

Josh Gilley, Alex Wizes, Clae Wyckoff

CSCI 362 - Software Engineering

Professor Bowring

10/1/2020

Chapter 2

Testing plan:

Testing Process:

- In order to test the desired method of the class we are interested in, we want to call a program at the command line that runs each test. We can do this for individual methods or we can call a specific program that automatically runs all of the tests at once. The results of the tests will be compared to the expected results of the test cases, and then these results will be displayed in the browser.

Requirements Traceability:

- TC01 | Creating a valid timestamp
- Timestamps are very important for every system, particularly for our endeavors in the Unix systems. In such systems, they are based on a running total of seconds that begins at the Unix epoch, which is advantageous as they can represent all timezones with a single measurement. As such, this facilitates users' ability to see a website (like Moodle) in real time as opposed to the timezone the server is located in.

Tested Items:

- "Convert_to_timestamp" in the "type_base" class of the Moodle project
- /moodle/calendar/classes/type_base.php

Testing Schedule:

- For the next week, we need to establish core functionality between script, driver, and php testing classes. Once established, we will spend this first week testing the Unix

Timestamp method of the type_base class with its many test cases. The following weeks will be dedicated to applying the same techniques to other test cases that we will discover. Methods that use any arithmetic to calculate grades, or calendar dates. Successful connections to the database / browser could be another test. We are currently looking into this.

Test Recording Procedures:

- All of the actual test results will be compared to the expected test results contained in various appropriately labeled json files.
- The results of the comparisons described above will be displayed in a browser. The information will be properly and clearly labeled in a uniform manner with each test case getting its own portion of the report.

Hardware and Software Requirements:

- Linux OS
- A computer

Constraints:

- We have a limited schedule to figure out the relation of different pieces to our framework.

Example of our testing template:

```
"testId": "TC01",  
"designedBy": "Clae, Josh & Alex",  
"testDesignedDate": "10/04/2020",  
"classTested": "type_base",  
"methodTested": "convert_to_timestamp",  
"testingInputs": "2017, 02, 11, 17, 34",  
"Expected Output": "1486834460",  
"success": "pass"
```