Automated Testing Using the Sugar Framework

Team Seawolves
Shaina Mainar, David Spry, AJ Williams

Overview of Sugar

sugarlabs

- H/FOSS
- Desktop geared towards educating youth with fun, interactive games
- Python 2-ish
- Still being developed
    - Port to Python 3

The Process

1

Plan

A test plan was developed to make sure that we were hitting milestones and deliverables by the due dates. It was important that we knew what was due and when.

2

Test Framework

A Framework architecture diagram was necessary so that we understood what components needed to be created and how they connected to each other.

3

Fault Injection

Faults were injected in select parts of the source code to ensure that the test framework will still run and catch the errors .

4

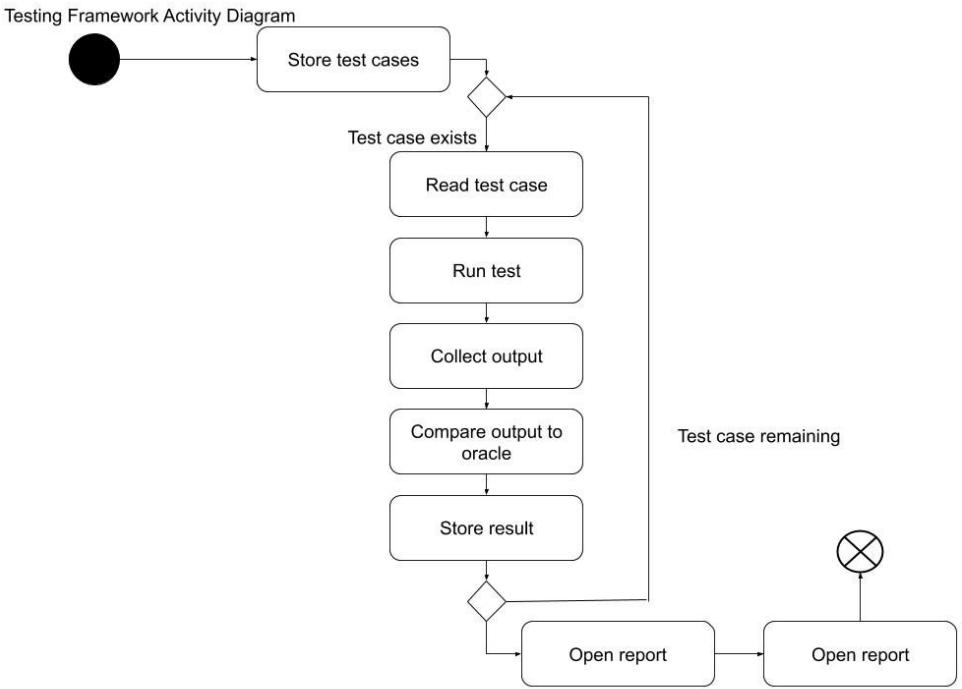
Final Documentation

The five deliverables were compiled into a final report. The findings in the report are presented and a poster was created to compress the information of the final report in an easy to read, professional way.

Why Sugar?

- Simple
- Python can run in command line
- Little to no dependencies
- No drivers

Test Framework



Test Case Template

Test Number: Integer number of the test
Requirement: The Requirement being tested
Component: The name of the class that contains the method to be tested
Method: The method to be tested
Test Input: The input to be used in the test
Expected Outcomes: The oracle or expected outcome from the input

Test Case Example

```
01
Accepts an integer and outputs the birth timestamp in seconds from the epoch
agepicker
agepicker.calculate_birth_timestamp
5
1077212919
```

Fault Injection

functions.factorial at line 302
WAS res *= n - 1
CHANGED TO res = n - 1

functions._do_gcd at line 266
WAS return _do_gcd(b, a % b)
CHANGED TO return _do_gcd(b, a / b)

agepicker.calculate_birth_timestamp at line 48
WAS birth_timestamp = int(1234892919.655932 - age_in_seconds)
CHANGED TO birth_timestamp = (1234892919.655932 - age_in_seconds)

agepicker.calculate_age at line 54
WAS age = int(math.floor(age_in_seconds / _SECONDS_PER_YEAR) + 0.5)
CHANGED TO age = int(math.floor(age_in_seconds / _SECONDS_PER_YEAR))

calculate.findchar at line 60
WAS level -= 1
CHANGED TO level += 1

Test Case	Requirement	Method	Input	Output	Oracle	Result
05	Accepts an age and outputs the birth time stamp in seconds	agepicker.calculate_birth_timestamp	3000000000	-9.46079987651e+16	-9460799876510708	FAIL!
06	Accepts a birth time stamp and outputs the age	agepicker.calculate_age	-1000000	39	39	Pass
07	Accepts a birth time stamp and outputs the age	agepicker.calculate_age	1000000	39	39	Pass
08	Accepts a birth time stamp and outputs the age	agepicker.calculate_age	0	39	39	Pass
09	Accepts a birth time stamp and outputs the age	agepicker.calculate_age	3000000000	-56	-55	FAIL!
10	Accepts a birth time stamp and	agepicker.calculate_age	1.5	39	39	Pass

Specifications

- Ubuntu 18
- Python 2
- Python GI library
- From project root directory, go to Test Automation and run
    - “bash ./scripts/runAllTests.sh” (no quotes)

Results

Demo

Sugar Desktop App
Wed Nov 20 18:03:16 EST 2019

Test Results

Test Case	Requirement	Method	Input	Output	Oracle	Result
01	Accepts an integer and outputs the birth timestamp in seconds from the epoch	agepicker.calculate_birth_timestamp	5	1077212919	1077212919	Pass
02	Accepts a zero and outputs the birth timestamp using the time epoch	agepicker.calculate_birth_timestamp	0	1234892919	1234892919	Pass
03	Accepts a negative integer and outputs the birth timestamp from the time epoch	agepicker.calculate_birth_timestamp	-5	1392572919	1392572919	Pass
04	Accepts a float and outputs the birth timestamp from the epoch	agepicker.calculate_birth_timestamp	5.5	1061444919	1061444919	Pass
05	Accepts a very large long integer and outputs the birth timestamp from the time epoch	agepicker.calculate_birth_timestamp	3000000000	-94607998765107088	-94607998765107088	Pass
06	Accepts a negative integer as the birth time stamp and outputs the age	agepicker.calculate_age	-1000000	39	39	Pass
07	Accepts a positive integer as the birth time stamp and outputs the age	agepicker.calculate_age	1000000	39	39	Pass
08	Accepts a birth time stamp and outputs the age	agepicker.calculate_age	0	39	39	Pass
09	Accepts a long integer as the birth time stamp and outputs the age	agepicker.calculate_age	3000000000	-55	-55	Pass
10	Accepts a float as the birth time stamp and outputs the age	agepicker.calculate_age	1.5	39	39	Pass
11	Accepts an empty string of characters and a character, output the index of the character in the string	calculate.findchar	"", 'a'	-1	-1	Pass
12	Accepts a string characters and an empty character, output the index of the character in the string	calculate.findchar	"abc", ''	-1	-1	Pass

Retrospect

Challenges

- No scripting experience
- Little version control experience
- No documentation
- Project specifications not specific
- Assigning roles
- Incomplete source code (porting to Python 3

Takeaways

- Understanding of Bash scripting
- Architecture frameworks and looking at big picture
- Proper documentation
- Importance of a well thought-out plan
- Working with a team
- Potential industry experience