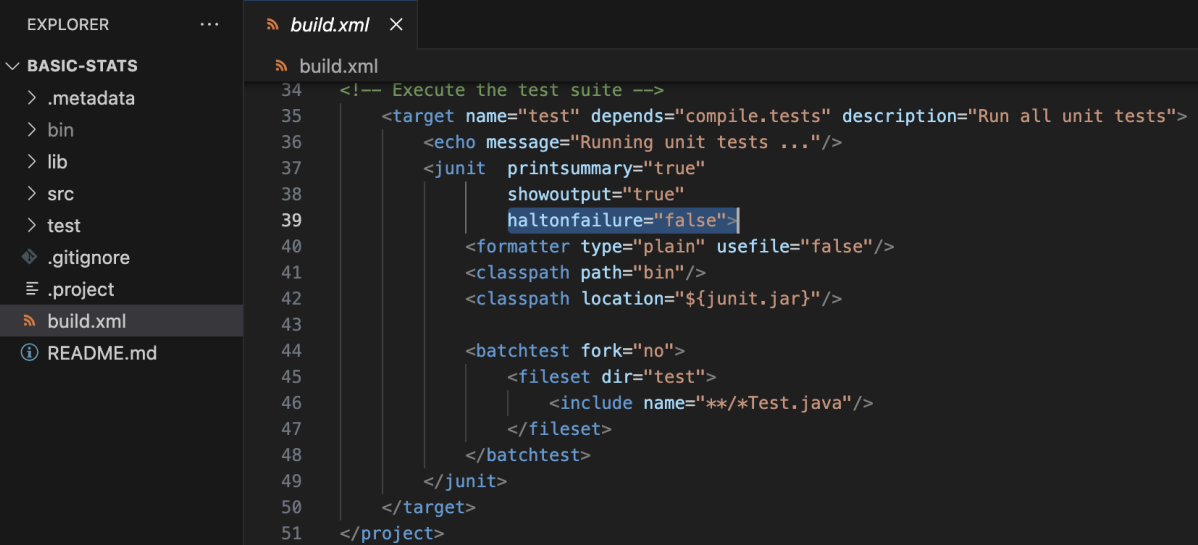


IN-CLASS – 3 EXERCISE

Team: CodeCraftersH

Members: Harshita Loganathan, Hemangani Nagarajan

1. Why does the described automated testing infrastructure not catch the defect?



```
34 <!-- Execute the test suite -->
35 <target name="test" depends="compile.tests" description="Run all unit tests">
36     <echo message="Running unit tests ..."/>
37     <junit printsummary="true"
38           showoutput="true"
39           haltonfailure="false">
40         <formatter type="plain" usefile="false"/>
41         <classpath path="bin"/>
42         <classpath location="${junit.jar}"/>
43
44         <batchtest fork="no">
45             <fileset dir="test">
46                 <include name="**/*Test.java"/>
47             </fileset>
48         </batchtest>
49     </junit>
50 </target>
51 </project>
```

The automated testing infrastructure's failure to detect the defect can be attributed to a confluence of various factors. To begin with, the Ant build configuration may be configured to not terminate the build in the event of test failures, as evidenced by the JUnit task's `haltonfailure="false"` parameter. As a consequence, the `reportBug` command would remain inactive, irrespective of test failures. The setting `haltonfailure="false"` allows the build to succeed even if there are test failures, which is why the automated testing system failed to detect the defect.

2. How could the developers improve the testing infrastructure (either manual or automated) to immediately notice test failures in the future?

To enhance our testing infrastructure and quickly identify test failures, we should:

1. Adjust our Ant build configuration to halt the build if tests fail, by setting `haltonfailure="true"` in our JUnit task.
2. Implement a Continuous Integration (CI) system like Travis CI, Jenkins, or GitHub Actions for automatic test execution and immediate failure notifications.
3. Expand test coverage, including edge cases in unit tests, adding integration tests, and end-to-end tests for realistic user scenarios.
4. Use static code analysis tools like SonarQube for bug detection, and code coverage tools like JaCoCo to identify untested code.
5. For GUI applications, integrate automated UI testing tools like TestComplete or Selenium, and perform regular performance testing.
6. Combine automated testing with consistent manual testing, especially for user experience and usability, and encourage exploratory testing.
7. Implement mandatory code reviews, provide training in effective testing methodologies, and regularly update testing documentation.
8. Establish real-time monitoring systems with alerting mechanisms and comprehensive logging systems for quick issue detection and resolution.

3. For git bisect, how many commits exist between v1.0.0 and the HEAD revisions (including v1.0.0 and HEAD)? What command(s) did you use to determine the number?

```
harshitaloganathan@Harshitas-MacBook-Air basic-stats % git rev-list --count v1.0.0..HEAD
```

36

Command Used:

git rev-list --count v1.0.0..HEAD

4. Based on the git bisect results, which commit (commit hash and log message) introduced the defect? How did you independently verify (meaning another approach other than git bisect) that this commit indeed introduced the defect?

After using `git bisect` to identify the problematic commit, the commit message "Commented and cleaned up the source code" and its hash, f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f, were noted as the source of the bug. To independently verify this, the preceding commit, bda5f0214b51f6cb77e00a7869f32119daddb47, was examined. This involved checking out to this previous commit and running the Ant build process:

1. `ant clean` was executed to clean the build environment.
2. `ant compile` was used to compile the source code.
3. `ant -lib lib/ test` was run to execute the tests.

The result was zero failures, indicating that this previous commit did not contain the defect. In contrast, when the same steps were applied to the identified buggy commit (f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f), the tests failed. This procedure confirmed that the defect was indeed introduced in the commit with the message "Commented and cleaned up the source code". Attached below are the screenshots of the process involved.

```
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git branch
* master
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect start
status: waiting for both good and bad commits
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect good v1.0.0
status: waiting for bad commit, 1 good commit known
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect bad master
Bisecting: 17 revisions left to test after this (roughly 4 steps)
[cc717c651ac06b5fd7c1767f90947aac58a33586] Updated Override annotations

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
compile:
[javac] Compiling 15 source files to /Users/hemanganinagarajan/git/basic-stats/bin
compile.tests:
test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0.001 sec
[junit] FAILED
[junit] expected:<8.0> but was:<7.5>
[junit] junit.framework.AssertionFailedError: expected:<8.0> but was:<7.5>
[junit]     at BasicStatsTest.testMedian(BasicStatsTest.java:44)
[junit]     at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]     at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]
BUILD FAILED
/Users/hemanganinagarajan/git/basic-stats/build.xml:39: Test BasicStatsTest failed

Total time: 0 seconds
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect bad
Bisecting: 8 revisions left to test after this (roughly 3 steps)
[b6462c91631b264a2e6b252335fe77bc3ca14d268] Removed autogenerated stubs
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
```

```

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:
[javac] Compiling 14 source files to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect good
Bisecting: 4 revisions left to test after this (roughly 2 steps)
[d75539f6b799140f6aa83e88689499e342ca7183] Placed BasicStats into its own Util folder
⊗ hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:
[javac] Compiling 14 source files to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0.001 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0.001 sec
[junit] Testcase: testMedian took 0 sec

```

```

[junit] FAILED
[junit] expected:<8.0> but was:<7.5>
[junit] junit.framework.AssertionFailedError: expected:<8.0> but was:<7.5>
[junit] at BasicStatsTest.testMedian(BasicStatsTest.java:44)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]

BUILD FAILED
/Users/hemanganinagarajan/git/basic-stats/build.xml:39: Test BasicStatsTest failed

Total time: 0 seconds
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect bad
Bisecting: 1 revision left to test after this (roughly 1 step)
[f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f] Commented and cleaned up the source code
⊗ hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:
[javac] Compiling 14 source files to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0.001 sec
[junit] FAILED
[junit] expected:<8.0> but was:<7.5>
[junit] junit.framework.AssertionFailedError: expected:<8.0> but was:<7.5>
[junit] at BasicStatsTest.testMedian(BasicStatsTest.java:44)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]

BUILD FAILED
/Users/hemanganinagarajan/git/basic-stats/build.xml:39: Test BasicStatsTest failed

Total time: 0 seconds

```

```

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect bad
Bisecting: 0 revisions left to test after this (roughly 0 steps)
[bda5f0214b51f6cb77e00a7869f32119daddb47] More refactorings
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:
[javac] Compiling 5 source files to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect good
f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f is the first bad commit
commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:58:11 2016 -0400

    Commented and cleaned up the source code

src/BasicStats.java | 19 ++++++-----
src/Controller.java | 2 +-
src/ModeView.java   | 1 -
src/Model.java      | 1 +
src/ResetCtrl.java  | 1 -
5 files changed, 14 insertions(+), 10 deletions(-)

```

```

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git checkout f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f
M
build.xml
Previous HEAD position was bda5f02 More refactorings
HEAD is now at f7f175d Commented and cleaned up the source code
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git show
commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f (HEAD)
commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f (HEAD)
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:58:11 2016 -0400

    Commented and cleaned up the source code

diff --git a/src/BasicStats.java b/src/BasicStats.java
index dfd8909..9ac00cc 100644
--- a/src/BasicStats.java
+++ b/src/BasicStats.java
@@ -68,27 +68,32 @@ public class BasicStats {
     return mode;
 }

+ //Helper method
+ public static void bubbleSort(double[] numArray) {
+
+     int n = numArray.length;
+     double temp = 0;
+
+     for (int i = 0; i < n; i++) {
+         for (int j = 1; j < (n - i); j++) {
+             int size = numArray.length;
+             double hold = 0;
+
+             //Iterate to bubble up 0(n^2)
+             for (int i = 0; i < size; i++) {
+                 //Second loop
+                 for (int j = 1; j < (size - j); j++) {
+                     //Bubble up
+                     if (numArray[j - 1] > numArray[j]) {
+                         temp = numArray[j - 1];
+                         hold = numArray[j - 1];
+                         numArray[j - 1] = numArray[j];
+                         numArray[j] = temp;
+                         numArray[j] = hold;
+                     }
+                 }
+             }
+         }
+     }
+ }

```

```

+ //Helper method
public static double[] getArrayDouble(ArrayList<Double> doubles) {
    double[] result = new double[doubles.size()];

+ //O(n) iteration to convert result
    for (int i = 0 ; i < doubles.size(); i++) {
        result[i] = doubles.get(i);

...skipping...
commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f (HEAD)
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:58:11 2016 -0400

```

Commented and cleaned up the source code

```

diff --git a/src/BasicStats.java b/src/BasicStats.java
index dfd8909..9ac00cc 100644
--- a/src/BasicStats.java
+++ b/src/BasicStats.java
@@ -68,27 +68,32 @@ public class BasicStats {
    return mode;
}

+ //Helper method
public static void bubbleSort(double[] numArray) {

-    int n = numArray.length;
-    double temp = 0;
-
-    for (int i = 0; i < n; i++) {
-        for (int j = 1; j < (n - i); j++) {
+    int size = numArray.length;
+    double hold = 0;

+    //Iterate to bubble up O(n^2)
+    for (int i = 0; i < size; i++) {
+        //Second loop
+        for (int j = 1; j < (size - i); j++) {
+            //Bubble up
+            if (numArray[j - 1] > numArray[j]) {
-                temp = numArray[j - 1];
-                hold = numArray[j - 1];
-                numArray[j - 1] = numArray[j];
-                numArray[j] = temp;
+                numArray[j - 1] = numArray[j];
+                numArray[j] = hold;
+            }
        }
    }
}

```

```

}
}

+ //Helper method
public static double[] getArrayDouble(ArrayList<Double> doubles) {
    double[] result = new double[doubles.size()];

+ //O(n) iteration to convert result
    for (int i = 0 ; i < doubles.size(); i++) {
        result[i] = doubles.get(i);
    }
}

diff --git a/src/Controller.java b/src/Controller.java
© hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:
[javac] Compiling 5 source files to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0.001 sec
[junit] FAILED
[junit] expected:<8.0> but was:<7.5>
[junit] junit.framework.AssertionFailedError: expected:<8.0> but was:<7.5>
[junit] at BasicStatsTest.testMedian(BasicStatsTest.java:44)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]

BUILD FAILED
/Users/hemanganinagarajan/git/basic-stats/build.xml:39: Test BasicStatsTest failed

Total time: 0 seconds
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % ant -lib lib/ test
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

```

```

init:

compile:
[javac] Compiling 1 source file to /Users/hemanganinagarajan/git/basic-stats/bin

compile.tests:

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds

```

```

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git add src/BasicStats.java
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git commit -m "resolved"
[detached HEAD b02460a] resolved
1 file changed, 1 insertion(+), 1 deletion(-)
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git log
commit b02460a961c9b105d702fe16b2e93c8f217ed87e (HEAD)
Author: Hemangani Nagarajan <nhema2231@gmail.com>
Date: Thu Nov 16 21:56:55 2023 -0500

    resolved

commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:58:11 2016 -0400

    Commented and cleaned up the source code

commit bda5f0214b51f6cb77e00a7869f32119daddb47
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:52:46 2016 -0400

    More refactorings

commit 6462c91631b264a2e6b252335fe77bc3ca14d268
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:51:37 2016 -0400

    Removed autogenerated stubs

commit c5fc3b7de251fa35900c20db2673085565afb5c6
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:48:54 2016 -0400

    Large refactoring for implementation using BasicStats. Added Mode View and accepts now Doubles instead of integers

commit 1f1ab2776685f4bfa251308a4f8e38b34995a7c6
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 22:53:58 2016 -0400

    Laid out implementation for model, view, and controller

commit 946af6f4aa97049f526bf919f17c3b61f77ce07a
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 22:39:40 2016 -0400

    Added skeleton for Views

commit 6c57e680cdc883ca7a690a45c6e3caf722c3009e
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 22:34:34 2016 -0400

    Added skeletons for Controller and Model

commit d2539a7f4eb4f3e988b3c39653666676e72b10f9
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git diff b02460a961c9b105d702fe16b2e93c8f217ed87e > file.diff
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git diff f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f > error.diff
● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % █

```

5. In interactive mode in git bisect, after how many steps (git bisect calls) did you identify the defect-inducing commit?

In the interactive mode of git bisect, we identified the defect-inducing commit after 5 steps. The defective commit was identified in the sixth step of the process.

6. Which git command can you use to undo the defect-inducing commit? Briefly explain what problem may occur when undoing a commit and what best practice generally mitigates this problem.

The matter of reverting a commit that introduced defects can be resolved using the `git revert` command in Git. The purpose of this command is to generate a fresh commit which essentially undoes the modifications implemented in the commit that was previously specified. The following is how we would utilize it:

```
git revert <commit-hash>
```

Potential Issue When Reverting a Commit

One frequent obstacle that may arise during the process of reverting a commit is the occurrence of merge conflicts. Typically, these conflicts occur when the modifications present in the commit that is being reverted coincide with alterations introduced in subsequent commits. When this occurs, Git is incapable of resolving these conflicts automatically; manual intervention is required to resolve them.

Best Practice to Overcome This Challenge

In order to efficiently mitigate the risk of merge conflicts, it is imperative that we possess a comprehensive comprehension of the modifications present in the commit to which we are reverting and their interplay with the present state of the codebase. It is judicious to conduct the revert operation in an independent branch. This approach enables the management of conflicts and the testing of modifications in isolation, thereby guaranteeing the preservation of the main codebase throughout the procedure. We have to review and resolve conflicts carefully.

7. For the above, you used git bisect in interactive mode. Alternatively, you should now write a command `<cmd>` such that the following two calls of `git bisect` will automatically determine the defect-inducing commit:

```
git bisect start HEAD v1.0.0
```

```
git bisect run <cmd>
```

Note: You may write a script (instead of a single command) and call it as your command. If you do, please include the content of your script in your answer.

```
$ script.sh
1  #!/bin/bash
2  # Script name: bisect-script.sh
3
4  # This script is used for automating the git bisect process.
5  # It compiles the project using Ant, runs tests, and determines
6  # if the current commit is good or bad based on test results.
7
8  # Step 1: Clean the project
9  # This removes any previous build artifacts to ensure a clean state.
10 ant clean
11
12 # Check if 'ant clean' was successful. Exit with an error if it fails.
13 if [ $? -ne 0 ]; then
14     echo "Ant clean failed."
15     exit 1
16 fi
17
18 # Step 2: Compile the project
19 # This compiles the source code of the project.
20 ant compile
21
22 # Check if 'ant compile' was successful. Exit with an error if it fails.
23 if [ $? -ne 0 ]; then
24     echo "Ant compile failed."
25     exit 1
26 fi
27
28 # Step 3: Run the tests
29 # This runs the project's tests. Assuming 'ant' returns a non-zero status on failure.
30 ant -lib lib/ test
31
32 # Step 4: Determine the result of the tests
33 # If the tests pass (exit status 0), the commit is good. Otherwise, it's bad.
34 if [ $? -eq 0 ]; then
35     echo "Tests passed. This is a good commit."
36     exit 0
37 else
38     echo "Tests failed. This is a bad commit."
39     exit 1
40 fi
```



```

● hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats % git bisect run sh script.sh
running 'sh' 'script.sh'
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

clean:
[delete] Deleting directory /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
[mkdir] Created dir: /Users/hemanganinagarajan/git/basic-stats/bin

compile:
[javac] Compiling 14 source files to /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:

compile.tests:
[javac] Compiling 1 source file to /Users/hemanganinagarajan/git/basic-stats/bin

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0.001 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
Tests passed. This is a good commit.
Bisecting: 2 revisions left to test after this (roughly 2 steps)
[6462c91631b264a2e6b252335fe77bc3ca14d268] Removed autogenerated stubs
running 'sh' 'script.sh'
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

clean:
[delete] Deleting directory /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL

```

```

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
[mkdir] Created dir: /Users/hemanganinagarajan/git/basic-stats/bin

compile:
[javac] Compiling 15 source files to /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:

compile.tests:
[javac] Compiling 1 source file to /Users/hemanganinagarajan/git/basic-stats/bin

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
[junit]
[junit] Testcase: testMean took 0.001 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0.001 sec

BUILD SUCCESSFUL
Total time: 0 seconds
Tests passed. This is a good commit.
Bisecting: 0 revisions left to test after this (roughly 1 step)
[f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f] Commented and cleaned up the source code
running 'sh' 'script.sh'
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

clean:
[delete] Deleting directory /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
[mkdir] Created dir: /Users/hemanganinagarajan/git/basic-stats/bin

compile:

```

```

compile:
[javac] Compiling 15 source files to /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:

compile.tests:
[javac] Compiling 1 source file to /Users/hemanganinagarajan/git/basic-stats/bin

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec
[junit] Tests run: 4, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0.001 sec
[junit] FAILED
[junit] expected:<8.0> but was:<7.5>
[junit] junit.framework.AssertionFailedError: expected:<8.0> but was:<7.5>
[junit] at BasicStatsTest.testMedian(BasicStatsTest.java:44)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit] at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
[junit]

BUILD FAILED
/Users/hemanganinagarajan/git/basic-stats/build.xml:39: Test BasicStatsTest failed

Total time: 0 seconds
Tests failed. This is a bad commit.
Bisecting: 0 revisions left to test after this (roughly 0 steps)
[bda5f0214b51f6cb77e00a7869f32119daddb47] More refactorings
running 'sh' 'script.sh'
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

clean:
[delete] Deleting directory /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
[mkdir] Created dir: /Users/hemanganinagarajan/git/basic-stats/bin

```

```

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:
[mkdir] Created dir: /Users/hemanganinagarajan/git/basic-stats/bin

compile:
[javac] Compiling 15 source files to /Users/hemanganinagarajan/git/basic-stats/bin

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: /Users/hemanganinagarajan/git/basic-stats/build.xml

init:

compile:

compile.tests:
[javac] Compiling 1 source file to /Users/hemanganinagarajan/git/basic-stats/bin

test:
[echo] Running unit tests ...
[junit] Running BasicStatsTest
[junit] Testsuite: BasicStatsTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
[junit]
[junit] Testcase: testMean took 0 sec
[junit] Testcase: testMode took 0 sec
[junit] Testcase: testCentralTendency took 0 sec
[junit] Testcase: testMedian took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
Tests passed. This is a good commit.
f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f is the first bad commit
commit f7f175dc8bc5b3693f99b2f8e799b51c0d0d9b9f
Author: DeveloperTommy <its.tommy.nguyen@gmail.com>
Date: Mon Sep 26 23:58:11 2016 -0400

    Commented and cleaned up the source code

src/BasicStats.java | 19 ++++++-----
src/Controller.java | 2 +-
src/ModeView.java   | 1 -
src/Model.java      | 1 +
src/ResetCtrl.java  | 1 -
5 files changed, 14 insertions(+), 10 deletions(-)
bisect found first bad commit
○ hemanganinagarajan@Hemanganis-MacBook-Pro basic-stats %

```