

Chapter 5

Injecting Faults into the Code

Introduction

In chapter 5, we injected five faults into the Wheelmap-frontend code (not the test cases themselves), causing five of our test cases to fail. The objective of this exercise was to make at least five, but not all, test cases fail. This was a unique challenge for our team because we are testing a graphical user interface. Many of the components that we wanted to adjust to fail our test cases would break the whole application. Below are the changes we made, including the path of the changed file, the line number altered, the update to that line, and the test case impacted:

Changes and Associated Test Cases

	Path to File	Line Number	Line Changes	
1	<u>Onboarding.js</u>	Line 44	<code>const startButtonCaption = t`Okay, let's gooo!`;</code>	TestCase1
2	<u>Onboarding.js</u>	Line 46	<code>const skipAnalyticsButtonCaption = t`Continue without cookiesss`;</code>	TestCase2
3	<u>/src/components/ Map/ addLocateControlToMap.js</u>	Line 45	<code>title: t`Show me where I ammm`,</code>	TestCase3
4	<u>/src/components/ Map/ addLocateControlToMap.js</u>	Line 1103	<code></code>	TestCase16
5	<u>/src/components/ Map/ addLocateControlToMap.js</u>	Line 1086	<code></code>	TestCase17

Figure 1: Changes to test cases

- **Test Case 1:** We altered the text on the Continue With Cookies button as it was defined within the Oboarding.js file. The difference between this text and the original text in the test case resulted in an assertion error, failing the test.
- **Test Case 2:** This change was very similar to the one in test case 1. We altered the text for the Skip Analytics button within the code while leaving it with the original text in the text case. This again caused an assertion error and a failed test.
- **Test Case 3:** Here we changed the text associated with the location control button that recenters the user's map to his or her location. Since the test case checks for the correct text, this causes it to fail.
- **Test Case 16:** This test case checks that the correct link opens when the user clicks Improve This Map. We altered the associated link within the addLocateControlToMap.js file so that the assertion statement checking the link would fail.
- **Test Case 17:** This test case change used the same logic as that in test case 16, but altered a different link.

TestMap Component Test

Start Time: 2020-11-24 11:20:28

Duration: 49.42 s

Summary: Total: 6, Pass: 1, Fail: 5

TestMap for WheelMap	Component	Requirement	Oracle	Status	
test_001	ImproveThisMap-nav-link	Link redirect	PASS	Fail	View
test_002	sozialhelden-logo	Link redirect	PASS	Fail	View
test_003	CookieButton	elementTextRendered	PASS	Fail	View
test_016	NoCookieButton	elementTextRendered	PASS	Fail	View
test_017	leaflet-interactive	userLocationButtonRendered	PASS	Fail	View
<div> <div>Pass</div> <div></div> </div>					

Total: 6, Pass: 1, Fail: 5 -- Duration: 49.42 s

Figure 2: The failed tests displaying in the test case report

Experience Report

The nature of our project made this an interesting experience for our team. When we were first working together to brainstorm how we could make our existing test cases fail, we realized that with certain changes, the program would present an error in the user interface that our test case would either circumvent (and still pass) or halt on (and present an error rather than a fail). We learned that failure of the assert statement within the test case is truly what causes the test to fail. Our objective then became finding specific enough code items we could change so that a test case assert statement would fail, but the whole application would stay in-tact.