

2021-05-27

Testing Tutor: Integrating an Inquiry-Based Learning Approach into Testing Pedagogy

Manuel Larios
Western Oregon University, mlarios12@wou.edu

Jason Weber
Western Oregon University, jweber18@wou.edu

Sam Torris
Western Oregon University, storris20@wou.edu

Nathan Camuso
Western Oregon University, ncamuso20@wou.edu

Jeffrey Carver
University of Alabama, carver@cs.ua.edu

See next page for additional authors

Follow this and additional works at: <https://digitalcommons.wou.edu/aes>

Recommended Citation

Larios, Manuel; Weber, Jason; Torris, Sam; Camuso, Nathan; Carver, Jeffrey; Walia, Gursimran; and Cordova, Lucas, "Testing Tutor: Integrating an Inquiry-Based Learning Approach into Testing Pedagogy" (2021). *Academic Excellence Showcase Proceedings*. 300.
<https://digitalcommons.wou.edu/aes/300>

This Poster is brought to you for free and open access by the Student Scholarship at Digital Commons@WOU. It has been accepted for inclusion in Academic Excellence Showcase Proceedings by an authorized administrator of Digital Commons@WOU. For more information, please contact digitalcommons@wou.edu, kundas@mail.wou.edu, bakersc@mail.wou.edu.

Author(s)

Manuel Larios, Jason Weber, Sam Torris, Nathan Camuso, Jeffrey Carver, Gursimran Walia, and Lucas Cordova

Testing Tutor: Integrating an Inquiry-Based Learning Approach into Testing Pedagogy



Jason Weber, Manuel Larios, Sam Torris, Nathan Camuso, and Lucas Cordova (PI), Western Oregon University
 Gursimran Walia (PI), Georgia Southern University
 Jeffrey Carver (PI), University of Alabama

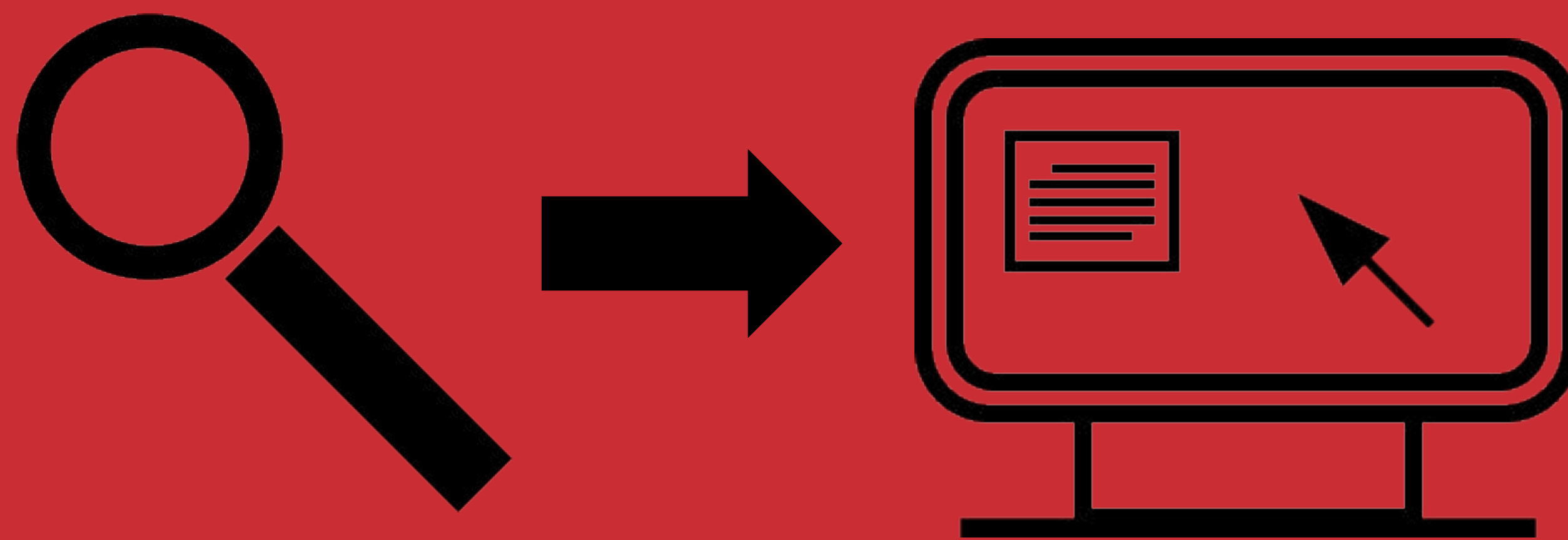


ABSTRACT

Software testing is an inherently critical aspect of computer science education, however current pedagogical and industry tools lack inductive learning approaches which lead to trial-and-error behavior. Research into software testing pedagogy and the construction of a learning tool (Testing Tutor) is being developed in collaboration with University of Alabama and Georgia Southern University, made possible by an NSF grant.

Testing Tutor will identify the testing concepts that are attributable to missing test cases and will provide feedback (via a customizable feedback engine) to correct test case deficiencies along with learning resources (textual and video). WOU's research team is focused on building the subsystem to support the Python and C# languages, which WOU and many universities utilize in their curriculum. In addition, the team is developing a repository of assignments that will be shared with instructors at other institutions.

SUMMARY



The Testing Tutor prototype used in initial empirical studies by the PIs has shown a positive effect on students' learning of testing concepts and increased test code quality. WOU's research team is focused on extending Testing Tutor to architecturally support more programming languages, building a repository of reference assignments and test suites to share with faculty from other institutions, improve the feedback engine, and design additional empirical studies for the next academic year.

RESULTS

Designed a microservice cloud/server architecture for the Testing Tutor platform as a way to make adding new programming languages easier. Implemented an engine for the Python programming language to support WOU's lower division programming courses. Contributing reference assignments and test suites to the Testing Tutor assignment repository.

Additionally, if a student selects one of the concepts that Testing Tutor has identified as an area needing improvement, they will be shown resources related to that concept.

| Submission # | Submitted | Mode | Tests Covered (%) |
|--------------|------------------------|---------------|---------------------------|
| 1 | 11/25/2019 11:28:30 PM | Learning Mode | Engine threw an Exception |
| 2 | 11/25/2019 11:31:13 PM | Learning Mode | Engine threw an Exception |
| 3 | 11/25/2019 11:39:46 PM | Learning Mode | 20% |
| 4 | 11/25/2019 11:43:11 PM | Learning Mode | 40% |
| 5 | 11/25/2019 11:47:02 PM | Learning Mode | 60% |
| 6 | 11/25/2019 11:48:11 PM | Learning Mode | 80% |

Submission #6 Feedback - Demo Assignment - Feedback at Process Level Revised

Coverage and Redundant Tests

Feedback

Open Issues: Not all cases covered for Overdrawing account, Not all cases covered for Withdrawal

Closed Issues: All cases covered for Deposit

Learning Concept(s): You have missed this test case. [Double check the boundary condition\(s\) around your logic.](#)

[Branch coverage conceptual concept](#)

Submission history & progress

Instructor can choose to show code coverage

Open/closed equivalence class issues

Links to learning concepts

The figure above illustrates the feedback that Testing Tutor provides to students.

Boundary Value Analysis and Equivalence Partitioning

Software Testing TUTORIAL

In software testing, the Boundary Value Analysis (BVA) is a black box test design technique based on test cases. This technique is applied to see if there are any bugs at the boundary of the input domain. Thus, with this method, there is no need of looking for these errors at the center of this input.

BVA helps in testing the value of boundary between both valid and invalid boundary partitions. With this technique, the boundary values are tested by the creation of test cases for a particular input field.

Boundary Value Analysis

Boundary

Boundary Values

The extreme ends or boundary partitions might depict the values of lower-upper, start-end, maximum-minimum, inside-outside etc.

FEATURES

- Instructors may assign assignments from or add assignments to a public, private, or organization repository.
- Assignments may be in Java, Python, or C#.
- In **Learning Mode**, students are given feedback based on their testing coverage of instructor code.
- In **Development Mode**, students are given feedback based on their testing coverage of their own code.

Next Steps: Finish developing Python and begin developing C# engine. Design studies to conduct at WOU for the 2021-22 school year.