Curtis Motes, Paige Peck, Ryan Lile

CSCI 362 Software Engineering

November 29, 2016

Fault Injections

Experiences:

For this deliverable, we managed meet up the day after Deliverable 4 was presented and write all our fault injections in this one meet up. The first thing that we accomplished was cleaning up the formatting and errors that we had from Deliverable 4. Most of these were spelling errors and output formatting. We also made sure that we had a proper README that fully explained how to run our testing framework and how to add tests to it. Once we completed this task, we began work on our five fault injections. This didn't take too long as we already had an idea on where we wanted to inject these faults. We decided we wanted to make a fault injection into the timer file since we had several tests that could be checked with these faults. The other class we injected faults into was the music file as a lot of our tests ran the Note class and injecting a fault here could help show us that most of our tests would fail. After running the testing framework, we saw that all our tests we expected to fail failed. We then commented out the lines that caused faults with a comment at the end of the line to indicate it was a fault injection and which fault injection it was. We learned that multiple tests in a framework shows that a fault occurred and can more easily show where it occurred when there are multiple calls to the methods. We also learned that one of the methods that was expected to pass before our fault injections fails due to it bypassing a precondition in a separate class.

Fault template:

> Location of file
>
> The line number that was changed, the class, and the method
>
> Original line
>
> Fault injected line
>
> Tests that will fail

Faults:

Fault 1)

> TODO-Team_Name/TestAutomation/project/Jython/library/timer.py
>
> Line 299, Timer2 class, setDelay method
>
> if self._timeInterval == timeInterval:

if self._timeInterval < timeInterval:

Test 5 fails

Fault 2)

TODO-Team_Name/TestAutomation/project/Jython/library/timer.py

Line 320, Timer2 class, start method

if self._running == True

if self._running == False

Test 1 fails

Fault 3)

TODO-Team_Name/TestAutomation/project/Jython/library/timer.py

Line 284, Timer2 class, setRepeat method

self._repeat = flag

self._repeat = !flag

Test 2 fails

Fault 4)

TODO-Team_Name/TestAutomation/project/Jython/library/music.py

Line 1026, Note class, contructor

if type(value) == int and value != REST and (value < 0 or value > 127):

if type(value) == int and value != REST and (value > 0 or value > 127):

Test 9-16,18-22, 24, and 25 fails

Fault 5)

TODO-Team_Name/TestAutomation/project/Jython/library/music.py

Line 1026, Phrase class, addChord method

n = Note(pitches[i], 0.0, dynamic, panoramic, length)

n = Note(pitches[i]+1jhj, 0.0, dynamic, panoramic, length)

Test 16 fails