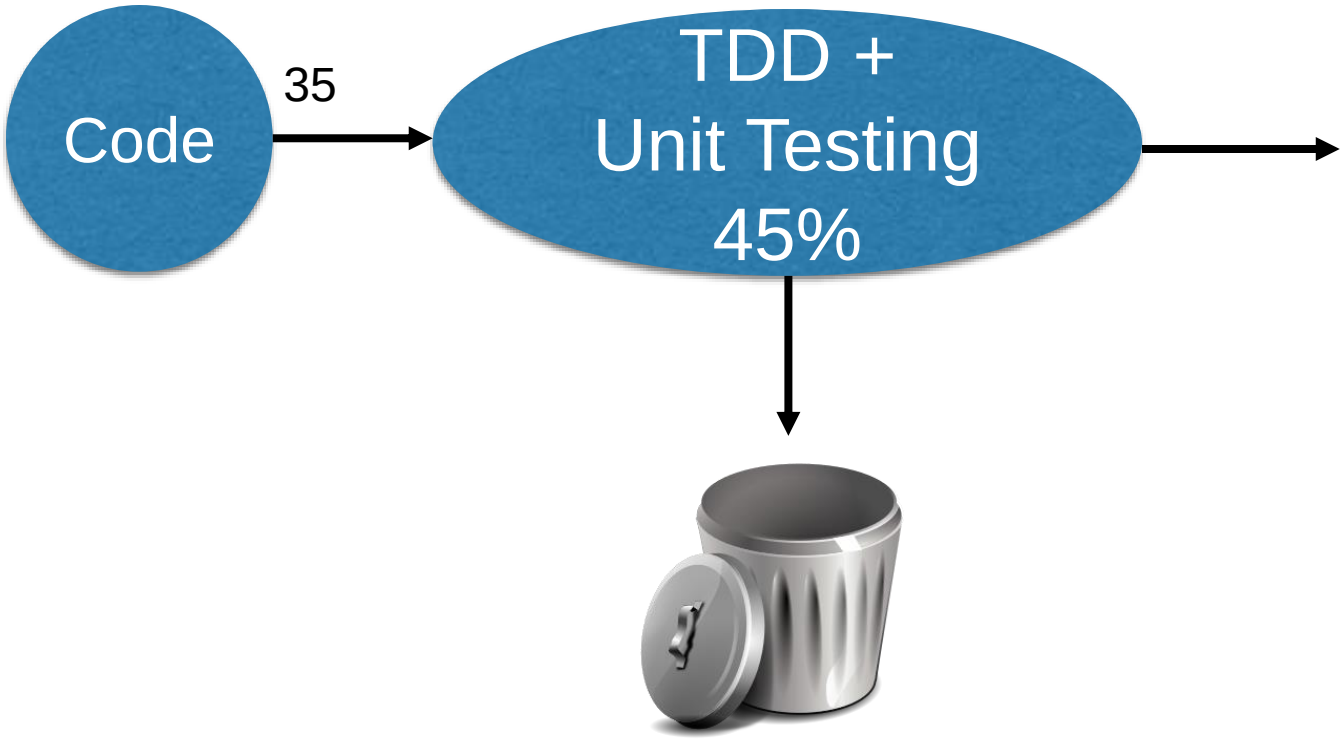
TDD: Disadvantages

- Test and product code are written by the same developer (or pair), effectively limiting their thoroughness to their skill
  - (similar to a Code Review)

- Increase code churn:
  - Throw away (or refactor) tests when code is refactored

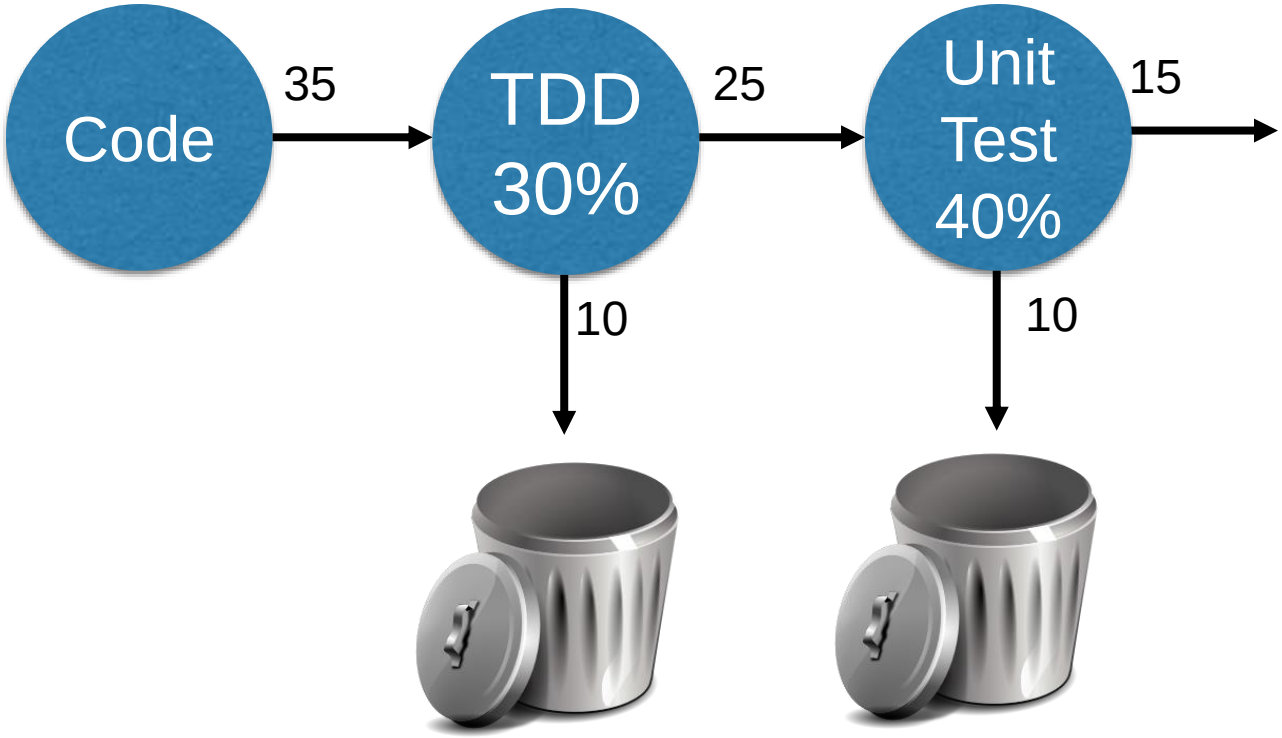
TDD: Modeling

- Version 1: Model TDD as a more effective unit-level testing activity



TDD: Modeling

- Version 2: Model TDD as a distinct defect removal activity with an effectiveness of about 30% preceding a traditional unit-level test



Static Analysis

"Smart people make dumb mistakes"

Static Analysis: Description

- Automated review of source code — essentially free (i.e., cheap to run)
- "Static" because code is not executed
- Static Analysis Tools available for most important languages
- Many tools have user-editable rules that configure what errors they will flag
  - e.g., "I'm only interested in null pointer exceptions"

Static Analysis: Example Tools

- **Java:** FindBugs, Checkstyle, ThreadSafe, Coverity
- **JavaScript:** Google's Closure, JSLint, JSHint
- **Objective-C**: Clang (in Xcode)
- **C#:** FxCop, Coverity
- **C/C++:** Coverity, PRQA QA C, QA C++, SLAM Project
- **PHP:** RIPS
- **Python:** Pylint

- Special Purpose: Fortify

Sample Code for Review

```
// try to get the service
Object obj = ServiceManager.getService(FileOpenerService.class, myFinder);

// Preferred service is not found, use the only one available instead
if ((obj == null) && (!myFinder.equals(finders[0])))
    obj = ServiceManager.getService(FileOpenerService.class, finders[0]);
// Open the file!
((FileOpenerService)obj).openFile(fileName, view);
```

Sample Code for Review

```
// try to get the service
Object obj = ServiceManager.getService(FileOpenerService.class, myFinder);

// Preferred service is not found, use the only one available instead
if ((obj == null) && (!myFinder.equals(finders[0])))
    obj = ServiceManager.getService(FileOpenerService.class, finders[0]);
// Open the file!
((FileOpenerService)obj).openFile(fileName, view);
```

Possible null pointer dereference of obj

Class:
FileOpenerService (org.jedit.core) line 79

Method:
open (org.jedit.core.FileOpenerService.open(String, View))

Priority:
Medium Confidence Correctness

Problem classification:
Correctness (Null pointer dereference)
NP_NULL_ON_SOME_PATH (Possible null pointer dereference)

Notes:
Dereferenced at FileOpenerService.java:[line 79]
Value loaded from obj
Known null at FileOpenerService.java:[line 76]
FindNullDeref (NP|RCN)

Bug Details

Possible null pointer dereference

There is a branch of statement that, if executed, guarantees that a null value will be dereferenced, which would generate a `NullPointerException` when the code is executed. Of course, the problem might be that the branch or statement is infeasible and that the null pointer exception can't ever be executed; deciding that is beyond the ability of FindBugs.

Example Error found by FindBugs

- "Possible null pointer dereference of obj"

```
71
72          // try to get the service
73          Object obj = ServiceManager.getService(FileOpenerService.class, myFinder);
74
75          // Preferred service is not found, use the only one available instead
76          if ((obj == null) && (!myFinder.equals(finders[0])))
77              obj = ServiceManager.getService(FileOpenerService.class, finders[0]);
78          // Open the file!
79          ((FileOpenerService)obj).openFile(fileName, view);
80      }
81  }
82
```

FindBugs-IDEA: jEdit (found 670 bug items in 1357 classes) more...
- Of Concern (579 items)
- Troubling (78 items)
- Scary (13 items)
  - Null pointer dereference (9 items)
    - Non-null field is not initialized (2 items)
      - Non-null field foldHandler is not initialized by new org.gjt.sp.jedit
      - Non-null field foldHandler is not initialized by new org.gjt.sp.jedit
    - Possible null pointer dereference (5 items)
      - Possible null pointer dereference of attributes in new org.gjt.sp.je
      - Possible null pointer dereference of mediaSize in org.gjt.sp.jedit.p
      - Possible null pointer dereference of mediaSize in org.gjt.sp.jedit.p
      - Possible null pointer dereference of mediaSize in org.gjt.sp.jedit.p
      - Possible null pointer dereference of obj in org.jedit.core.FileOpen
  - Method may return null, but is declared @Nonnull (2 items)
  - equals() method does not check for null argument (0 items)
- Static use of type Calendar or DateFormat (2 items)
- Unsynchronized Lazy Initialization (1 item)
- Dubious method invocation (1 item)

Preview FileOpenerService.java:

```
75          // Preferred service is not found, use the only
76          if ((obj == null) && (!myFinder.equals(finders[0
77              obj = ServiceManager.getService(FileOpenerSe
78          // Open the file!
79          ((FileOpenerService)obj).openFile(fileName, view
80      }
81  }
82
```

Possible null pointer dereference of obj

Class: FileOpenerService (org.jedit.core) line 79

Method: open (org.jedit.core.FileOpenerService.open(String, View))

Priority: Medium Confidence Correctness

Problem classification: Correctness (Null pointer dereference) NP_NULL_ON_SOME_PATH (Possible null pointer dereference)

Notes: Dereferenced at FileOpenerService.java:[line 79]; Value loaded from obj; Known null at FileOpenerService.java:[line 76]; FindNullDeref (NP|RCN)

Possible null pointer dereference

There is a branch of statement that, if executed, guarantees that a null value will be dereferenced, which would generate a NullPointerException when the code is executed. Of course, the problem might be that the branch or statement is infeasible and that the null pointer exception can't ever be executed; deciding that is beyond the ability of FindBugs.

Compilation completed successfully in 1m 43s 591ms (9 minutes ago)

http://findbugs.sourceforge.net/ (IntelliJ IDEA Plugin)

Example Error found by FindBugs

- "Incorrect lazy initialization and update of static filed Chunk.fontSubstList"

```
//}}}

//{{{ getFontSubstList() method
private static Font[] getFontSubstList()
{
    if (fontSubstList == null)
    {
        if (fontSubstSystemFontsEnabled)
        {
            Font[] systemFonts = GraphicsEnvironment.getLocalGraphicsEnvironment().getAllFonts();

            fontSubstList = new Font[preferredFonts.length +
                                     systemFonts.length];
```

Incorrect lazy initialization and update of static field Chunk.fontSubstList

Class: Chunk (org.gjt.sp.jedit.syntax) lines 519-537

Method: getFontSubstList (org.gjt.sp.jedit.syntax.Chunk.getFontSubstList())

Field: fontSubstList

Problem classification: Multithreaded correctness (Unsynchronized Lazy Initialization) LI_LAZY_INIT_UPDATE_STATIC (Incorrect lazy initialization and update of static field)

Notes: On field org.gjt.sp.jedit.syntax.Chunk.fontSubstList LazyInit (LI)

Incorrect lazy initialization and update of static field

This method contains an unsynchronized lazy initialization of a static field. After the field is set, the object stored into that location is further updated or accessed. The setting of the field is visible to other threads as soon as it is set. If the futher accesses in the method that set the field serve to initialize the object, then you have a very serious multithreading bug, unless something else prevents any other thread from accessing the stored object until it is fully initialized.

http://findbugs.sourceforge.net/ (IntelliJ IDEA Plugin)

Sonarcloud example

README.md

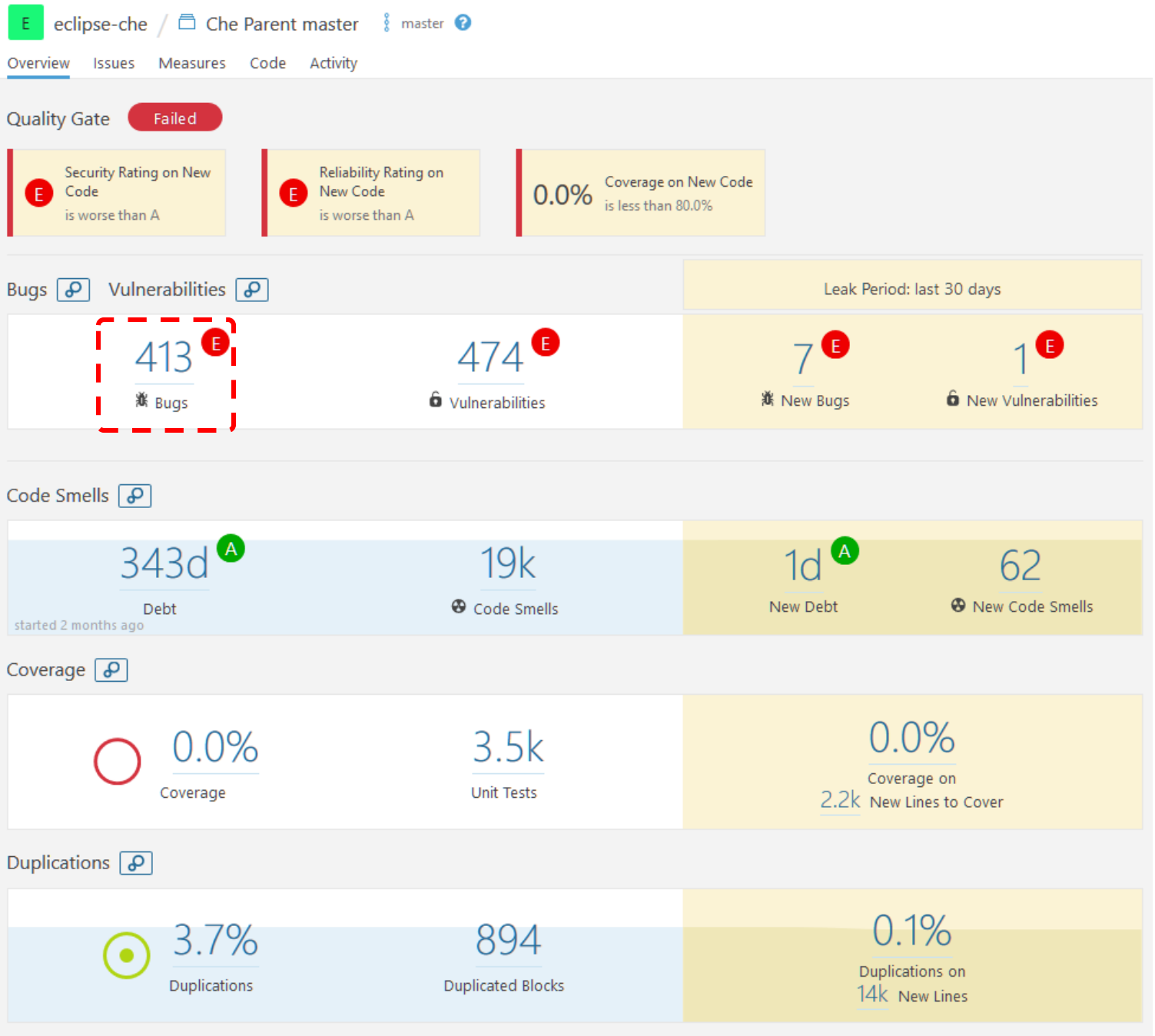
Eclipse Che - Eclipse Next-Generation IDE

license Eclipse | build passing | scanned on sonarcloud

https://www.eclipse.org/che/. Next-generation Eclipse platform, developer workspace serve
workspaces that include their dependencies including embedded containerized runtimes, W
makes workspaces distributed, collaborative, and portable to run anywhere on a desktop o

https://github.com/eclipse/che

Sonarcloud example



https://sonarcloud.io/dashboard?id=org.eclipse.che%3Ache-parent%3Amaster

Sonarcloud example

sonarcloud

Explore | Search for projects, sub-projects and files... | Log in

E eclipse-che / Che Parent master | master

April 16, 2018, 10:35 AM Version 6.4.0-SNAPSHOT

Overview | Issues | Measures | Code | Activity

Filters | Clear All Filters

Display Mode: Issues | Effort

Type | Clear
- Bug 413
- Vulnerability 474
- Code Smell 19k

Severity
- Blocker 55
- Critical 6
- Major 279
- Minor 73
- Info 0

Resolution
Status
Creation Date
Rule
Tag

↑ ↓ to select issues | ← → to navigate | 1 / 413 issues

Che Plugin :: Java :: Java Deb... / pom.xml

Update this scope and remove the "systemPath". — 6 months ago — L293 — 1
Bug | Critical | Open | Not assigned | 5min effort — lock-in, maven

Che Core :: API :: Core master / src/.../org/eclipse/che/api/core/util/DefaultProcessManager.java

Either re-interrupt this method or rethrow the "InterruptedException". — 2 years ago — L30
Bug | Major | Open | Not assigned | 15min effort — cwe, multi-threading

Che Core :: API :: Core master / src/.../org/eclipse/che/api/core/util/FileCleaner.java

Either re-interrupt this method or rethrow the "InterruptedException". — 2 years ago — L76
Bug | Major | Open | Not assigned | 15min effort — cwe, multi-threading

Che Core :: API :: Core master / src/.../org/eclipse/che/api/core/util/RateExceedDetector.java

Cast one of the operands of this integer division to a "double". — 2 years ago — L73
Bug | Minor | Open | Not assigned | 5min effort — cert, cwe, misra, overflow, sans-top25-ri...

Cast one of the operands of this division operation to a "double". — 2 years ago — L73
Bug | Minor | Open | Not assigned | 5min effort — cert, cwe, misra, overflow, sans-top25-ri...

https://sonarcloud.io/dashboard?id=org.eclipse.che%3Ache-parent%3Amaster

Static Analysis

- Run static analysis before integration on a sprint/master branch
  - nobody wants to break the build for a defect the static analysis tool would have caught!
  - No developers wants to review code containing bugs that static analysis tools would have caught
    - Developer time is far more important and costly ($$$)

Static Analysis: Measuring Defect Removal Effectiveness

- Direct Measurement: Just count #defects repaired

- Remember… finding a defect doesn't remove it

Static Analysis: Activities Removing the Same Defects

- Compare to manual code review but much, much cheaper

Static Analysis: Advantages

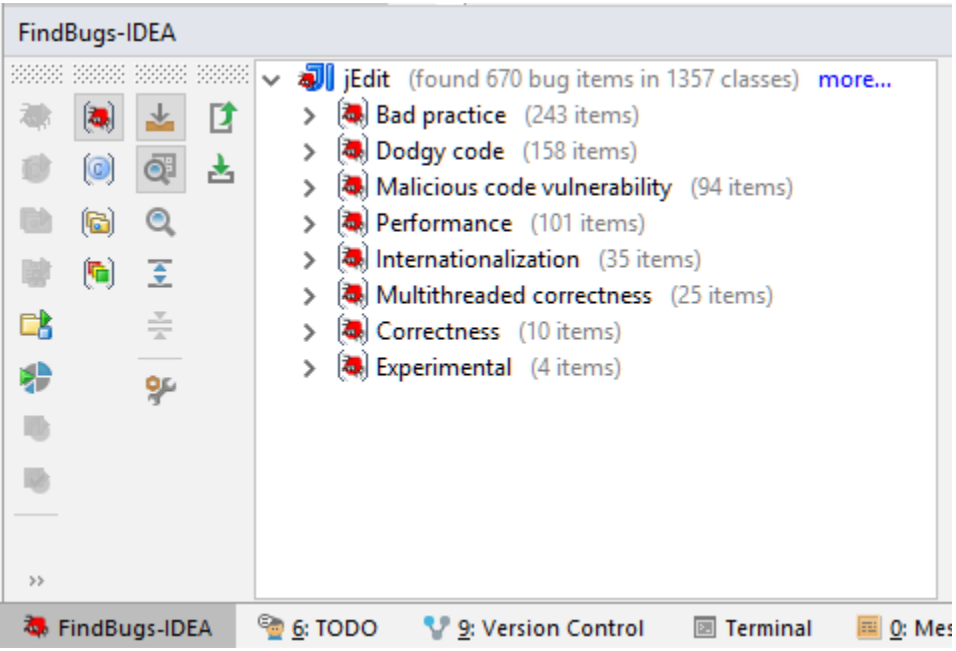
- Ridiculously simple and easy to use

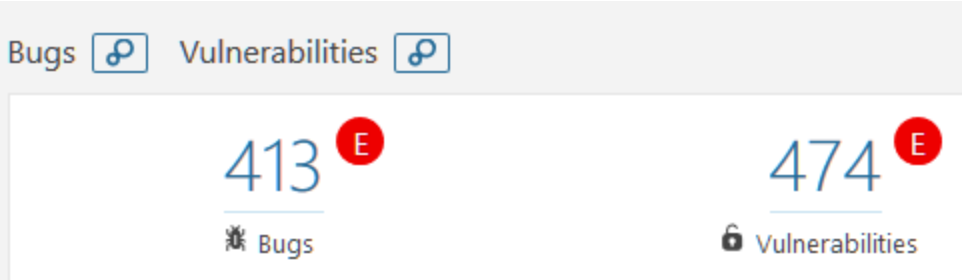
- High-end tools can find cross-class problems, e.g.:
  - dereferencing a null pointer
  - memory leaks
  - security vulnerabilities
  - division by zero, etc.
- that even code reviews may overlook

Static Analysis: Disadvantages

- Some tools can report noise* rather than customer-visible defects
  - Developers need to configure the rules

*large number of "errors/warning/code smells" that aren't important enough to fix by developers





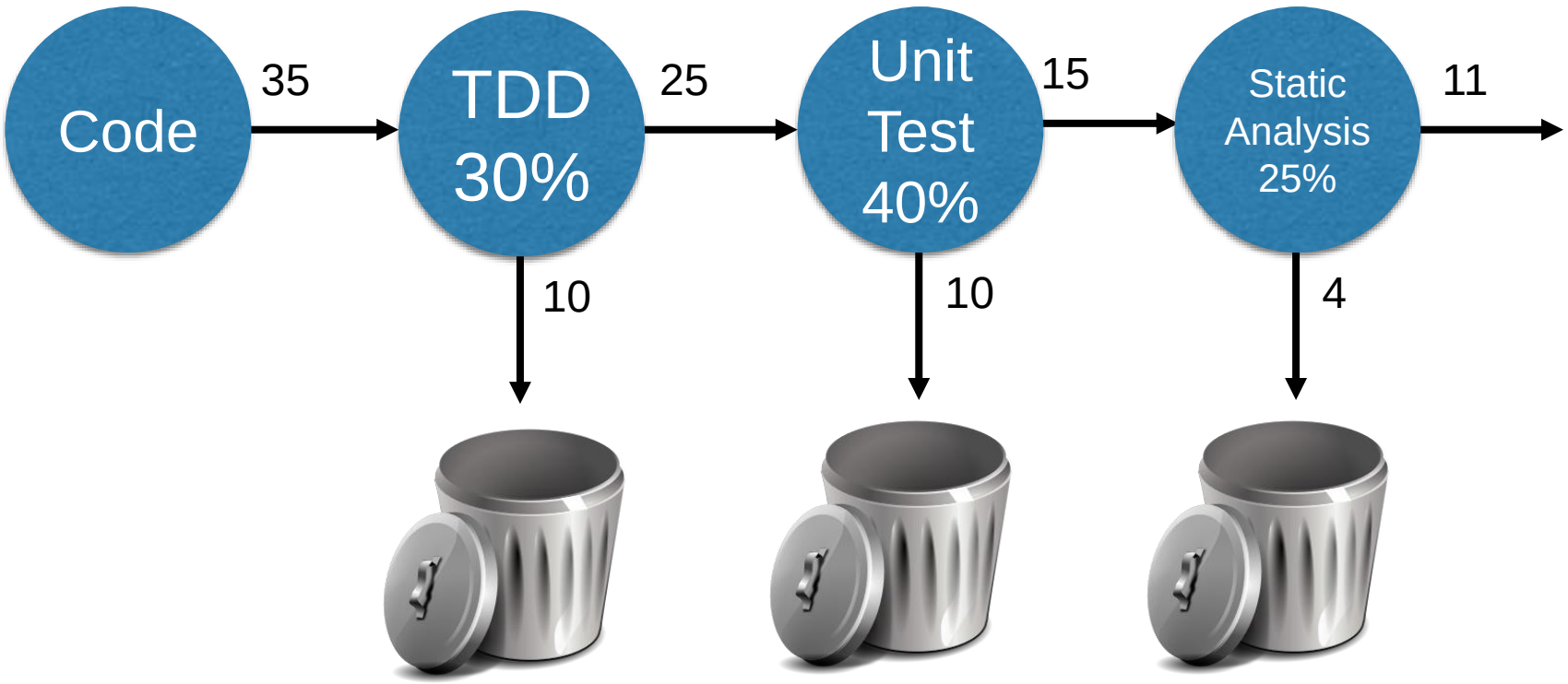
Static Analysis: Modeling

- In real-world development, Static Analysis probably should follow TDD and Unit-Level Testing

- And should always proceed a manual Code Review (to avoid wasting the Code Review team's time on what can be automated)

- Effectiveness claims vary widely (15..90%) depending upon the definition of a "defect"

Modeling Static Analysis



Code Reviews

Code Reviews

- Includes:
  - Formal Code Inspections
  - Code Reviews
  - Peer Reviews
  - Code Walkthroughs — any examination of the source code to find defects

Code Review

- Code reviews place a burden on other Developers
- You likely want to clean-up your code as much as possible before sending it out for review by others

Where to Perform Code Reviews?

- And you likely want to have it reviewed before you push it to the remote repository and share its defects with your Team by breaking the build (or worse!)
  - That means… you probably want to perform code reviews on either your local repository…
  - Or on a feature/story branch if you're using them (as your exposure is limited to just those few other developers contributing to Tasks on the same story)
  - Pull-Requests are great for code reviews

Code Review within Pull Request Example

implementation of task #

GitHub, Inc. [US] | https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3

Conversation 1 | Commits 1 | Files changed 1

Changes from all commits | 1 file | +5 −0

Unified | Split | Review changes

implementation of task #19

bgdndit committed 12 minutes ago

commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 src/Main.java

Show comments | View

```
@@ -1,5 +1,10 @@
 public class Main
 {
+    private void demoMethodTask19()
+    {
+        //implementation of task #19 in testPRBranch
+    }
+
```

Write | Preview

leave feedback directly in the code...

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Cancel | Add single comment | Start a review

Code Review within Pull Request Example

implementation of task #

GitHub, Inc. [US] | https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3

Conversation 1 | Commits 1 | Files changed 1

Changes from all commits | 1 file | +5 −0

Unified | Split | Review changes

implementation of task #19

bgdndit committed 12 minutes ago

commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 src/Main.java

Show comments | View

```
@@ -1,5 +1,10 @@
 public class Main
 {
+    private void demoMethodTask19()
+    {
+        //implementation of task #19 in testPRBranch
```

Write | Preview

leave feedback directly in the code...

pping, selecting them, or pasting from the

orted

ncel | Add single comment | Start a review

Avoid communication overhead (e.g., specifying which file, version to checkout via email)

Code Review: Why does it work?

- "Another set of eyes" will often see something, especially integration problems, you miss

- The act of explaining how your code works sometimes reveals its shortcomings to you

- Group Synergy: The result of a group can exceed the sum of its individuals working alone.
  - Somehow, the interactions of the group prompt its individuals to higher performance than possible when working alone.

Code Review: Many Variations

- Informal Code Review, Peer Review: Development team meets and discusses the source code
- Walkthrough: Author leads the development team through the code, explaining how it works
- Tool Assisted: e.g., GitHub Pull-Requests Workflow, Gerrit

Code Review: Many Variations

- Informal Code Review, Peer Review: Development team meets and discusses the source code

- Walkthrough: Author leads the development team through the code, explaining how it works

- Tool Assisted: e.g., GitHub Pull-Requests Workflow, Gerrit

gerrit / gerrit-server/src/main/java/com/google/gerrit/server/change/**PatchSetInserter.java**

```
106  private PatchSet patchSet;
107  private ChangeMessage changeMessage;
108  private SshInfo sshInfo;
109  private ValidatePolicy validatePolicy = ValidatePolicy.GERRIT;
110  private boolean draft;
111  private boolean runHooks;
112  private boolean sendMail;
113  private Account.Id uploader;
114  private BatchRefUpdate batchRefUpdate;
115
116  @Inject
117  public PatchSetInserter(ChangeHooks hooks,
118      ReviewDb db,
```

```
110  private PatchSet patchSet;
111  private ChangeMessage changeMessage;
112  private SshInfo sshInfo;
113  private ValidatePolicy validatePolicy = ValidatePolicy.GERRIT;
114  private boolean draft;
115  private boolean runHooks = true;
```

Stefan Beller — Why do you move this out of the constructor? Initially I assumed this… Jan 28 2:55 PM
Dave Borowitz — Because it would be identical between the two constructors, so it sa… Jan 28 3:19 PM

```
116  private boolean sendMail = true;
117  private Account.Id uploader;
118  private BatchRefUpdate batchRefUpdate;
119
120  @AssistedInject
121  public PatchSetInserter(ChangeHooks hooks,
122      ReviewDb db,
```

Code Reviews: Measuring Defect Removal Effectiveness

- Direct Measurement:
  - Count the number of repaired bugs
- The effectiveness of the formal inspection process can achieve 65%*
- The informal reviews used in CS471/CS481 perform at the 25..30% level

* (Jones, "Measuring Defect Potentials and Defect Removal Effectiveness" 2008)

Code Reviews: Advantages

- Cost to repair defects found by Code Reviews is usually low because defect reports often cite a particular method or even line-of-code where the problems lie
- Lightweight code reviews can be cost effective in terms of #defectsRemoved / dollar
- Shared code ownership

Code Reviews: Disadvantages

- Some studies report that as many as 75% of the defects found by Code Reviews are "code evolution" (health) problems, not customer-visible functionality
  - If your team is not repairing these problems, then Code Reviews are wasting their time

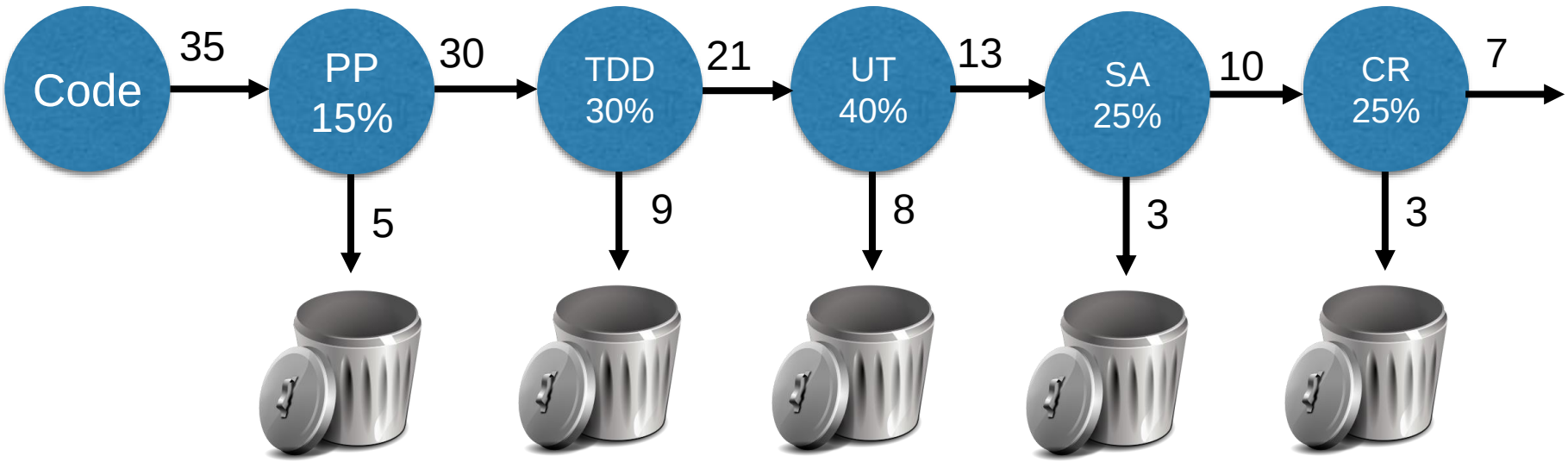
- The formal heavyweight approaches may be expensive in terms of #defectsRemoved / dollar

Code Reviews: Modeling

- Code Reviews should usually be placed following Static Analysis in your defect removal pipeline

- In CS471/CS481, you can expect 25% effectiveness

Modeling Everything Discussed So-Far



Exercise: Code Review (Whiteboard only)

```
public class Triangle {
  int a, b, c;     //The lengths of the three sides
  Triangle(int a, int b, int c) {
    a = a;
    b = b;
    c = c;
  }
  //Two equal sides
  boolean isIsosceles() {
    if (a==b) return true;
    if (b==c) return true;
    return false;
  }
  //Three equal sides
  boolean isEquilateral() {
    if ((a==b) && (b==c)) return true;
    return false;
  }
  //Three unequal sides
  boolean isScalene() {
    if ((a!=b) && (b!=c)) return true;
    return false;
  }
}
```

"Upstream"* Defect Removal Activities

- Pair Programming
- Test-Driven Development
- Unit-Level Testing
- Static Analysis
- Code Reviews

*"Up-Front"/Before integration

"Downstream"* Defect Removal Activities

- Integration Testing
- Regression Testing
- System-Level Testing
  - Acceptance Testing
  - Beta Testing

*"Back-End"/After Integration

Integration Testing

Integration-Level Testing: Description

- Verifies that a subset of the units work well together
- Exercises sub-systems (larger than a unit, smaller than the entire system):
  - package(s)
  - two or more classes working together
- Preferably initiated by an automated build system
- May use *test dummies / fakes / stubs / spyes / mocks / doubles* for major sub-systems (e.g., database)

Unit Test vs. Integration Test

Unit Test vs. Integration Test

See animated version at:

https://media.giphy.com/media/307rbPDRHIHwbmcOBy/giphy.gif

Integration-Level Testing Example

- Module A: "Retrieve Image from Database"
  - should have Unit Tests
- Module B: "Display Image"
  - should have Unit Tests
- Integration tests (Module A + Module B):
  - "Will the code be able to display data loaded from a database?"

Integration-Level Testing: Effectiveness

- 1996 Jones: 25..40%
- Kan*: 36%
- Values may be different for modern projects
- CS471 expectation: <20%

* Kan, Stephen. *Metrics and Models in Software Quality Engineering 2nd Edition*. Addison Wesley. 2003.

Integration-Level Testing: Why do it?

- Cheapest downstream activity focused on integration defects

- Is sometimes the last fully-automated (i.e., cheaper than manual tests) defect removal activity
  - Cost ($) is low ⇒ can be executed on <u>every</u> build
  - Helps to discover if you broke-the-build

- Most thorough regression testing activity if you don't have automated acceptance tests executing on every build

Regression Testing

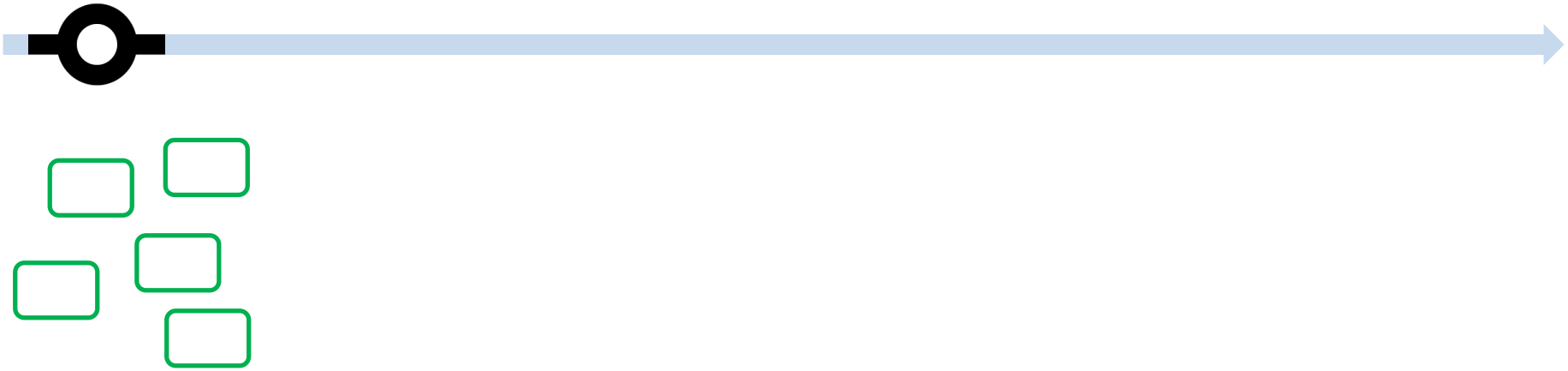
Regression Testing

- Re-runs tests that previously passed
- Looking for code changes that introduced new defects into previously working source code
  - Did my code change break your perfectly working code?
- Can be performed at many levels
  - unit
  - integration (preferred)
  - system
- Can use both white and black-box approaches
- Works well if automated in Continuous Integration

Throughout development, team adds Unit-, Integration- and System-level tests

Legend

- Unit-level Test
- Integration-level Test
- System-level Test
- Commit



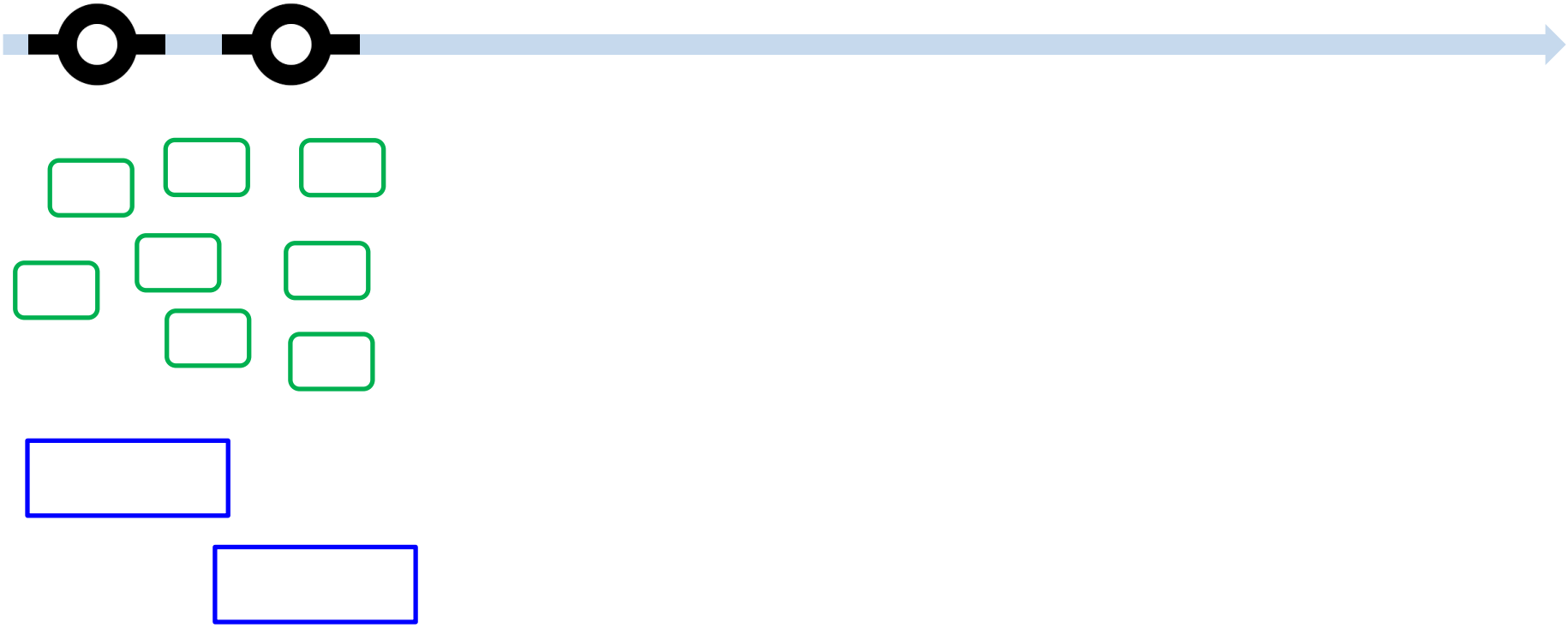
Legend

Unit-level Test

Integration-level Test

System-level Test

Commit



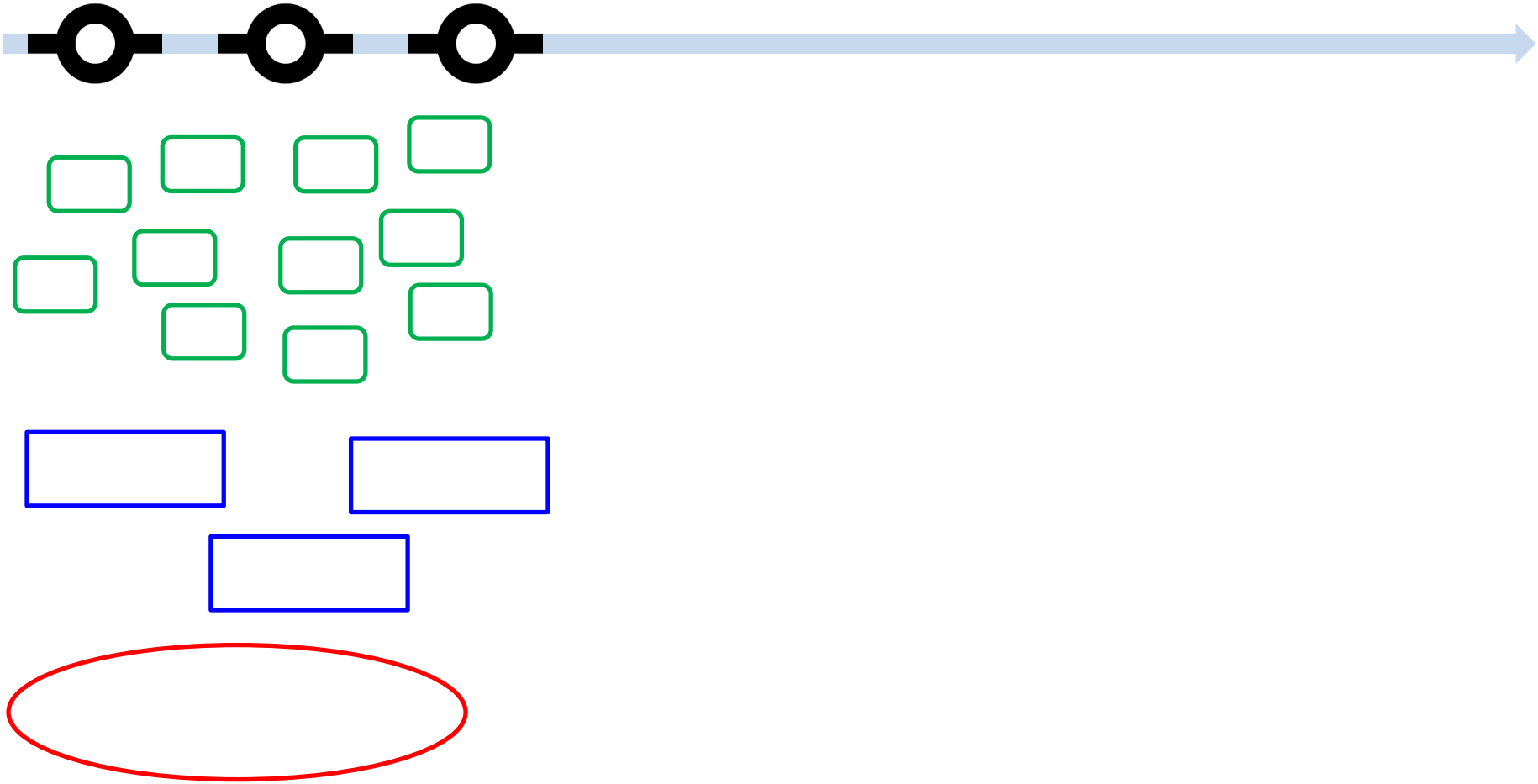
Legend

Unit-level Test

Integration-level Test

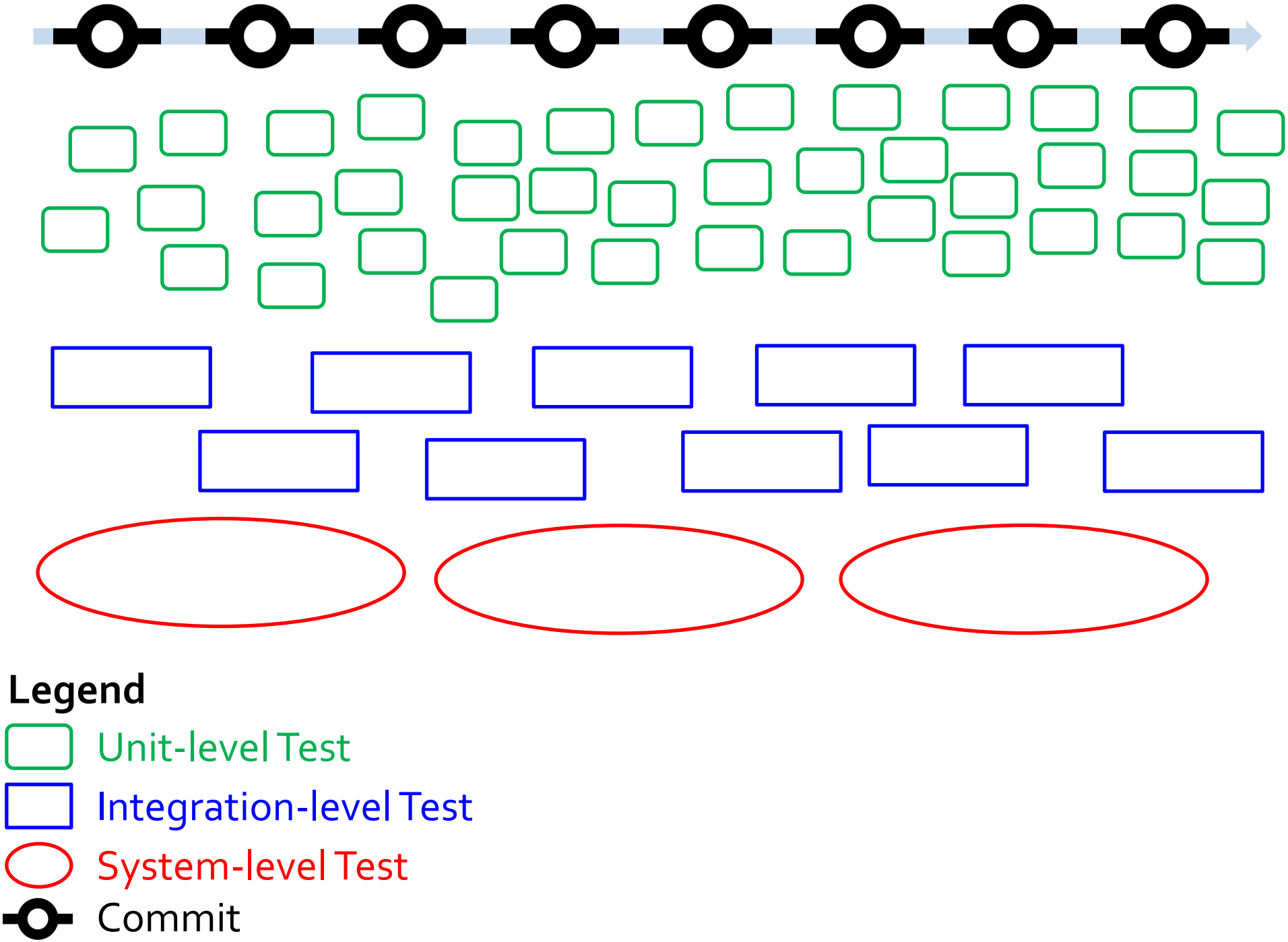
System-level Test

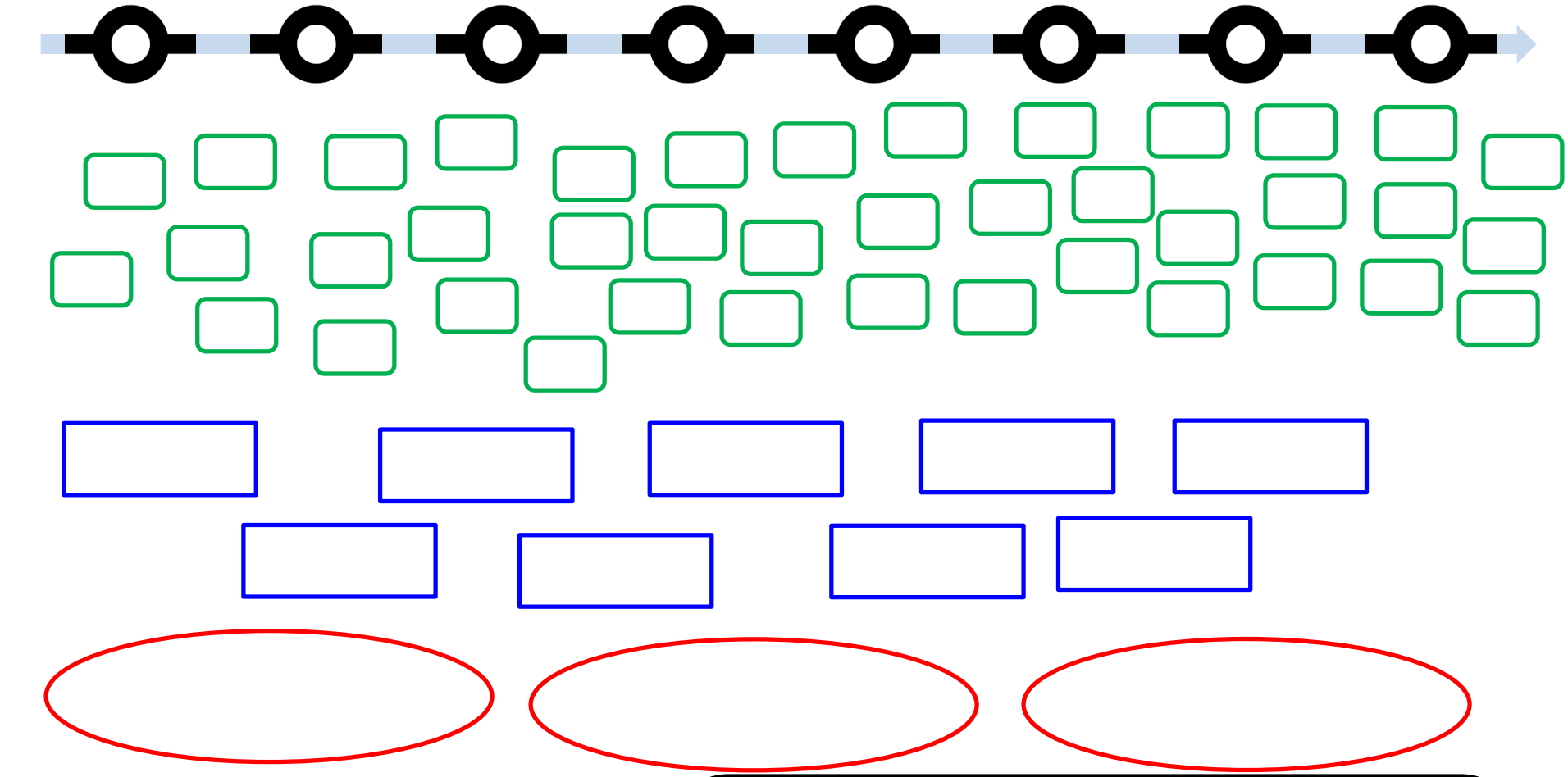
Commit



Legend

- Unit-level Test
- Integration-level Test
- System-level Test
- Commit


Legend
Unit-level Test
Integration-level Test
System-level Test
Commit



Legend

- Unit-level Test
- Integration-level Test
- System-level Test
- Commit

For each new commit, the existing Unit-, Integration- and System-level tests become regression tests (that ideally should be run automatically)

System Testing

System-Level Testing

- Exercising the fully-assembled product
  - Example: Acceptance Testing, Beta Testing

- Verifies that all the individual units work together

White Box Approach

- Exercises the internal structure of the source code as opposed to its external functionality
  - aka Structural Testing

- Example: Code Review

- NB: Testers must have access to the source code

Black Box Approach

- Testers don't have access to the source code
- Exercises the product without examining its internal structure
  - Focus on input / output
- Example:
  - Acceptance Testing
  - Stress Test
  - Any exercise of an interface's functionality without examination of its underlying source code

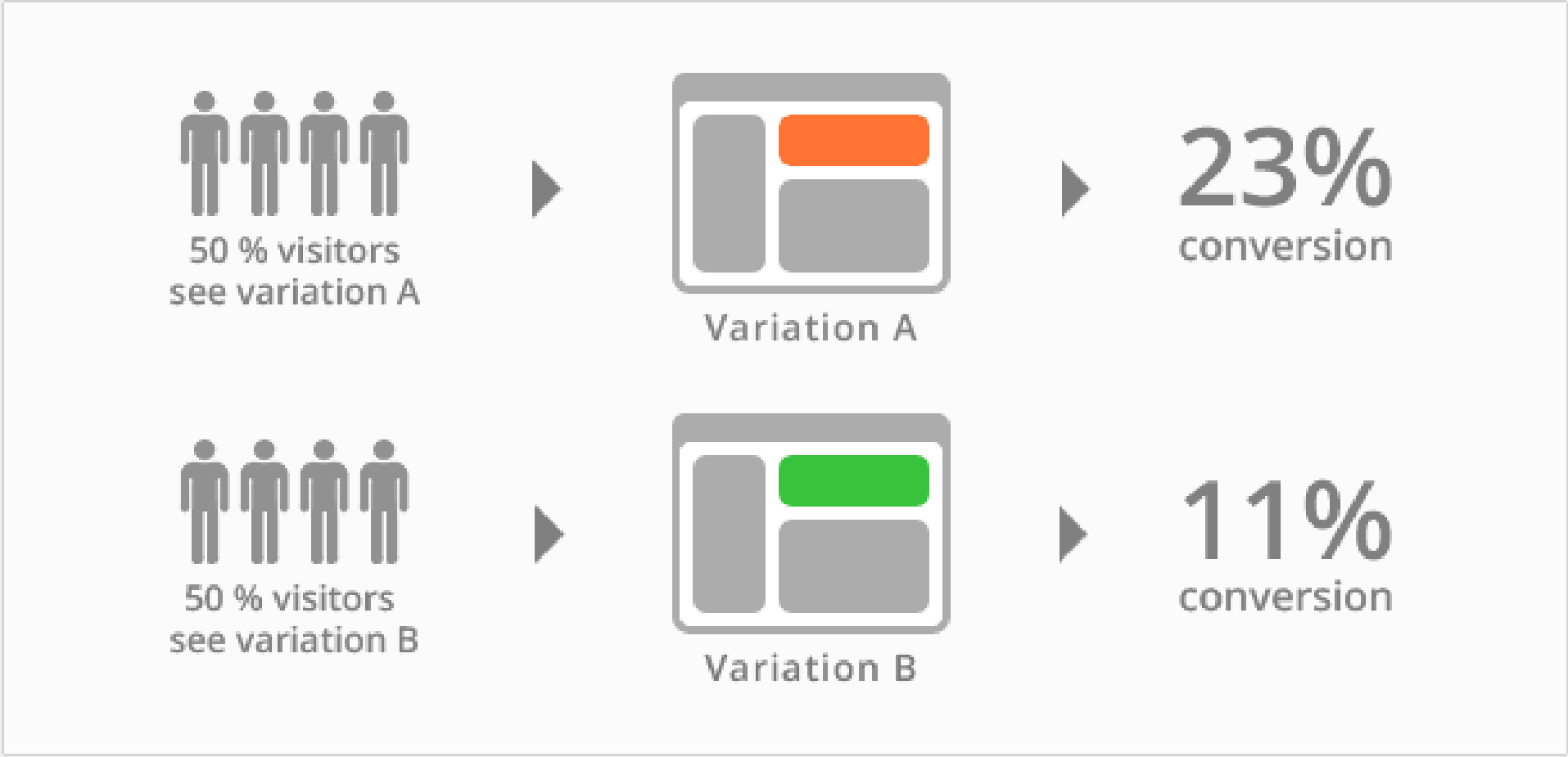
Functional Testing Approach

- A form of System-Level Black Box testing of a product's functionality

- Yes, there is non-functional testing:
  - performance testing
  - usability testing
  - A/B testing
  - load/stress testing
  - reliability testing
  - etc.

Example A/B testing

- Compare variation A with B



https://vwo.com/ab-testing/

Acceptance Testing

Acceptance Testing

- System-level, black-box test of the entire product
- Often dominated by a functional approach
  - But may include usability, reliability, stress & performance
- "Acceptance" refers to the customer's "Acceptance Criteria" (i.e., how will the customer know if a User Story is "Done")

Manual vs. Automated Testing

- *Manual Testing*
  - Test procedure executed by a human (e.g., keyboard-and-mouse)
  - Preferably following a written script to promote repeatability

- *Automated Testing*
  - Computer executed testing
    - NB: automation refers to the execution of the test, not the analysis and repair of the underlying problem
  - Executable script (shell or special-purpose test tool or harness)

Acceptance Testing: Manual / Automated?

- Acceptance Testing is often performed on a GUI

- Manual GUI tests may not yield reproducible results without a written "test procedure" (a script for humans)

- Automation can be a significant investment
  - https://en.wikipedia.org/wiki/List_of_GUI_testing_tools
  - http://blog.dreamcss.com/tools/gui-testing-tools/
  - Automation works best when you need to run the same test over and over without changing the GUI

Selenium – Automating Web Testing

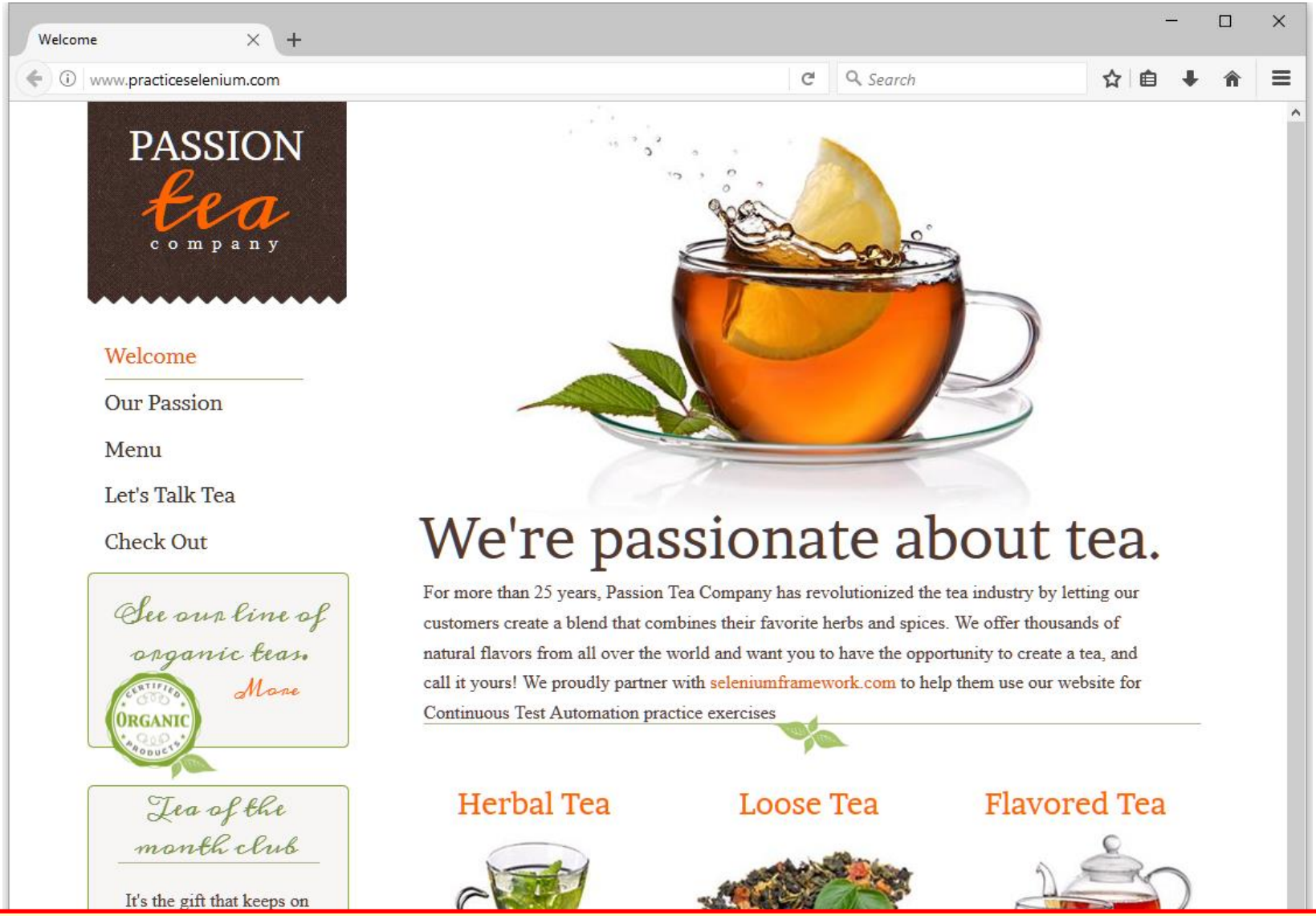
- *Selenium automates browsers*
- Facilitates testing of web applications automatically
- Selenium "is like jUnit for testing GUI and Web applications"
  - write your tests in: Java, C#, Ruby, Python, JS (Node)
  - run your tests in: Firefox, Chrome, IE, Safari, etc.

Sample Java code using Selenium WebDriver that automates executing a web page in Firefox

```
WebDriver driver = new FirefoxDriver();
driver.get("http://www.practiceselenium.com/");
System.out.println("Successfully loaded the website " + "http://www.practiceselenium.com/");
Thread.sleep(2000);
WebElement elementMenu = driver.findElement(By.linkText("Menu"));
elementMenu.click();
Thread.sleep(2000);

WebElement elementCheckout = driver.findElement(By.linkText("Check Out"));
elementCheckout.click();
Thread.sleep(2000);

WebElement elementEmail = driver.findElement(By.id("email"));
elementEmail.sendKeys("johndoe@someemail.com");
Thread.sleep(100);

WebElement elementName = driver.findElement(By.id("name"));
elementName.sendKeys("John Doe");
//…
Select selectionCardType = new Select(driver.findElement(By.id("card_type")));
selectionCardType.selectByVisibleText("Mastercard");
//…
WebElement elementPlaceOrder =
driver.findElement(By.xpath("/html/body/div/div/div[1]/div/div[1]/div/div/form/div/button"));
elementPlaceOrder.click();
Thread.sleep(2000);
System.out.println("TODO: Assert if checkout operation was successful");
driver.quit();
```

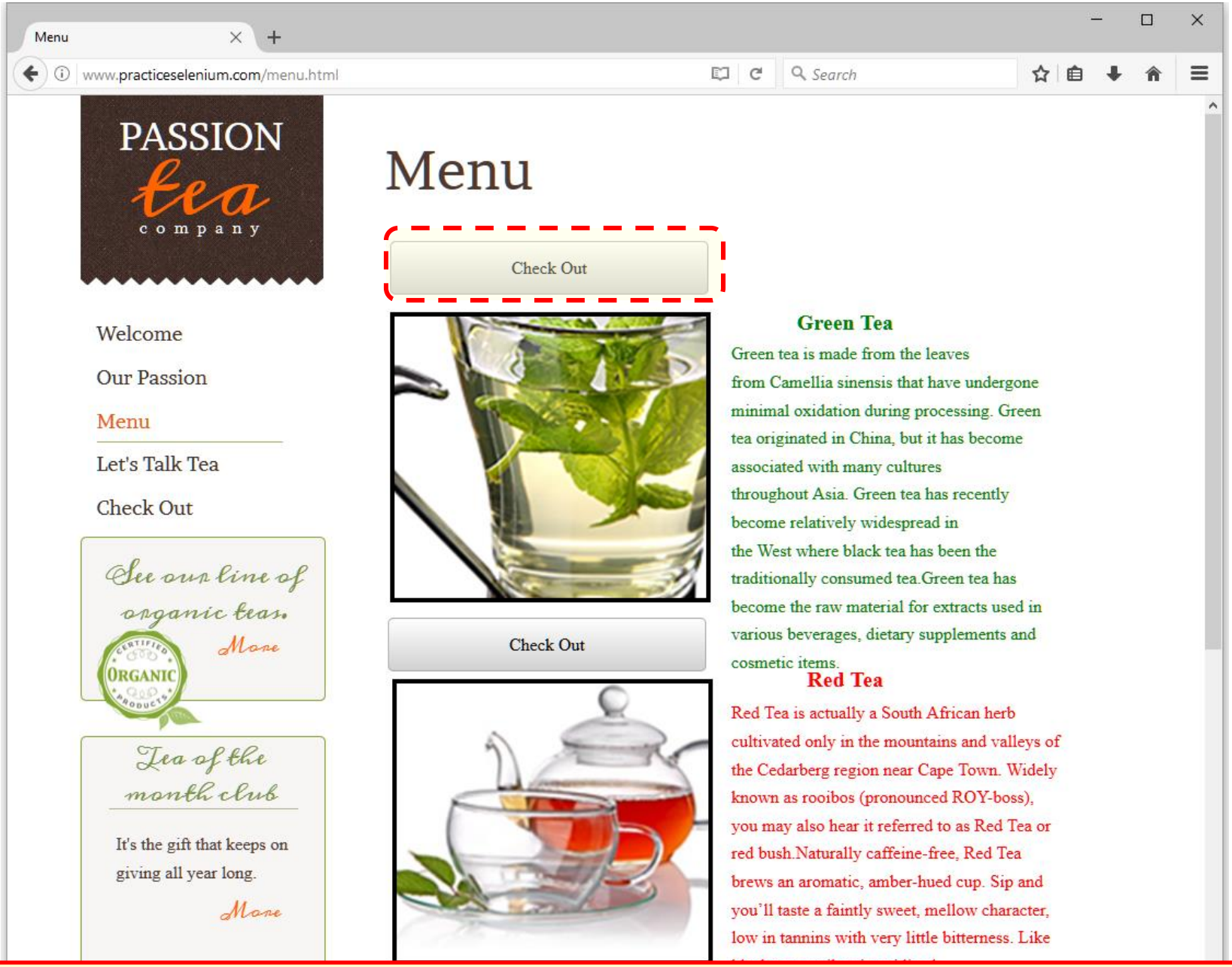
```
WebDriver driver = new FirefoxDriver();
driver.get("http://www.practiceselenium.com/");
System.out.println("Successfully loaded the website " + "http://www.practiceselenium.com/");
Thread.sleep(2000);
WebElement elementMenu = driver.findElement(By.linkText("Menu"));
elementMenu.click();
Thread.sleep(2000);

WebElement elementCheckout = driver.findElement(By.linkText("Check Out"))
elementCheckout.click();
Thread.sleep(2000);

WebElement elementEmail = driver.findElement(By.id
elementEmail.sendKeys("johndoe@someemail.com
Thread.sleep(100);

WebElement elementName = d
id("name"));
elementName.sendKeys(
//…
Select selecti
w Select(driver.findElement(By.id("card_type")));
selectionCardTy
cByVisibleText("Mastercard");
//…
WebElement elementPlaceOrder =
driver.findElement(By.xpath("/html/body/div/div/div[1]/div/div[1]/div/div/form/div/button"));
elementPlaceOrder.click();
Thread.sleep(2000);
System.out.println("TODO: Assert if checkout operation was successful");
driver.quit();
```

Load a page and execute a checkout process

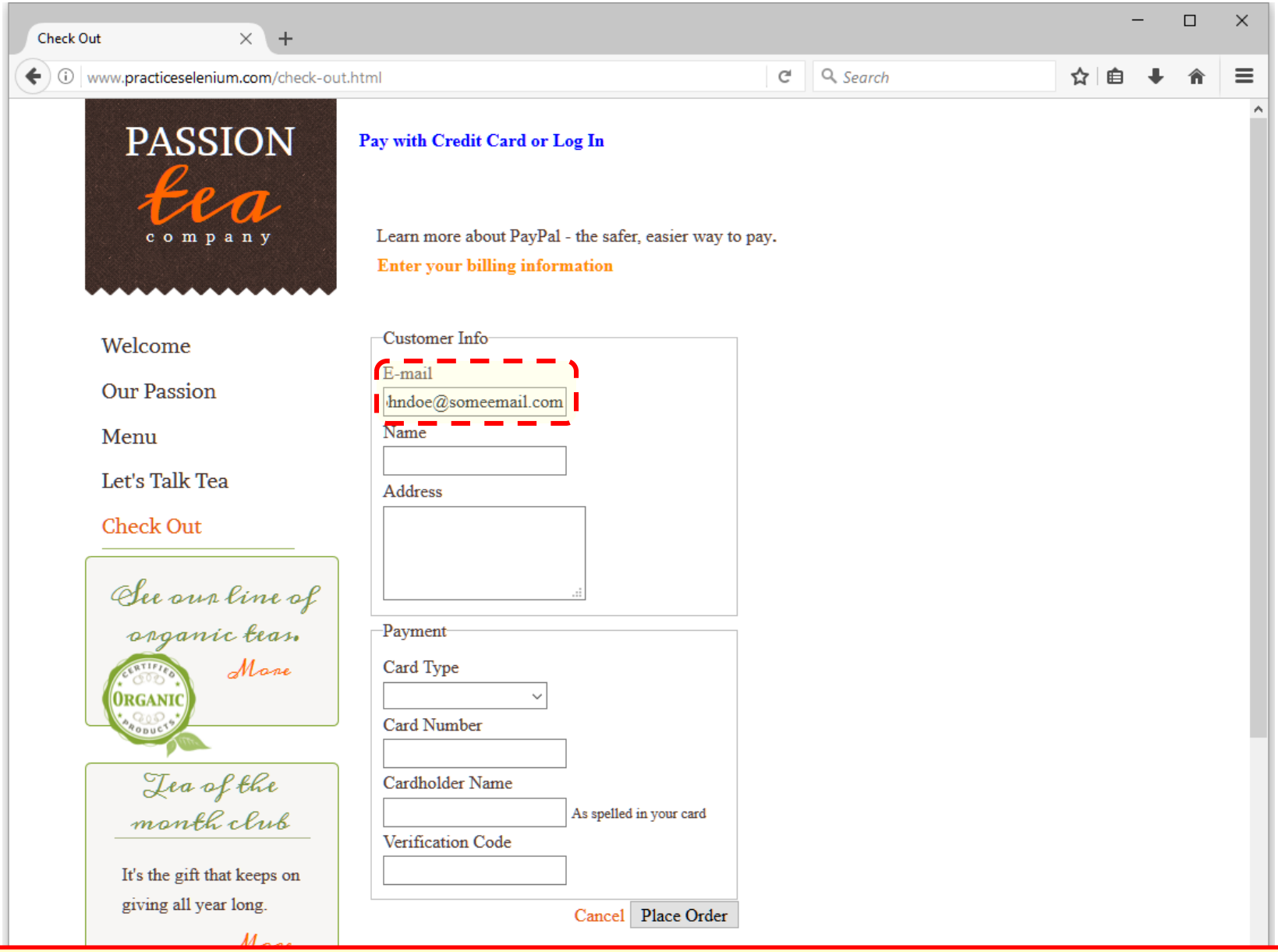



```
//start a new instance of Firefox and load the web page
WebDriver driver = new FirefoxDriver();
driver.get("http://www.practiceselenium.com/");
```


```
WebElement elementMenu = driver.findElement(By.linkText("Menu"));
elementMenu.click();
```




```
WebElement elementCheckout = driver.findElement(By.linkText("Check Out"));
elementCheckout.click();
```




```
//fill email address text field
WebElement elementEmail = driver.findElement(By.id("email"));
elementEmail.sendKeys("johndoe@someemail.com");
```

Use FireBug (http://getfirebug.com/) to inspect elements and get their XPaths if needed

Check Out

www.practiceselenium.com/check-out.html

Check Out

See our line of organic teas.

More

Tea of the month club

It's the gift that keeps on giving all year long.

More

Address

Payment

Card Type

Card Number

Cardholder Name

As spelled in your card

Verification Code

Cancel Place Order

1

2

3

Console HTML CSS Script DOM Net Cookies

Search by text or CSS selector

Edit button.bt...-primary div.form-actions form.form-horizontal div.container div.wsb-h...-element

```
<form class="form-horizontal" action="menu.html" method="get">
  <fieldset>
  <fieldset>
  <div class="form-actions" style="text-align: right">
    <a class="btn" href="menu.html">Cancel</a>
    <button class="btn btn-primary">Place Order</button>
  </div>
</form>
</div>
</div>
</div>
```

Style Computed Layout DOM Events

```
button, html                    site.cs...6678408 (line 1)
input[type="button"],
input[type="reset"],
input[type="submit"]
{
    cursor: pointer;
}

button, select {                site.cs...6678408 (line 1)
    text-transform: none;
}
```

Use FireBug (http://getfirebug.com/) to inspect elements and get their XPaths if needed

Check Out

www.practiceselenium.com/check-out.html

Check Out

See our line of organic teas.

More

Tea of the month club

It's the gift that keeps on giving all year long.

More

Address

Payment

Card Type

Card Number

Cardholder Name

As spelled in your card

Verification Code

Cancel Place Order

Console HTML CSS Script DOM Net Cookies

Search by text or CSS selector

Edit button.bt...-primary div.form-actions form.form-horizontal div.container div.wsb-h...-element

```
<form class="form-horizontal" action="menu.html" method="get">
  <fieldset>
  <fieldset>
  <div class="form-actions" style="text-align: right">
    <a class="btn" href="menu.html">Cancel</a>
    <button class="btn btn-primary">Place Order</button>
  </div>
</form>
</div>
</div>
</div>
```

4

Copy HTML

Copy innerHTML

Copy XPath

Copy CSS Path

Style Computed Layout DOM Events

```
button, html                    site.cs...6678408 (line 1)
input[type="button"],
input[type="reset"],
input[type="submit"]
{
    cursor: pointer;

ect {                           site.cs...6678408 (line 1)
nsform: none;

t {                             site.cs...6678408 (line 1)
```

Check Out

www.practiceselenium.com/check-out.html

PASSION tea company

Pay with Credit Card or Log In

Learn more about PayPal - the safer, easier way to pay.

Enter your billing information

Welcome

Our Passion

Menu

Let's Talk Tea

Check Out

See our line of organic teas.

More

Tea of the month club

It's the gift that keeps on giving all year long.

More

Customer Info

E-mail

hndoe@someemail.com

Name

John Doe

Address

123 Main Street
Beverly Hills, CA 90210

Payment

Card Type

Mastercard

Card Number

1234567890123456

Cardholder Name

John Doe as well... As spelled in your card

Verification Code

555

Cancel Place Order

```
WebElement elementPlaceOrder =
driver.findElement(By.xpath("/html/body/div/div/div[1]/div/div[1]/div/div/form/div/button"));
elementPlaceOrder.click();
```

Selenium Resources

- http://docs.seleniumhq.org/
  - Official Selenium page where you can download the Selenium WebDriver from

- http://toolsqa.com/selenium-tutorial/
  - very detailed and easy to follow tutorial for getting up to speed running with Selenium in Eclipse to test web apps.

Selenium Resources

- http://www.seleniumframework.com/demo-sites/
  - provides a list of websites to practices your Selenium tests

- http://www.techbeamers.com/websites-to-practice-selenium-webdriver-online/
  - provides a list of websites to practices your Selenium tests

Marathon – Automating GUI Testing

- Cross-platform GUI test automation framework for:
  - Java/Swing
  - Java/FX
  - Web applications
- Very similar in functionality/usage with Selenium
- Allows to test Java GUI apps

Marathon Resources

- https://marathontesting.com/
  - Official Marathon page

- https://confengine.com/selenium-conf-2016/proposal/2532/java-swing-java-fx-application-testing-using-selenium-webdriver
  - 1 hour video tutorial to get started
    - sample code used in video (can be easily imported as an Eclipse project) and tested:
      - https://github.com/jalian-systems/marathon-demo

Eggplant – Automating Web/GUI Testing

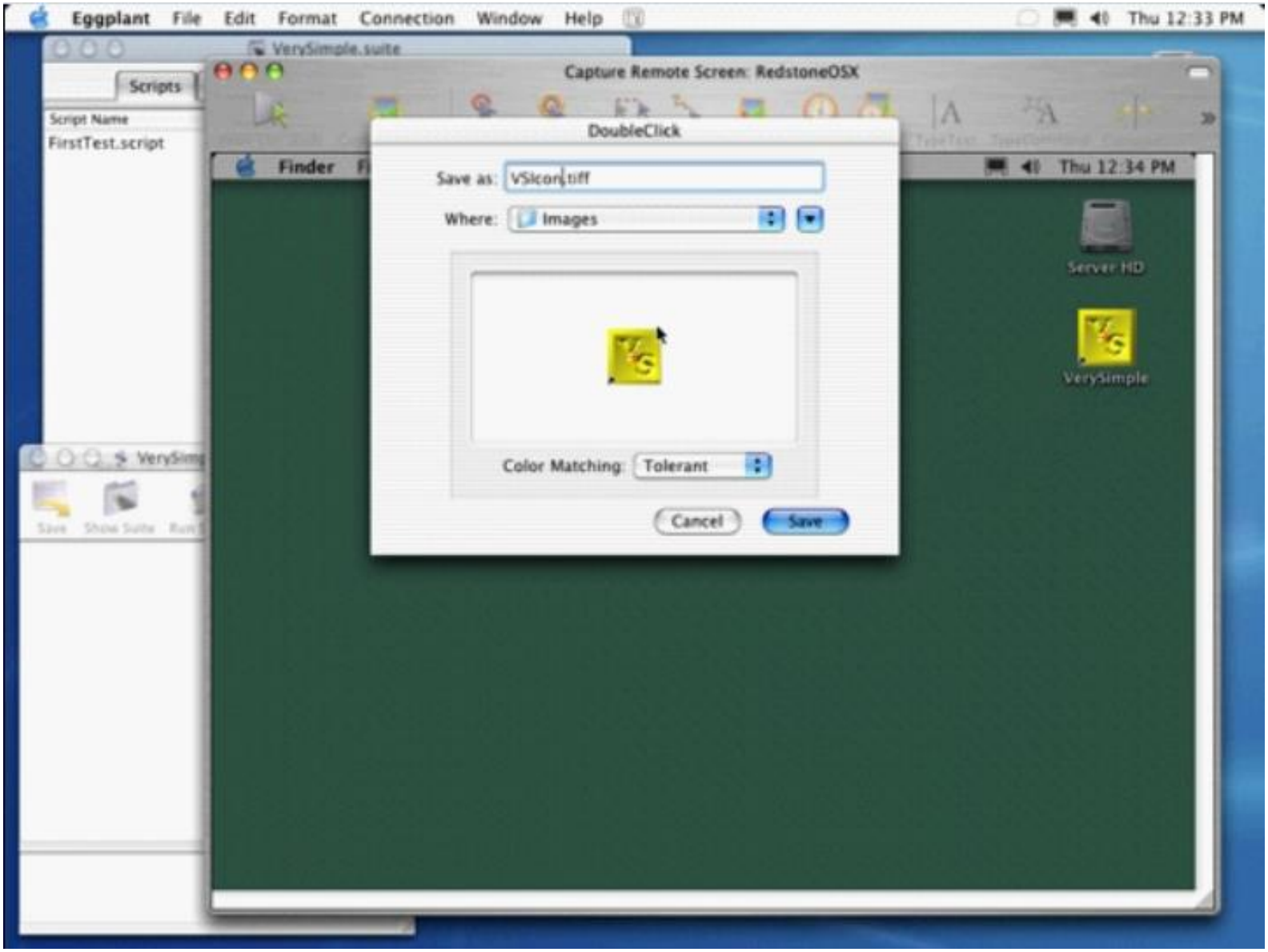
- Cross-platform GUI test automation framework for any application

- It uses image analysis/recognition to identify:
  - which button to press
  - where to input text
  - where to scroll
  - etc.

- https://www.testplant.com/eggplant/testing-tools/

Short video demoing Eggplant functionality

https://vimeo.com/27490840



Acceptance Testing: Effectiveness

- 1996Jones: 25..50%
- 2003Kan (IBM): 57%
- CS481 indirect measurement: 40%

Everybody Does Acceptance Testing

- It's likely the most commonly used defect removal activity
  - It is mandatory to do it
- It finds defects that customers actually care about!
  - However…

Acceptance Testing (AT): Warnings

- By itself, AT is inadequate
  - unless your business can tolerate shipping half your defects to customers

- Management will sometimes react to quality problems by throwing money at AT
  - The cost of removing the next defect with AT rises dramatically (and reaches a saturation point)
  - Solution: complement AT with another defect removal activity