



# NewLeaf Final Presentation

## Moodle Automated Testing Framework

CSCI 362 - Fall 2020

Luke McGuire, Chris Taylor, Kasper Dugaw

## Picking a Project



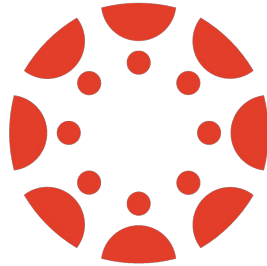


# Canvas

What is it?

Why we considered it?

Why didn't we choose it?



canvas



# WheelMap

What is it?

Why we considered it?

Why didn't we choose it?



**wheelmap.org**  
Rollstuhlgerechte Orte finden.



# Moodle

What is it?

Why we chose it?





# Building Moodle

Good documentation

Final build works

Running their tests

External libraries





# Good Documentation

Step by step instructions

Detailed reasoning

## Step 4: Download Moodle

Setup your local repository and download Moodle, We will use /opt for this installation.

- [Git](#) is what is called a "version control system". By using [git](#) it will be much easier down the road to update the moodle core application. Within Step 5 there is a little more detail on why we put the moodle core application code in the /opt directory.

```
cd /opt
```

Download the Moodle Code and Index

```
sudo git clone git://git.moodle.org/moodle.git
```

Change directory into the downloaded Moodle folder

```
cd moodle
```

Retrieve a list of each branch available

```
sudo git branch -a
```

Tell [git](#) which branch to track or use

```
sudo git branch --track MOODLE_39_STABLE origin/MOODLE_39_STABLE
```

Finally, Check out the Moodle version specified

```
sudo git checkout MOODLE_39_STABLE
```

# Final Build Works

Works with MySQL server

Can edit services

The screenshot displays the NewLeaf dashboard. At the top, there is a navigation bar with a hamburger menu icon, the text 'NewLeaf', and user information for 'Chris Taylor' including a notification bell, a chat bubble, and a profile icon. Below the navigation bar is a sidebar with a 'Dashboard' header and several menu items: 'Site home', 'Calendar', 'Private files', 'Content bank', and 'Site administration'. The main content area is divided into several sections. The 'Recently accessed courses' section shows 'No recent courses'. The 'Course overview' section includes filters for 'All (except removed from view)', 'Course name', and 'Card', and shows 'No courses'. The 'Timeline' section shows 'No upcoming activities due'. The 'Private files' section shows 'No files available' and a link to 'Manage private files...'. The 'Online users' section shows '1 online user (last 5 minutes)' and lists 'Chris Taylor'. The 'Latest badges' section shows 'You have no badges to display'. A 'Customise this page' button is located in the top right corner of the dashboard area.

NewLeaf

Chris Taylor

Customise this page

Dashboard

- Site home
- Calendar
- Private files
- Content bank
- Site administration

Recently accessed courses

No recent courses

Course overview

All (except removed from view) Course name Card

No courses

Show All

Timeline

No upcoming activities due

Private files

No files available

[Manage private files...](#)

Online users

1 online user (last 5 minutes)

Chris Taylor

Latest badges

You have no badges to display



# Running Their Tests

PHPUnit

Overall success or failures

Detailed output

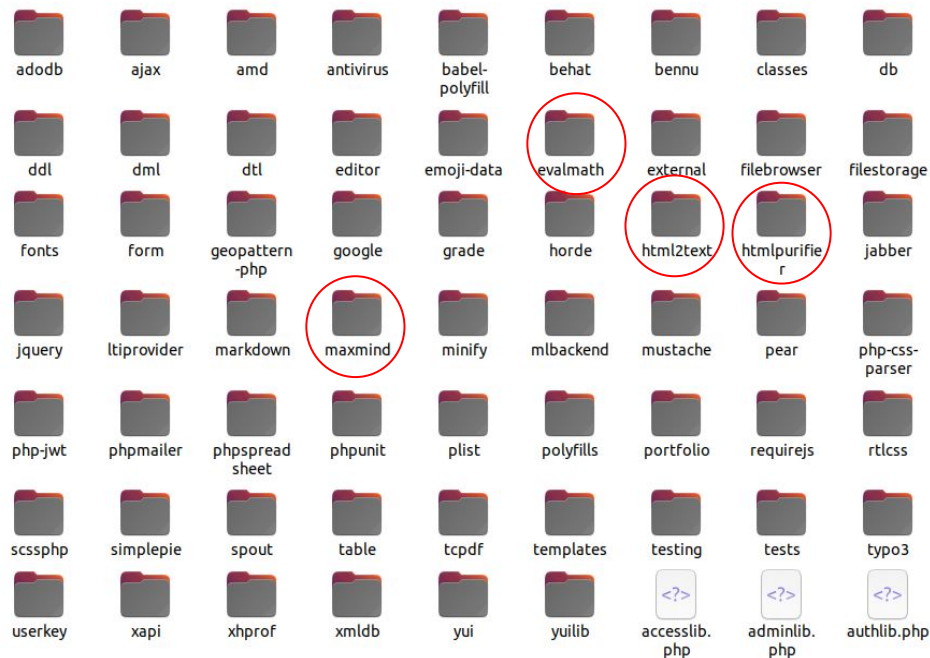
```
Moodle 3.9.2 (Build: 20200914), ccd4ef8ddd03d98b84e3231866b8b1e024dab1db
Php: 7.4.3, mysqli: 8.0.21-0ubuntu0.20.04.4, OS: Linux 5.4.0-47-generic x86_64
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

..... 59 / 15908 ( 0%)
.....ES..... 118 / 15908 ( 0%)
.....F 177 / 15908 ( 1%)
.....SSSSSSSS..... 236 / 15908 ( 1%)
..... 295 / 15908 ( 1%)
..... 354 / 15908 ( 2%)
.....F..... 413 / 15908 ( 2%)
..... 472 / 15908 ( 2%)
..... 531 / 15908 ( 3%)
..... 590 / 15908 ( 3%)
..... 649 / 15908 ( 4%)
..... 708 / 15908 ( 4%)
..... 767 / 15908 ( 4%)
..... 826 / 15908 ( 5%)
..... 885 / 15908 ( 5%)
..... 944 / 15908 ( 5%)
..... 1003 / 15908 ( 6%)
..... 1062 / 15908 ( 6%)
..... 1121 / 15908 ( 7%)
..... 1180 / 15908 ( 7%)
```

# External Libraries

Majority not being tested with PHPUnit

Circles indicate libraries tested with our framework





# The Test Plan

Developing Test Cases

Test Case Template

Test Cases



# Developing Test Cases

Find method to test

Figure out example input

Solve expected output NOT using the method

Create JSON file with required fields



# Test Case Template

```
{  
    "id": Positive integer,  
    "driver": "Driver"  
    "requirement": "Requirement",  
    "component": "Component Name",  
    "method": "Method Name",  
    "input": "Input",  
    "output": "Expected Output"  
}
```



## evaluate() Method

```
{  
    "id": 1,  
    "driver": "evalmath_evaluate.driver.php",  
    "requirement": "basic addition functionality",  
    "component": "evalmath",  
    "method": "evaluate",  
    "input": "4+5",  
    "expected": 9  
}
```



# getText() Method

```
{  
  "id": 6,  
  "driver": "html2text_getText.driver.php",  
  "requirement": "extract text from html with multiple nested tags",  
  "component": "html2text",  
  "method": "getText",  
  "Input": "PCFET0NUWVBFiGh0bWw+PGh0bWw+PGJvZHk+PGgxPlRoZSBIZWFkaW5nPC9oMT48cD5BIHBhcmFncmFwaC48L3A+PC9ib2R5PjwvHRtbD4=",  
  "expected": "VEhFIEhFQURJTkcKcEgcGFyYWdyYXB0Lgo="
```

THE HEADING

A paragraph.

```
<!DOCTYPE html> <html><body>  
  <h1>The Heading</h1>  
  <p>A paragraph.</p>  
</body></html>
```



## cidr() Method

```
{  
    "id": 11,  
    "driver": "maxmind_GeoIp2_Util_cidr.driver.php",  
    "requirement": "convert ip/prefix to cidr notation",  
    "component": "maxmind_GeoIp2_Util",  
    "method": "cidr",  
    "input": "1.2.3.4/16",  
    "expected": "1.2.0.0/16"  
}
```





# getSigFigs() Method

```
{  
    "id": 16,  
    "driver": "htmlpurifier_UnitConverter_getSigFigs.driver.php",  
    "requirement": "calculation of significant figures in decimal strings",  
    "component": "htmlpurifier_UnitConverter",  
    "method": "getSigFigs",  
    "input": "0001.0123",  
    "expected": "5"  
}
```



# ifthenelse() Method

```
{  
  "id": 26,  
  "driver": "evalmath_ifthenelse.driver.php",  
  "requirement": "evaluating if and returning then or else",  
  "component": "evalmath",  
  "method": "ifthenelse",  
  "input": "true,3,4",  
  "expected": 3  
}
```



# Designing the Framework

- Our framework consists of three main components
  - The Controller (runAllTests.py)
  - Drivers
  - Test Cases



# The Controller: runAllTests.py

- Loading Test Cases
  - Test cases are loaded, executed, and saved individually to reduce memory load
- Error Handling
  - The controller handles a variety of errors including malformed test cases and drivers
- Output Generation
  - Output is displayed in the default web browser in a sortable table



# Drivers

Accepts input at as an argument

Separates values by delimiter if needed

Interfaces with the Moodle source code

Passes input to method

Returns the result to stdout


```
1 <?php
2
3 //
4 // maxmind_GeoIp2_Util_cidr.driver.php
5 //
6 // lukem1
7 // Newleaf
8 // 16 November 2020
9 //
10
11
12 // Requirements\
13 require "../project/moodle/lib/maxmind/GeoIp2/Util.php";
14
15
16 // Input must be passed as an argument
17 if (!(sizeof($argv) == 2)) {
18     echo "Error: Driver expected 1 argument and recieved " . (sizeof($argv)-1) . ".\n";
19     return -1;
20 }
21
22
23 // Method to perform a test
24 // Echos: { "output": output }
25 function test($input) {
26     //echo "Input: " . $input . "\n";
27
28     $ary = explode("/", $input);
29
30     $util = new GeoIp2\Util;
31     $output = $util->cidr($ary[0], (int) $ary[1]);
32
33     echo "{ \"output\": \"" . $output . "\" }";
34 }
35
36
37 // Run the test with input from argv
38
39 $in = $argv[1];
40
41 test($in);
```



# Injecting Faults and Testing the Framework

- To test our framework we implemented 5 fault cases to inject into the Moodle source code
- We developed 1 fault case for each of the initial 5 drivers we developed
- These fault cases are designed to mimic errors we though a developer might make


## Fault 1




```
138 protected $callbackSearch = array(  
139     '/<(h)[123456]( [^>]*)?>(.*?)</h[123456]>/i',           // h1 - h6  
140     '/[ ]*<(p)( [^>]*)?>(.*?)</p>[ ]*/si',                 // <p> with surrounding whitespace.  
141  
142     // FAULT: typo br -> brk  
143     '/<(brk)[^>]*>[ ]*/i',                                   // <br> with leading whitespace after the newline.  
144  
145     '/<(b)( [^>]*)?>(.*?)</b>/i',                             // <b>  
146     '/<(strong)( [^>]*)?>(.*?)</strong>/i',                 // <strong>  
147     '/<(th)( [^>]*)?>(.*?)</th>/i',                         // <th> and </th>  
148     '/<(a) [^>]*href="(\\'|\\")(\\'|\\")\\2([>]*)?>(.*?)</a>/i' // <a href="">  
149 );
```

In the definition of a regex filter used for finding html break elements we replaced `<br>` with `<brk>`.

## Fault 2



```
17 public static function cidr($ipAddress, $prefixLen)
18 {
19     $ipBytes = inet_pton($ipAddress);
20     $networkBytes = str_repeat("\0", \strlen($ipBytes));
21
22     $curPrefix = $prefixLen;
23     for ($i = 0; $i < \strlen($ipBytes)-1 && $curPrefix > 0; $i++) { // FAULT: Added -1
24         $b = $ipBytes[$i];
25         if ($curPrefix < 8) {
26             $shiftN = 8 - $curPrefix;
27             $b = \chr(0xFF & (\ord($b) >> $shiftN) << $shiftN);
28         }
29         $networkBytes[$i] = $b;
30         $curPrefix -= 8;
31     }
32 }
```



We modified the execution range of the for loop here.




### Fault 3



```
192 public function getSigFigs($n)
193 {
194     $n = ltrim($n, '0+'); // FAULT: Changed '0+-' to '0+'
195     $dp = strpos($n, '.'); // decimal position
196     if ($dp === false) {
197         $sigfigs = strlen(rtrim($n, '0'));
198     } else {
199         $sigfigs = strlen(ltrim($n, '0. ')); // eliminate extra decimal character
200         if ($dp !== 0) {
201             $sigfigs--;
202         }
203     }
204     return $sigfigs;
205 }
```

Here we removed a minus sign so that this function no longer trims negative numbers properly.

## Faults 4-5



```
// if the token is a binary operator, pop two values off the stack, do the operation, and push the result back on
} elseif (in_array($token, array('+', '-', '*', '/', '^', '>', '<', '==', '<=', '>='), true)) { // FAULT: Changed '-' to '+'
    if (is_null($op2 = $stack->pop())) return $this->trigger(get_string('internalerror', 'mathslib'));
    if (is_null($op1 = $stack->pop())) return $this->trigger(get_string('internalerror', 'mathslib'));
    switch ($token) {
        case '+':
            $stack->push($op1+$op2); break;
        case '-':
            $stack->push($op1-$op2); break;
        case '*':
```

```
585     static function mod($op1, $op2) {
586         return $op2 % $op1; // FAULT: Swapped the locations of $op2, $op1
587     }
```

In these two cases we placed a + operator where a - should be and reversed the order of the variables in a modulo operation.



# Automating Fault Injection With MoodleMod

- To automate the fault injection process we developed another script  
`./scripts/moodleMod.sh <action>`
- Inject action
  - Copies fault definition files into the Moodle source code
- Reset action
  - Resets the Moodle repository to its original state

This script also automates the process of cloning the Moodle repo with the clone action.



# Five Step Setup

1. **Install PHP:** `sudo apt install php7.4-cli ...`
2. **Clone the repository:** `git clone https://github.com/csci-362-02-2020/New-Leaf`
3. **Clone Moodle:** `./scripts/moodleMod.sh clone`
4. **(optional) Inject faults:** `./scripts/moodleMod.sh inject`
5. **Run the tests:** `./scripts/runAllTests.py`

| id | component         | method     | input                | expected         | output           | result |
|----|-------------------|------------|----------------------|------------------|------------------|--------|
| 1  | evalmath          | evaluate   | 4+5                  | 9                | 9                | Pass   |
| 2  | evalmath          | evaluate   | 5-6                  | -1               | -1               | Pass   |
| 3  | evalmath          | evaluate   | 5*5                  | 25               | 25               | Pass   |
| 4  | evalmath          | evaluate   | 25/5                 | 5                | 5                | Pass   |
| 5  | evalmath          | evaluate   | 100>10               | 1                | 1                | Pass   |
| 6  | html2text         | getText    | PCFETONUW ...        | VehFIEhFQU ...   | VehFIEhFQU ...   | Pass   |
| 7  | html2text         | getText    | PHA+cGFyYW...        | cGFyYWdyYX ...   | cGFyYWdyYX ...   | Pass   |
| 8  | html2text         | getText    | PHA+cGFyYW...        | cGFyYWdyYX ...   | cGFyYWdyYX ...   | Pass   |
| 9  | html2text         | getText    | PGgxPmhIYW ...       | SEVBREIORz ...   | SEVBREIORz ...   | Pass   |
| 10 | html2text         | getText    | PGJvZHk+PH ...       | cGFyYWdyYX ...   | cGFyYWdyYX ...   | Pass   |
| 11 | maxmind_Geo ...   | cidr       | 1.2.3.4/16           | 1.2.0.0/16       | 1.2.0.0/16       | Pass   |
| 12 | maxmind_Geo ...   | cidr       | 1.2.3.4/31           | 1.2.3.4/31       | 1.2.3.4/31       | Pass   |
| 13 | maxmind_Geo ...   | cidr       | 1.2.3.4/0            | 0.0.0.0/0        | 0.0.0.0/0        | Pass   |
| 14 | maxmind_Geo ...   | cidr       | fff.fff.fff.fff.1... | fff.:/16         | fff.:/16         | Pass   |
| 15 | maxmind_Geo ...   | cidr       | 2acd:fff:aabb...     | 2acd:fff:aa00... | 2acd:fff:aa00... | Pass   |
| 15 | maxmind_Geo ...   | cidr       | 2acd:fff:aabb...     | 2acd:fff:aa00... | 2acd:fff:aa00... | Pass   |
| 16 | htmlpurifier_ ... | getSigFigs | 0001.0123            | 5                | 5                | Pass   |
| 17 | htmlpurifier_ ... | getSigFigs | 1.1                  | 2                | 2                | Pass   |
| 18 | htmlpurifier_ ... | getSigFigs | 12.304               | 5                | 5                | Pass   |
| 19 | htmlpurifier_ ... | getSigFigs | -10.01               | 4                | 4                | Pass   |
| 20 | htmlpurifier_ ... | getSigFigs | 000.000              | 0                | 0                | Pass   |
| 21 | evalmath          | mod        | 7,7                  | 0                | 0                | Pass   |
| 22 | evalmath          | mod        | 7,6                  | 1                | 1                | Pass   |
| 23 | evalmath          | mod        | 100,50               | 0                | 0                | Pass   |
| 24 | evalmath          | mod        | 50,60                | 50               | 50               | Pass   |
| 25 | evalmath          | mod        | 1,1                  | 0                | 0                | Pass   |
| 26 | evalmath          | ifthenelse | true,3,4             | 3                | 3                | Pass   |
| 27 | evalmath          | ifthenelse | 1,3,4                | 3                | 3                | Pass   |
| 28 | evalmath          | ifthenelse | 0,3,4                | 4                | 4                | Pass   |
| 29 | evalmath          | ifthenelse | 1+1==2,3,4           | 3                | 3                | Pass   |
| 30 | evalmath          | ifthenelse | 0==0,7,4             | 7                | 7                | Pass   |

Summary: 30 / 30 tests passed.

# Results

Left: Results prior to fault injection.

Right: Results post fault injection.

| id | component         | method     | input                | expected         | output                           | result |
|----|-------------------|------------|----------------------|------------------|----------------------------------|--------|
| 1  | evalmath          | evaluate   | 4+5                  | 9                | 9                                | Pass   |
| 2  | evalmath          | evaluate   | 5-6                  | -1               | PHP Fatal error: Uncaught Error: | Fail   |
| 3  | evalmath          | evaluate   | 5*5                  | 25               | 25                               | Pass   |
| 4  | evalmath          | evaluate   | 25/5                 | 5                | 5                                | Pass   |
| 5  | evalmath          | evaluate   | 100>10               | 1                | 1                                | Pass   |
| 6  | html2text         | getText    | PCFETONUW ...        | VehFIEhFQU ...   | VehFIEhFQU ...                   | Pass   |
| 7  | html2text         | getText    | PHA+cGFyYW...        | cGFyYWdyYX ...   | cGFyYWdyYX ...                   | Pass   |
| 8  | html2text         | getText    | PHA+cGFyYW...        | cGFyYWdyYX ...   | cGFyYWdyYX ...                   | Fail   |
| 9  | html2text         | getText    | PGgxPmhIYW ...       | SEVBREIORz ...   | SEVBREIORz ...                   | Pass   |
| 10 | html2text         | getText    | PGJvZHk+PH ...       | cGFyYWdyYX ...   | cGFyYWdyYX ...                   | Pass   |
| 11 | maxmind_Geo ...   | cidr       | 1.2.3.4/16           | 1.2.0.0/16       | 1.2.0.0/16                       | Pass   |
| 12 | maxmind_Geo ...   | cidr       | 1.2.3.4/31           | 1.2.3.4/31       | 1.2.3.0/31                       | Fail   |
| 13 | maxmind_Geo ...   | cidr       | 1.2.3.4/0            | 0.0.0.0/0        | 0.0.0.0/0                        | Pass   |
| 14 | maxmind_Geo ...   | cidr       | fff.fff.fff.fff.1... | fff.:/16         | fff.:/16                         | Pass   |
| 15 | maxmind_Geo ...   | cidr       | 2acd:fff:aabb...     | 2acd:fff:aa00... | 2acd:fff:aa00...                 | Pass   |
| 16 | htmlpurifier_ ... | getSigFigs | 0001.0123            | 5                | 5                                | Pass   |
| 17 | htmlpurifier_ ... | getSigFigs | 1.1                  | 2                | 2                                | Pass   |
| 18 | htmlpurifier_ ... | getSigFigs | 12.304               | 5                | 5                                | Pass   |
| 19 | htmlpurifier_ ... | getSigFigs | -10.01               | 4                | 5                                | Fail   |
| 20 | htmlpurifier_ ... | getSigFigs | 000.000              | 0                | 0                                | Pass   |
| 21 | evalmath          | mod        | 7,7                  | 0                | 0                                | Pass   |
| 22 | evalmath          | mod        | 7,6                  | 1                | 6                                | Fail   |
| 23 | evalmath          | mod        | 100,50               | 0                | 50                               | Fail   |
| 24 | evalmath          | mod        | 50,60                | 50               | 10                               | Fail   |
| 25 | evalmath          | mod        | 1,1                  | 0                | 0                                | Pass   |
| 26 | evalmath          | ifthenelse | true,3,4             | 3                | 3                                | Pass   |
| 27 | evalmath          | ifthenelse | 1,3,4                | 3                | 3                                | Pass   |
| 28 | evalmath          | ifthenelse | 0,3,4                | 4                | 4                                | Pass   |
| 29 | evalmath          | ifthenelse | 1+1==2,3,4           | 3                | 3                                | Pass   |
| 30 | evalmath          | ifthenelse | 0==0,7,4             | 7                | 7                                | Pass   |

Summary: 23 / 30 tests passed.



# Live Demo



# Conclusions

- How our Framework Turned Out
  - Efficiency and error handling
  - Report generation and features
  - Automation and ease of use
- Overall Thoughts
  - Project Management Skills
  - Designing and Implementing Tests
  - Working on Open Source projects

# Q&A

## Introduction

In this project our team designed, implemented, and tested an automated PHP testing framework and several test cases for Moodle, an open source learning management system which we found lacked tests for external libraries included in their source code.



## Framework Design

Our framework consists of three main components, the controller, test cases, and drivers.

The controller is responsible for parsing and executing test cases as well as generating the results into a report.

The test cases are a set of JSON files which specify individual tests and provide the controller with the data necessary to execute them including the driver that should be used to run the test, the input that should be given to the functionality being tested, and the expected output of the provided input.

Finally, the drivers are responsible for interfacing with the Moodle code and providing the controller with the ability to test a particular method or class and returning the results of a test back to the controller.

## Implementing Tests

After implementing our framework we developed 30 tests for the Moodle project for 6 functions found in the project's libraries. The modular design of our framework provides a simple framework for creating new drivers and tests, and examples of these can be seen in Figures 3 and 4 respectively.



## NewLeaf Automated Testing Framework

Luke McGuire, Chris Taylor, Kasper Dugaw  
College of Charleston Department of Computer Science

| test_id | test_name    | status | result |
|---------|--------------|--------|--------|
| 1       | test_name_1  | pass   | pass   |
| 2       | test_name_2  | pass   | pass   |
| 3       | test_name_3  | pass   | pass   |
| 4       | test_name_4  | pass   | pass   |
| 5       | test_name_5  | pass   | pass   |
| 6       | test_name_6  | pass   | pass   |
| 7       | test_name_7  | pass   | pass   |
| 8       | test_name_8  | pass   | pass   |
| 9       | test_name_9  | pass   | pass   |
| 10      | test_name_10 | pass   | pass   |
| 11      | test_name_11 | pass   | pass   |
| 12      | test_name_12 | pass   | pass   |
| 13      | test_name_13 | pass   | pass   |
| 14      | test_name_14 | pass   | pass   |
| 15      | test_name_15 | pass   | pass   |
| 16      | test_name_16 | pass   | pass   |
| 17      | test_name_17 | pass   | pass   |
| 18      | test_name_18 | pass   | pass   |
| 19      | test_name_19 | pass   | pass   |
| 20      | test_name_20 | pass   | pass   |
| 21      | test_name_21 | pass   | pass   |
| 22      | test_name_22 | pass   | pass   |
| 23      | test_name_23 | pass   | pass   |
| 24      | test_name_24 | pass   | pass   |
| 25      | test_name_25 | pass   | pass   |
| 26      | test_name_26 | pass   | pass   |
| 27      | test_name_27 | pass   | pass   |
| 28      | test_name_28 | pass   | pass   |
| 29      | test_name_29 | pass   | pass   |
| 30      | test_name_30 | pass   | pass   |

Figure 1: Test results prior to fault injection.

| test_id | test_name    | status | result |
|---------|--------------|--------|--------|
| 1       | test_name_1  | pass   | pass   |
| 2       | test_name_2  | fail   | fail   |
| 3       | test_name_3  | pass   | pass   |
| 4       | test_name_4  | fail   | fail   |
| 5       | test_name_5  | pass   | pass   |
| 6       | test_name_6  | fail   | fail   |
| 7       | test_name_7  | pass   | pass   |
| 8       | test_name_8  | fail   | fail   |
| 9       | test_name_9  | pass   | pass   |
| 10      | test_name_10 | fail   | fail   |
| 11      | test_name_11 | pass   | pass   |
| 12      | test_name_12 | fail   | fail   |
| 13      | test_name_13 | pass   | pass   |
| 14      | test_name_14 | fail   | fail   |
| 15      | test_name_15 | pass   | pass   |
| 16      | test_name_16 | fail   | fail   |
| 17      | test_name_17 | pass   | pass   |
| 18      | test_name_18 | fail   | fail   |
| 19      | test_name_19 | pass   | pass   |
| 20      | test_name_20 | fail   | fail   |
| 21      | test_name_21 | pass   | pass   |
| 22      | test_name_22 | fail   | fail   |
| 23      | test_name_23 | pass   | pass   |
| 24      | test_name_24 | fail   | fail   |
| 25      | test_name_25 | pass   | pass   |
| 26      | test_name_26 | fail   | fail   |
| 27      | test_name_27 | pass   | pass   |
| 28      | test_name_28 | fail   | fail   |
| 29      | test_name_29 | pass   | pass   |
| 30      | test_name_30 | fail   | fail   |

Figure 2: Test results post fault injection.

## Testing and Results

To validate our testing framework we designed a system to inject faults into the Moodle source code and executed our framework before and after injecting the faults. The results of this procedure can be seen in Figures 1 and 2.

```

1 // Moodle Core
2 // Moodle Core
3 // Moodle Core
4 // Moodle Core
5 // Moodle Core
6 // Moodle Core
7 // Moodle Core
8 // Moodle Core
9 // Moodle Core
10 // Moodle Core
11 // Moodle Core
12 // Moodle Core
13 // Moodle Core
14 // Moodle Core
15 // Moodle Core
16 // Moodle Core
17 // Moodle Core
18 // Moodle Core
19 // Moodle Core
20 // Moodle Core
21 // Moodle Core
22 // Moodle Core
23 // Moodle Core
24 // Moodle Core
25 // Moodle Core
26 // Moodle Core
27 // Moodle Core
28 // Moodle Core
29 // Moodle Core
30 // Moodle Core
31 // Moodle Core
32 // Moodle Core
33 // Moodle Core
34 // Moodle Core
35 // Moodle Core
36 // Moodle Core
37 // Moodle Core
38 // Moodle Core
39 // Moodle Core
40 // Moodle Core
41 // Moodle Core
42 // Moodle Core
43 // Moodle Core
44 // Moodle Core
45 // Moodle Core
46 // Moodle Core
47 // Moodle Core
48 // Moodle Core
49 // Moodle Core
50 // Moodle Core
51 // Moodle Core
52 // Moodle Core
53 // Moodle Core
54 // Moodle Core
55 // Moodle Core
56 // Moodle Core
57 // Moodle Core
58 // Moodle Core
59 // Moodle Core
60 // Moodle Core
61 // Moodle Core
62 // Moodle Core
63 // Moodle Core
64 // Moodle Core
65 // Moodle Core
66 // Moodle Core
67 // Moodle Core
68 // Moodle Core
69 // Moodle Core
70 // Moodle Core
71 // Moodle Core
72 // Moodle Core
73 // Moodle Core
74 // Moodle Core
75 // Moodle Core
76 // Moodle Core
77 // Moodle Core
78 // Moodle Core
79 // Moodle Core
80 // Moodle Core
81 // Moodle Core
82 // Moodle Core
83 // Moodle Core
84 // Moodle Core
85 // Moodle Core
86 // Moodle Core
87 // Moodle Core
88 // Moodle Core
89 // Moodle Core
90 // Moodle Core
91 // Moodle Core
92 // Moodle Core
93 // Moodle Core
94 // Moodle Core
95 // Moodle Core
96 // Moodle Core
97 // Moodle Core
98 // Moodle Core
99 // Moodle Core
100 // Moodle Core

```

Figure 3: One of the drivers written for our framework.

```

1 {
2   "id": 1,
3   "driver": "evaluator.evaluate_driver.php",
4   "requirements": "basic, addition, functionality",
5   "component": "evaluator",
6   "method": "evaluate",
7   "input": "4+5",
8   "expected": 9
9 }

```

Figure 4: One of the test cases written for our framework.

## Conclusions

Over the course of this project our team learned quite a bit about the software engineering process, including project planning and management, designing and producing good tests, and how to effectively work as a team on a software project.

We were also pleased with how our testing framework itself ended up turning out. Although we had some setbacks, we were not only able to design and implement our vision, but expand upon it. After completing the basic functionality of our framework earlier than expected we were able to spend time improving our project by adding features including improved output generation, improved error handling, and scripts to automate the process of fault injection. Additionally, we were also glad we were able to produce something we felt was valuable by writing our tests for Moodle libraries that did not have pre-existing tests.

To conclude, if it's not already clear, we were very satisfied with the results of our project and our development as a team and as software engineers.

## Acknowledgments

Thanks to the Moodle project and its contributors:  
<https://moodle.org/>

Thanks to our instructor and faculty advisor:  
Dr. Jim Bowring

## Additional Information

Our GitHub repository, which contains the entirety of our project including source code and our final report can be found at the link below.

<https://github.com/csci-362-02-2020/New-Leaf>

